Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων
Πληροφορικής

Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης
(Zero Trust Networking Architecture με το OpenZiti project)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Οδυσσέας Γ. Λιοδάκης

Επιβλέπων : Ευστάθιος Δ. Συκάς
Ομότιμος Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2025

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων
Πληροφορικής

Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης

(Zero Trust Networking Architecture με το OpenZiti project)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Οδυσσέας Γ. Λιοδάκης

Επιβλέπων : Ευστάθιος Δ. Συκάς

Ομότιμος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10η Μαρτίου 2025.

.....	.....	.....
Ιωάννα Ρουσσάκη	Ευστάθιος Συκάς	Νικόλαος Μήτρου
Αν. Καθηγήτρια Ε.Μ.Π	Ομότιμος Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π

Αθήνα, Μάρτιος 2025

.....

Οδυσσέας Γ. Λιοδάκης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει την Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης μέσω της πλατφόρμας ανοικτού κώδικα “OpenZiti”. Αρχικά τίθεται το θεωρητικό υπόβαθρο του θέματος. Παρουσιάζονται οι αρχές που διέπουν την Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης, αλλά και οι μηχανισμοί που μπορούν να χρησιμοποιηθούν για την εφαρμογή τους. Γίνεται εκτενής αναφορά σε ασφαλείς μηχανισμούς ταυτοποίησης χρηστών και εφαρμογών, που αποτελεί και τον πυρήνα της Αρχιτεκτονικής. Περιγράφεται η λειτουργία των ψηφιακών πιστοποιητικών X.509, των Javascript Object Notation Web Tokens (JWTs) αλλά και των Time-based One-Time Passwords (TOTP). Επίσης περιγράφεται το πρωτόκολλο ασφαλών, κρυπτογραφημένων συνδέσεων TLS, διαφορετικές εκδόσεις του, η εκδοχή αμφίδρομης εφαρμογής του (mutual TLS) αλλά και οι μηχανισμοί στους οποίους βασίζεται για τη δημιουργία συνδέσεων, όπως ο αλγόριθμος δημιουργίας κοινών κρυπτογραφικών κλειδιών Diffie-Hellman. Κατόπιν παρουσιάζεται η πλατφόρμα δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti. Γίνεται αναφορά στα στοιχεία λογισμικού, καθώς και τις λογικές οντότητες που χρησιμοποιεί για τις λειτουργίες της. Στα πλαίσια της παρούσας διπλωματικής εργασίας αναπτύχθηκε εφαρμογή γραμμής εντολών που έχει τη δυνατότητα να αποστέλλει αιτήματα HTTP μέσω δικτύου που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης χρησιμοποιώντας την πλατφόρμα OpenZiti. Η εφαρμογή γραμμής εντολών χρησιμοποιεί τη βιβλιοθήκη του cURL project, libcurl(), και κατάλληλα εργαλεία ανάπτυξης λογισμικού της πλατφόρμας OpenZiti για τη γλώσσα προγραμματισμού C (OpenZiti SDK-C). Τέλος, προκειμένου να εξεταστεί ενδεχόμενη επίδραση της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης στις επιδόσεις των εφαρμογών πραγματοποιήθηκε συγκριτική μελέτη μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip) σε αιτήματα HTTP που αποστέλλονταν από την εφαρμογή γραμμής εντολών cURL προς την εφαρμογή διαχείρισης περιεχομένου βάσεων δεδομένων FIWARE Orion Context Broker, λαμβάνοντας υπ’ οψιν την υιοθέτηση ή μη της Αρχιτεκτονικής από τα αντίστοιχα δίκτυα που χρησιμοποιήθηκαν κατά τη διεξαγωγή των μετρήσεων.

Λέξεις-κλειδιά

Αρχιτεκτονική Μηδενικής Εμπιστοσύνης, Ασφάλεια Δικτύων, Ταυτοποίηση, Κρυπτογράφηση, RSA, mTLS, Diffie-Hellman, JWT, X.509, TOTP, HTTPS

Abstract

This diploma thesis examines Zero Trust Networking Architecture by utilizing the open-source platform “OpenZiti”. At first the theoretical background of the topic is established. The principles defining Zero Trust Networking Architecture are presented, along with the mechanisms that can be used to implement them. Secure authentication mechanisms, which form the core of the Architecture, are extensively discussed. The operation of X.509 digital certificates, Javascript Object Notation Web Tokens (JWTs) and Time-based One-Time Passwords (TOTP) is described. The TLS protocol for secure, encrypted connections is also described, including different versions, its bidirectional implementation (mutual TLS), and the mechanisms it relies on for establishing connections, such as the Diffie-Hellman cryptographic key exchange algorithm. The open-source Zero-Trust Networking Architecture platform “OpenZiti” is then presented. Its software components and the logical entities it uses for its operations are discussed. As part of this thesis, a command-line application was developed that can send HTTP requests through a Network which adopts Zero Trust Networking Architecture using the OpenZiti platform. The application uses cURL project’s library, libcurl(), and OpenZiti platform’s software development Kit (SDK) for the C programming language. Finally, to examine the potential impact of Zero Trust Networking Architecture on application performance, a comparative study of Round-Trip latency measurements was conducted on HTTP requests sent by cURL project’s command-line tool to the FIWARE Orion Context Broker context management application, taking into consideration the adoption or not of the Architecture by the respective Networks used during the measurements’ conduct.

Keywords

Zero Trust Architecture, Network Security, Authentication, Encryption, RSA, mTLS, Diffie-Hellman, JWT, X.509, TOTP, HTTPS

Πίνακας Περιεχομένων

Ευρετήριο Εικόνων, Πινάκων και Σχημάτων

Συντομογραφίες - Ακρωνύμια

		σελ.
ACL	Access Control Lists	19
API	Application Programming Interface	32
BJWT	Biometric JSON Web Tokens	32
CA	Certificate Authority	31
CARTA	Continuous Adaptive Risk and Trust Assessment	22
CMDH	Cryptanalysis with Modified Diffie-Hellman key exchange	36
CoAP	Constrained Application Protocol	91
CSR	Certificate Signing Request	51
CVE	Common Vulnerabilities and Exposures	33
DMZ	Demilitarized Zone	20
DNS	Domain Name System	37
DGTOTP	Dynamic Group Time-based One-Time Passwords	34
DID	Decentralized Identifiers	92
dOISP	Decentralized Open Interoperable Security Protocol	30
ECDH	Elliptic Curve Diffie-Hellman	28
ELS	Enterprise Level Security	35
EPS-AKA	Evolved Packet System based Authentication and Key Agreement	36
E2EE	End to End Encryption	22
FTP	File Transfer Protocol	60
GAN	Generative Adversarial Networks	91
GSAKA	Group Security Authentication and Key Agreement	36
HAIPE	High Assurance Internet Protocol Encryptor	30
HSM	Hardware Security Module	22
HTTP	HyperText Transfer Protocol	16
HTTPS	HTTP Secure	31
IoT	Internet of Things	15
IAM	Identity and Access Management	22
JSON	Javascript Object Notation	16
JWT	JSON Web Token	16
JWTAMH	JWT-based Authentication Mechanism for Hadoop	33
LD	Linked Data	54
LURK	Limited Use of Remote Keys	34
MFA	Multi-Factor Authentication	22
ML	Machine Learning	30
mTLS	mutual TLS	16
NFT	Non-Fungible Token	51
NGSI	Next Generation Service Interface	54
NIST	National Institute of Standards and Technology	22
ODHKE	Optimized Diffie-Hellman Key Exchange	36
OTP	One Time Password	29
OTT	One Time Token	51
OWASP	Open Web Application Security Project	32
PASETO	Platform Agnostic Security Tokens	32
PIV	Personal Identity Verification	35

PKI	Public Key Infrastructure	30
QKD	Quantum Key Distribution	92
QOTP	Quantum One-Time Passwords	34
REST	Representational State Transfer	32
RSA	Rivest–Shamir–Adleman	16
SDK	Software Development Kit	16
SDN	Software-Defined Networks	20
SDP	Software-Defined Perimeters	22
SCP	Secure Copy Protocol	62
SMTP	Simple Mail Transfer Protocol	60
SFTP	Secure Shell File Transfer Protocol	62
STCA	Stacked Token-based Continuous Authentication	29
S/MIME	Secure/Multipurpose Internet Mail Extensions	31
TLS	Transport Layer Security	16
TOTP	Time-based One-Time Passwords	16
UDP	User Datagram Protocol	37
TCP	Transport Control Protocol	37
URL	Uniform Resource Locator	42
WSN	Wireless Sensors Networks	36
XAI	Explainable AI	91
YAML	Yet Ain't Markup Language	41
ZKP	Zero-Knowledge Proofs	29
ZTNA	Zero Trust Networking Architecture	15

1. Εισαγωγή

Στο σύγχρονο ψηφιακό περιβάλλον, όπου η ασφάλεια των δικτυακών υποδομών/υπηρεσιών αποτελεί κρίσιμη προτεραιότητα, η παρούσα Διπλωματική Εργασία εστιάζει στην Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης (Zero Trust Networking Architecture - ZTNA) μέσω της πλατφόρμας ανοικτού κώδικα OpenZiti. Η αυξανόμενη εμφάνιση δικτυακών απειλών, σε συνδυασμό με την εξάπλωση της χρήσης εξ αποστάσεως εργασίας και υβριδικών υποδομών, έχει καταστήσει τα παραδοσιακά μοντέλα ασφάλειας που βασίζονται στην έννοια της "λογικής περιμέτρου" αναποτελεσματικά. Η Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης αποτελεί μια καινοτόμο προσέγγιση που αμφισβητεί κάθε λογική δεδομένης εμπιστοσύνης, απαιτώντας συνεχή ταυτοποίηση και εξουσιοδότηση των χρηστών/εφαρμογών/συσκευών.

1.1 Το Πλαίσιο του Προβλήματος

Όπως προαναφέρθηκε συνοπτικα, οι συμβατικές τεχνικές ασφάλειας, όπως τα τείχη προστασίας (firewalls) και οι λογικές ζώνες εμπιστοσύνης, βασίζονται στην υπόθεση ότι οι εσωτερικοί κόμβοι ενός δικτύου θεωρούνται εκ προοιμίου αξιόπιστοι. Αυτή η στατική προσέγγιση αποδεικνύεται ανεπαρκής σε περιβάλλοντα όπου οι χρήστες, οι συσκευές και οι εφαρμογές βρίσκονται διασκορπισμένοι γεωγραφικά ή λειτουργούν σε υβριδικά νέφη. Εξίσου ανεπαρκείς εμφανίζονται παρόμοιες προσεγγίσεις και απέναντι σε σύγχρονες επιθέσεις, οι οποίες εκμεταλλεύονται ευάλωτα σημεία εντός των δικτύων ή χρησιμοποιούν τεχνικές όπως η "πλευρική κίνηση" (lateral movement), αφότου διαπεράσουν τους υπάρχοντες μηχανισμούς ασφαλείας. Η έλλειψη συνεχούς επαλήθευσης και η χρήση μοναδικών διαπιστευτηρίων ταυτοποίησης δημιουργούν ευπάθειες που μπορούν δυνητικά να αποτελέσουν τρωτά σημεία των δικτυακών υποδομών, εκμεταλλεύσιμα από επιτιθέμενους. Επιπλέον, η διεύρυνση της χρήσης συσκευών Διαδικτύου των Πραγμάτων (Internet of Things - IoT) έχει δημιουργήσει ένα δυναμικά μεταβαλλόμενο, κλιμακούμενο και εξαιρετικά πολύπλοκο τοπίο απειλών. Η ανάγκη για μια δυναμική και πιο αυστηρή προσέγγιση οδήγησε στην ανάπτυξη της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης (ZTNA). Σε αυτό το πλαίσιο, η Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης εισάγει μια ριζικά διαφορετική φιλοσοφία που μπορεί να συμπυκνωθεί στη φράση: "Ποτέ μην εμπιστεύεσαι, πάντα επαλήθευε" ("Never trust, always verify"), εφαρμόζοντας αυστηρούς ελέγχους πρόσβασης και κρυπτογράφηση για κάθε σύνδεση, ανεξαρτήτως της προέλευσης της.

Η Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης καλύπτει τις αδυναμίες των παραδοσιακών Αρχιτεκτονικών Ασφάλειας που περιγράφηκαν υιοθετώντας ορισμένες βασικές αρχές όπως:

1. Κατάτμηση Πόρων (Micro-Segmentation): Διαχωρισμός δεδομένων και υπηρεσιών σε διακριτές λογικές οντότητες με ανεξάρτητες μεταξύ τους πολιτικές και ελέγχους πρόσβασης.
2. Αρχή Ελάχιστων Προνομίων (Least-Privilege Access): Πρόσβαση μόνο στους απαραίτητους πόρους για την εκτέλεση των καθηκόντων κάθε χρήστη/εφαρμογής/συσκευής.
3. Κρυπτογράφηση από Άκρο σε Άκρο (End-to-End Encryption): Προστασία δεδομένων κατά τη μεταφορά.
4. Συνεχής Ταυτοποίηση (Continuous Authentication): Δυναμική και συνεχής ταυτοποίηση χρηστών και συσκευών.

Ωστόσο, η υλοποίηση αυτών των αρχών συχνά συναντά πρακτικές προκλήσεις, όπως η αύξηση της πολυπλοκότητας στη διαχείριση και η πιθανή επιβάρυνση των επιδόσεων των εφαρμογών. Αυτές οι παράμετροι δημιουργούν την ανάγκη για πειραματική αξιολόγηση των συμβιβασμών μεταξύ ασφαλείας και επιδόσεων.

1.2 Στόχοι και Προσέγγιση της παρούσας Διπλωματικής Εργασίας

Η παρούσα Διπλωματική Εργασία επιχειρεί να διερευνήσει τις δυνατότητες εφαρμογής της Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης μέσω της πλατφόρμας ανοιχτού κώδικα OpenZiti. Με αυτό τον τρόπο εξετάζεται η δυνατότητα της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης να ενισχύει την ασφάλεια των δικτυακών επικοινωνιών, χωρίς να επιδρά σημαντικά στις επιδόσεις των εφαρμογών. Κύριους στόχους της παρούσας Διπλωματικής Εργασίας αποτελούν:

1. Η ανάλυση των θεωρητικών θεμελίων της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης, με έμφαση στους μηχανισμούς συνεχούς ταυτοποίησης και κρυπτογράφησης, όπως τα ψηφιακά πιστοποιητικά X.509, τα Javascript Object Notation Web Tokens (JWTs), οι κωδικοί μιας χρήσης Time-based One-Time Passwords (TOTP) και το πρωτόκολλο ασφαλών συνδέσεων Transport Layer Security (TLS), με έμφαση στην αμφίδρομη εκδοχή του (mutual TLS - mTLS).
2. Η μελέτη της πλατφόρμας ανοικτού κώδικα OpenZiti ως εργαλείου εφαρμογής Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης.
3. Η ανάπτυξη πειραματικής εφαρμογής δικτυακής επικοινωνίας με δυνατότητα ενσωμάτωσης σε τοπολογίες δικτύων Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης.
4. Η μελέτη της επίδρασης της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης στις επιδόσεις των εφαρμογών.

1.3 Μεθοδολογία και Υλοποίηση

Για την επίτευξη των στόχων αυτών, πραγματοποιείται παρουσίαση των κύριων μηχανισμών εφαρμογής Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης, συμπεριλαμβανομένων των ψηφιακών πιστοποιητικών X.509, των JSON Web Tokens (JWTs) και των Time-based One-Time Passwords (TOTP). Παράλληλα, αναπτύχθηκε εφαρμογή γραμμής εντολών σε γλώσσα C που αξιοποιεί τη βιβλιοθήκη libcurl του cURL project σε συνδυασμό με το Software Development Kit (SDK) της πλατφόρμας OpenZiti. Η εφαρμογή επιτρέπει την αποστολή αιτημάτων Hyper Text Transfer Protocol (HTTP) μέσω δικτύων που υιοθετούν την Αρχιτεκτονική Μηδενικής Εμπιστοσύνης, ενώ επιτυγχάνει ασφαλή επικοινωνία μέσω πρωτοκόλλων όπως το mTLS και αλγορίθμων για τη δημιουργία ασφαλών κρυπτογραφημένων συνδέσεων όπως ο Rivest–Shamir–Adleman (RSA).

Για την αξιολόγηση της επίδρασης της Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης στις επιδόσεις των δικτυακών εφαρμογών, πραγματοποιήθηκε συγκριτική μελέτη μετρήσεων καθυστέρησης μετ΄επιστροφής (Round-Trip) σε αιτήματα HTTP που αποστέλλονταν προς την εφαρμογή διαχείρισης περιεχομένου βάσεων δεδομένων FIWARE Orion Context Broker. Οι μετρήσεις έλαβαν υπ’ όψιν συνδέσεις με χρήση TLS (industry standard) ως πρωτόκολλο ασφάλειας στρώματος μεταφοράς και συνδέσεις που χρησιμοποιούν mTLS για σύνδεση με δίκτυα OpenZiti, ενώ τα αιτήματα HTTP των οποίων οι καθυστερήσεις συγκρίθηκαν είχαν κλιμακούμενο μέγεθος δεδομένων (payload).

1.4 Δομή της Διπλωματικής Εργασίας

Στο κεφάλαιο 2 πραγματοποιείται η παρουσίαση των παραδοσιακών Αρχιτεκτονικών Ασφαλείας δικτύων (Access Control Lists, Zone-Based Αρχιτεκτονική κλπ.), τα όρια/περιορισμοί που συναντά η εφαρμογή τους, καθώς και η Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης που επιδιώκει να προσφέρει ένα πλαίσιο ασφαλών δικτυακών επικοινωνιών σε αντιδιαστολή με τις ελλείψεις των παραπάνω προσεγγίσεων.

Στο κεφάλαιο 3 παρουσιάζονται οι θεμελιώδεις αρχές της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης, με έμφαση σε έννοιες όπως η Αρχή Ελάχιστων Προνομίων (Least-Privilege Access), η Κατάτμηση Πόρων (Resource Segmentation) και η Συνεχής Ταυτοποίηση (Continual Authentication).

Στο κεφάλαιο 4 δίνεται ιδιαίτερη έμφαση στους μηχανισμούς που χρησιμοποιούνται για την εφαρμογή της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης, όπως τα ψηφιακά πιστοποιητικά X.509, τα JSON Web Tokens (JWTs), οι κωδικοί μιας χρήσης (TOTP) και το πρωτόκολλο ασφαλών συνδέσεων TLS, με έμφαση στην αμφίδρομη εκδοχή του (mTLS), αλλά και τις σύγχρονες τάσεις στη χρήση και τις βελτιστοποιήσεις τους για ποικίλες διαφορετικές εφαρμογές.

Το κεφάλαιο 5 αφιερώνεται στην παρουσίαση της πλατφόρμας Δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti, εξετάζοντας τη δομή της, τις λογικές οντότητες που χρησιμοποιεί για την εφαρμογή της Αρχιτεκτονικής, τα στοιχεία λογισμικού (όπως οι δρομολογητές και οι ελεγκτές), αλλά και τις λειτουργίες της, όπως η ταυτοποίηση χρηστών/εφαρμογών, τα πρωτόκολλα ασφαλών κρυπτογραφημένων συνδέσεων που χρησιμοποιεί και συνολικά τη διαδικασία πρόσβασης σε υπηρεσίες μέσω της πλατφόρμας.

Στο κεφάλαιο 6 παρουσιάζεται η πρακτική υλοποίηση που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας, παρουσιάζοντας τη σχεδίαση και υλοποίηση της εφαρμογής γραμμής εντολών και τα εργαλεία λογισμικού που χρησιμοποιήθηκαν. Αναλύονται οι τεχνικές λεπτομέρειες της ενσωμάτωσης της βιβλιοθήκης libcurl με τα εργαλεία ανάπτυξης λογισμικού SDK της πλατφόρμας OpenZiti.

Στο κεφάλαιο 7 εξετάζεται η διαφορά στις επιδόσεις των εφαρμογών κατά την ενσωμάτωσή τους σε τοπολογίες δικτύου που υιοθετούν την Αρχιτεκτονική Μηδενικής Εμπιστοσύνης και δικτύων που χρησιμοποιούν τις παραδοσιακές μεθόδους ασφαλούς επικοινωνίας και TLS. Δόθηκε έμφαση στις μετρήσεις καθυστέρησης μετ' επιστροφής (Round-Trip) υπό διαφορετικά σενάρια αιτημάτων HTTP. Οι μετρήσεις αφορούν τόσο απλά αιτήματα healthcheck όσο και σενάρια ανάκτησης οντοτήτων μεγάλου όγκου δεδομένων με έως και 10^4 χαρακτηριστικά (attributes). Τα αποτελέσματα της μελέτης παρουσιάζονται αναλυτικά και προσφέρουν μια αντιπροσωπευτική εικόνα και χρήσιμα συμπεράσματα αναφορικά με το κόστος της υιοθέτησης Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης σε σχέση με τις επιδόσεις των εφαρμογών.

Τέλος στο κεφάλαιο 8 συνάγονται γενικότερα συμπεράσματα για τη βιωσιμότητα της εφαρμογής Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης. Παράλληλα σκιαγραφούνται πιθανές μελλοντικές ερευνητικές προεκτάσεις που θα μπορούσαν να προκύψουν με εφαλτήριο την παρούσα Διπλωματική Εργασία, τόσο αναφορικά με την κλιμακωσιμότητα της εφαρμογής Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης, το πλαίσιο εφαρμογής της λαμβάνοντας υπ’ όψιν την εξέλιξη των κβαντικών υπολογιστών, πρακτικές

επιπτώσεις της εφαρμογής της Αρχιτεκτονικής, αλλά και ορισμένες παρατηρήσεις για τη βελτιστοποίηση των υλοποιήσεων.

1.5 Σημασία και Πιθανές Εφαρμογές

Η παρούσα Διπλωματική Εργασία φιλοδοξεί να συνεισφέρει στη βαθύτερη κατανόηση των πρακτικών επιπτώσεων της εφαρμογής Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης σε πραγματικά δικτυακά περιβάλλοντα. Οι πειραματικές μετρήσεις και τα συμπεράσματα που παρουσιάζονται προσφέρουν ποσοτικά δεδομένα που μπορούν δυνητικά να ληφθούν υπ' όψιν κατά τον σχεδιασμό ασφαλών δικτύων και εφαρμογών σε τομείς όπως:

- Υποδομές Κρίσιμων Συστημάτων (π.χ. ενέργειας, υγειονομικής περίθαλψης)
- Συστήματα IoT και Βιομηχανίας 4.0
- Πλατφόρμες Υπηρεσιών Νέφους (Cloud Services)
- Απομακρυσμένη Πρόσβαση σε Εταιρικούς Πόρους

Με την ανάπτυξη της πειραματικής εφαρμογής, αποδεικνύεται η δυνατότητα ενσωμάτωσης των αρχών της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης σε υπάρχουσες δικτυακές εφαρμογές, με διαχειρίσιμο αντίκτυπο στις επιδόσεις, προσφέροντας ένα πλαίσιο αναφοράς για μελλοντικές εφαρμογές που στοχεύουν στην εξάλειψη της έμφυτης εμπιστοσύνης σε κόμβους δικτύων. Τα ευρήματα συντείνουν προς την υιοθέτηση Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης για την ανάπτυξη ασφαλών εφαρμογών, αφού δεν παρατηρείται αξιοσημείωτη επίδραση στις επιδόσεις των εφαρμογών. Συνοπτικά, η παρούσα Διπλωματική Εργασία φιλοδοξεί να αποτελέσει ένα βήμα προς τη διεύρυνση της ενσωμάτωσης της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης στις δικτυακές επικοινωνίες, παρουσιάζοντας ταυτόχρονα τη θεωρητική της βάση και τις πρακτικές της δυνατότητες.

2. Αρχιτεκτονικές Ασφάλειας δικτύων Υπολογιστών

2.1 Λίστες Ελέγχου Πρόσβασης (Access Control Lists)

Η αρχιτεκτονική ασφαλείας δικτύων που βασίζεται σε Λίστες Ελέγχου Πρόσβασης (Access Control Lists – ACL) αποτελεί μία από τις πιο παραδοσιακές και ευρείας χρήσης στρατηγικές για τον έλεγχο της ροής δεδομένων και την προστασία της πρόσβασης σε δικτυακούς πόρους. Οι Λίστες Ελέγχου Πρόσβασης (ACLs) λειτουργούν ως σύνολα από κανόνες που ορίζουν τα κριτήρια βάσει των οποίων πακέτα δικτύου επιτρέπεται ή όχι, να προωθούνται από κόμβους δικτύου, ή ποια μεταχείριση πρέπει να έχουν. Σε αυτά τα πλαίσια, οι διαχειριστές δικτύου (network administrators) μπορούν να εφαρμόσουν πολιτικές ελέγχου πρόσβασης (access control policies), με βάση κριτήρια όπως η διεύθυνση πηγής (source address), η διεύθυνση προορισμού (destination address), το πρωτόκολλο που χρησιμοποιείται, ο αριθμός θύρας (port number), και άλλα πεδία στις επικεφαλίδες των πακέτων.

Σε αντίθεση με πιο σύγχρονες αρχιτεκτονικές ασφάλειας δικτύων, όπως η Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης (ZTNA) ή η δυναμική αξιολόγηση κινδύνου (dynamic risk assessment), οι Λίστες Ελέγχου Πρόσβασης (Access Control Lists) συχνά σχεδιάζονται και υλοποιούνται ως στατικοί κανόνες, που μεταβάλλονται σπανίως, παραδείγματος χάρη σε περίπτωση που αποφασιστεί μια αλλαγή στην πολιτική του εκάστοτε οργανισμού. Η στατικότητα των Λιστών Ελέγχου Πρόσβασης (ACLs) τους προσδίδει απλότητα και σαφήνεια, ωστόσο έχει ως αποτέλεσμα μειωμένη προσαρμοστικότητα για την αντιμετώπιση νέων επιθέσεων ή στην προσαρμογή του ελέγχου πρόσβασης σε πραγματικό χρόνο. Η ευκολία ρύθμισης καθώς και η δυνατότητα ενσωμάτωσης σε ποικιλία εργαλείων δικτύου όπως δρομολογητές (routers), μεταγωγείς (switches) αλλά και τείχη προστασίας (firewalls), έχουν ως συνέπεια οι Λίστες Ελέγχου Πρόσβασης να συναντώνται σε πλείστα περιβάλλοντα δικτύων — όπως εταιρικά τοπικά δίκτυα (Local Area Networks) ή δίκτυα ευρείας ζώνης (Wide Area Networks).

Η ενσωμάτωση των Λιστών Ελέγχου Πρόσβασης μπορεί να πραγματοποιηθεί σε μια σειρά από διαφορετικά επίπεδα. Λίστες Ελέγχου Πρόσβασης έχουν τη δυνατότητα να προωθούν ή να απορρίπτουν πακέτα, μη επιτρέποντας έτσι την αποστολή τους σε διαφορετικούς τομείς δικτύου, παίζοντας έτσι το ρόλο ενός φίλτρου, με σκοπό την εφαρμογή πολιτικής πρόσβασης σε επίπεδο δρομολογητή (router). Επιπρόσθετα stateful monitoring και άλλοι μηχανισμοί είναι δυνατόν να εφαρμοστούν συνδυαστικά με Λίστες Ελέγχου Πρόσβασης σε επίπεδο τείχους προστασίας (firewall), ενώ η κυκλοφορία πακέτων (network traffic) ανάμεσα σε διαφορετικά εικονικά τοπικά δίκτυα (Virtual Local Area Networks) μπορεί να οριοθετηθεί με εφαρμογή Λιστών Ελέγχου Πρόσβασης σε επίπεδο μεταγωγέα (switch).

Αξίζει να σημειωθεί, πως υφίστανται και μειονεκτήματα στις Λίστες Ελέγχου Πρόσβασης (ACLs), παρά την απλή τους ρύθμιση και ευκολία ενσωμάτωσης που τις χαρακτηρίζει. Η αναγκαιότητα επαλήθευσης της εύρυθμης και αποτελεσματικής λειτουργίας των Λιστών Ελέγχου Πρόσβασης από διαχειριστές δικτύου, μπορεί να αποδειχθεί απαιτητική διαδικασία, ιδιαιτέρως σε σενάρια που πιθανώς να περιλαμβάνουν περίπλοκες Λίστες Ελέγχου Πρόσβασης, με πολυάριθμους κανόνες. Επιπλέον, η στατική

φύση των ACLs σημαίνει ότι αυτές δεν προσαρμόζονται εύκολα σε δυναμικές αλλαγές του τοπίου δικτυακών απειλών, απαιτώντας έτσι συχνή αναθεώρηση και τροποποίηση των κανόνων. Η συμπερίληψη πιο εκλεπτυσμένων μηχανισμών ασφαλείας όπως οι διαδικασίες δυναμικής αξιολόγησης κινδύνων (dynamic risk assessment), η κρυπτογράφηση των δεδομένων ή η ταυτοποίηση (authentication) των χρηστών, κρίνεται αναγκαία, πέραν της εφαρμογής των Λιστών Ελέγχου Πρόσβασης, αφού οι ACLs εξαρτώνται από χαρακτηριστικά πακέτων δικτύου, εύκολα ανιχνεύσιμα και ευάλωτα.

Αυξημένα επίπεδα ασφάλειας και βελτιστοποίηση της χρήσης, μπορεί να προσφέρει ο συνδυασμός των Λιστών Πρόσβασης Ελέγχου με άλλες αρχιτεκτονικές και εργαλεία δικτύων, όπως τα Software-Defined Networks (SDN). Η υλοποίηση και η δυναμική ενημέρωση σύνθετων πολιτικών πρόσβασης μπορεί παραδείγματος χάρη να εφαρμοστεί με τη συγκεντρωτική διαχείριση Λιστών Ελέγχου Πρόσβασης από τον ελεγκτή (controller) ενός SDN. Έτσι η χρήση και ενσωμάτωση των Λιστών Ελέγχου Πρόσβασης (ACLs) είναι δυνατόν να παραμετροποιείται και να προσαρμόζεται στις ανάγκες και τις απαιτήσεις του σύγχρονου και ευμετάβλητου σκηνικού της ασφάλειας δικτύων.

2.2 Αρχιτεκτονική Ασφάλειας δικτύων Υπολογιστών βασισμένη σε λογικές Ζώνες

Η Αρχιτεκτονική Ασφάλειας δικτύων βασισμένη σε Ζώνες (Zone-Based Network Security Architecture) εισάγει μια πολυεπίπεδη προσέγγιση στην προστασία δικτύων, οργανώνοντας τους πόρους σε λογικές Ζώνες όπου κάθε ζώνη χαρακτηρίζεται από συγκεκριμένο επίπεδο εμπιστοσύνης, πολιτικές πρόσβασης και μηχανισμούς ελέγχου. Σε αντίθεση με την παραδοσιακή αρχιτεκτονική ασφάλειας που βασίζεται σε μία λογική περίμετρο, η οποία προϋποθέτει ότι το εσωτερικό του δικτύου είναι αξιόπιστο, η προσέγγιση που βασίζεται σε λογικές Ζώνες χωρίζει το δίκτυο σε τομείς, επιτρέποντας την απομόνωση κρίσιμων πόρων και την αυστηροποίηση των πολιτικών ασφαλείας σε κάθε διακριτή περιοχή. Το αποτέλεσμα είναι ένας πιο ευέλικτος σχεδιασμός, που υποστηρίζει εύκολες αναβαθμίσεις, αλλαγές στη ροή δεδομένων, και δυναμική προσαρμογή στις απαιτήσεις ασφάλειας.

Τα κύρια στοιχεία μιας Αρχιτεκτονικής Ασφάλειας δικτύων που βασίζεται σε λογικές Ζώνες περιλαμβάνουν: (α) τον ορισμό των ζωνών, όπου η κάθε ζώνη αντιπροσωπεύει ένα επίπεδο εμπιστοσύνης ή μια λειτουργική περιοχή του δικτύου (π.χ. Demilitarized Zone - DMZ -, εσωτερικό δίκτυο, δίκτυο διαχείρισης, δίκτυο επισκεπτών), (β) τη δημιουργία πολιτικών που καθορίζουν ποια είδη κυκλοφορίας επιτρέπονται ή απαγορεύονται μεταξύ των ζωνών, και (γ) την επιλογή κατάλληλων μηχανισμών ελέγχου (firewalls, Intrusion Preventions/Detection Systems, διακομιστές μεσολάβησης) για την επιβολή αυτών των πολιτικών. Με αυτόν τον τρόπο, καθίσταται δυνατή η εύκολη εφαρμογή της αρχής Ελάχιστων Προνομίων (least privilege) και η αποφυγή φαινομένων “πλευρικής” κίνησης επιτιθέμενων (lateral movement) εντός του δικτύου.

Η δυνατότητα εφαρμογής Αρχιτεκτονικής Ασφάλειας δικτύων βασισμένης σε λογικές Ζώνες στα αναδυόμενα οικοσυστήματα όπως το Διαδίκτυο των Πραγμάτων (IoT) και η Βιομηχανία 4.0 επιτρέπει την απομόνωση ευάλωτων συστημάτων, την τοποθέτηση κρίσιμων υπηρεσιών σε πιο αυστηρά ελεγχόμενα περιβάλλοντα, καθώς και τη θέσπιση αποδοτικών πολιτικών επικοινωνίας. Η προσέγγιση της Αρχιτεκτονικής ασφαλείας με λογικές Ζώνες (Zone based) μειώνει δραστικά τα δυνητικά ευάλωτα σημεία ενός δικτύου (attack surface), αφού οι επιτιθέμενοι δεν έχουν την ευχέρεια να αποκτήσουν ελεύθερη

πρόσβαση σε πόρους εντός του δικτύου ακόμα και αν αποκτήσουν αρχική πρόσβαση σε κάποια ζώνη χαμηλότερης εμπιστοσύνης.

Παράλληλα, η Αρχιτεκτονική Ασφάλειας που βασίζεται σε λογικές ζώνες είναι συμβατή και με πιο σύγχρονες προσεγγίσεις, όπως η Αρχιτεκτονική Μηδενικής Εμπιστοσύνης, αφού διευκολύνει την εισαγωγή κατάτμησης των πόρων (micro-segmentation) και την εφαρμογή δυναμικών πολιτικών ελέγχου πρόσβασης. Ενώ σε ένα μοντέλο Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης, η εμπιστοσύνη σε χρήστες/εφαρμογές/συσκευές αμφισβητείται διαρκώς, η προσέγγιση της Αρχιτεκτονική ασφαλείας με λογικές Ζώνες (Zone based) μπορεί να προσφέρει μια δομική βάση πάνω στην οποία να εφαρμόζονται οι κατάλληλοι έλεγχοι ταυτότητας, κρυπτογράφησης και ανάλυσης συμπεριφοράς. Επιπλέον, σε συνδυασμό με τεχνικές SDN, η Αρχιτεκτονική Ασφάλειας Δικτύων που βασίζεται σε λογικές ζώνες αποκτά δυναμικό χαρακτήρα, επιτρέποντας πιο ευέλικτο επανακαθορισμό ζωνών, πολιτικών και ροών δεδομένων σε πραγματικό χρόνο.

2.3 Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης

Η έννοια της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης αποτελεί μια καινοτόμο προσέγγιση στην ασφάλεια δικτύων, η οποία τα τελευταία χρόνια έχει προσελκύσει αυξημένο ενδιαφέρον από την ακαδημαϊκή και ερευνητική κοινότητα, καθώς και από τη βιομηχανία της κυβερνοασφάλειας. Βασίζεται στην αρχή της μη εμπιστοσύνης σε κανέναν κόμβο, χρήστη ή συσκευή εντός ή εκτός του δικτύου, μέχρις ότου επαληθευθούν με αυστηρές διαδικασίες τα δικαιώματα και οι πολιτικές πρόσβασης. Με άλλα λόγια, η παραδοσιακή υπόθεση ότι το εσωτερικό δίκτυο ενός οργανισμού είναι εγγενώς αξιόπιστο εγκαταλείπεται υπέρ μιας αρχιτεκτονικής εντελώς απαλλαγμένης από τέτοιες εικασίες, όπου η επαλήθευση απαιτείται συνεχώς, ανεξαρτήτως της θέσης ή του ρόλου ενός χρήστη ή συστήματος [1].

Σύμφωνα με τη διεθνή βιβλιογραφία, η Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης συνδέεται στενά με την αρχή της ελαχιστοποίησης των προνομίων πρόσβασης (least privilege access) και τη δυναμική πιστοποίηση ταυτότητας χρηστών, συσκευών, εφαρμογών και υπηρεσιών, ανεξαρτήτως της γεωγραφικής τους θέσης. Η θεώρηση αυτή συνεπάγεται την κατάργηση ή τον δραστικό περιορισμό των παραδοσιακών συστημάτων ασφάλειας που βασίζονται στη λογική "περίμετρο" (perimeter-based security, π.χ. firewalls) και την αντικατάστασή τους από ένα κατανεμημένο σύστημα ελέγχου πρόσβασης σε πραγματικό χρόνο, το οποίο βασίζεται σε συνεχείς ελέγχους ταυτότητας (continuous authentication) και εξουσιοδότησης (authorization) [2].

Σε αρκετές ακαδημαϊκές μελέτες, τα επιμέρους στοιχεία της αρχιτεκτονικής Μηδενικής Εμπιστοσύνης έχουν μελετηθεί ως προς την ικανότητά τους να εντοπίζουν, να αποτρέπουν και να επιβραδύνουν την "πλευρική" κίνηση επιτιθέμενων (lateral movement) εντός των δικτύων, ένα από τα κρισιμότερα προβλήματα στα σύγχρονα εταιρικά περιβάλλοντα. Η εφαρμογή της αρχής κατάτμησης των πόρων (micro-segmentation), όπως περιγράφεται σε αναγνωρισμένα συγγράμματα, συμβάλλει στην απομόνωση κρίσιμων πόρων, επιτρέποντας πρόσβαση μόνο σε εκείνους που διαθέτουν τα κατάλληλα διαπιστευτήρια.

Ακαδημαϊκά άρθρα εξετάζουν τους μηχανισμούς στους οποίους βασίζεται η Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης, όπως αλγόριθμοι για δυναμική αξιολόγηση κινδύνου, context-aware authentication, μηχανισμοί μηχανικής μάθησης για εντοπισμό ανωμαλιών, καθώς και πρωτόκολλα ασφαλούς επικοινωνίας που συνδυάζουν κρυπτογράφηση από άκρο σε άκρο (End-to-End encryption - E2EE) με συνεχή αξιολόγηση εμπιστοσύνης. Υλοποιήσεις Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης τείνουν να ενσωματώνουν λύσεις όπως Software-Defined Perimeters (SDP), που ορίζονται από λογισμικό και προσφέρουν λεπτομερή έλεγχο πρόσβασης. Στη βιβλιογραφία συχνά συνδυάζονται SDP με σύγχρονες πλατφόρμες διαχείρισης ταυτοτήτων και πρόσβασης (Identity and Access Management IAM) και τεχνικές πολυπαραγοντικής ταυτοποίησης (Multi-Factor Authentication MFA), ενώ υπάρχει και η δυνατότητα να ενισχύονται από Hardware Security Modules (HSMs) και tokenization.

Σημαντική είναι και η ερευνητική έμφαση στη συμμόρφωση με διεθνή πρότυπα και νομικά πλαίσια. Οι μελέτες του National Institute of Science and Technology (NIST) τεκμηριώνουν βέλτιστες πρακτικές για τον σχεδιασμό και την υλοποίηση της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης, καλύπτοντας ζητήματα όπως η διαχείριση κρυπτογραφικών κλειδιών, η παρακολούθηση και ανάλυση δικτυακής κίνησης καθώς και η διασύνδεση με περιβάλλοντα νέφους (cloud). Σε αυτό το πλαίσιο, η υιοθέτηση της φιλοσοφίας μηδενικής εμπιστοσύνης αποτελεί μια διαδικασία διαρκούς βελτίωσης, όπου οι οργανισμοί οφείλουν να επανεξετάζουν τακτικά τις πολιτικές πρόσβασης τους και να προσαρμόζουν τις αμυντικές τακτικές τους στο συνεχώς εξελισσόμενο τοπίο των κυβερνοεπιθέσεων.

Επιπλέον, έρευνες υπογραμμίζουν ότι η επιτυχής εφαρμογή της αρχιτεκτονικής ασφάλειας μηδενικής εμπιστοσύνης εξαρτάται και από την οργανωτική αντίληψη, τις διαδικασίες λήψης αποφάσεων καθώς και την κατάλληλη εκπαίδευση του προσωπικού. Παράλληλα επισημαίνεται η ανάγκη επανασχεδιασμού υφιστάμενων υποδομών, αντιμετώπισης αντιστάσεων στην αλλαγή και την ενοποίηση ετερογενών συστημάτων ταυτότητας και ελέγχου πρόσβασης. Τεχνικές συνεχούς προσαρμοζόμενης αξιολόγησης κινδύνου και εμπιστοσύνης (Continuous Adaptive Risk and Trust Assessment - CARTA) βοηθούν στην αυτόματη προσαρμογή των πολιτικών πρόσβασης βάσει του επιπέδου κινδύνου και των χαρακτηριστικών συμπεριφοράς χρηστών και συσκευών στο δίκτυο.

Τέλος, η ακαδημαϊκή έρευνα τονίζει πως η Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης δεν αποτελεί ένα στατικό προορισμό αλλά μια δυναμική διαδικασία. Πανεπιστημιακά εργαστήρια, ερευνητικά ινστιτούτα και ομάδες τυποποίησης συνεχίζουν να εξετάζουν νέες τεχνολογίες (όπως zero-knowledge proofs, confidential computing, homomorphic encryption), ώστε να θεμελιωθούν περαιτέρω τα επίπεδα εμπιστοσύνης και ασφάλειας για τα μελλοντικά ψηφιακά οικοσυστήματα. Η προσέγγιση αυτή προάγει ένα συνεχές πνεύμα καινοτομίας και προσαρμογής στις ολοένα αυξανόμενες και πολύπλοκες απειλές του διαδικτύου [3].

3. Θεμελιώδεις Αρχές της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης

3.1 Αρχή Ελάχιστων Προνομίων (Least-Privilege Access)

Μια από τις βασικές αρχές της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης είναι η αρχή Ελάχιστων Προνομίων (Least-Privilege Access). Σύμφωνα με την αρχή Ελάχιστων Προνομίων, σε κάθε χρήστη πρέπει να εκχωρούνται αποκλειστικά οι πόροι που είναι απολύτως απαραίτητοι για την εκτέλεση των καθηκόντων του, χωρίς να εκχωρούνται επιπλέον δικαιώματα ή πρόσβαση [4][5]. Η εφαρμογή της αρχής Ελάχιστων Προνομίων (Least-Privilege Access) προϋποθέτει συνεχή αξιολόγηση των δικαιωμάτων πρόσβασης των χρηστών, αλλά και συνεχή ταυτοποίηση τους. Επίσης, όπως έχει προαναφερθεί στην παρούσα εργασία, οι επιπτώσεις των παραβιάσεων σε ένα δίκτυο, συρρικνώνονται δραματικά όταν εφαρμόζοντας την αρχή Ελάχιστων Προνομίων, εκχωρούνται δικαιώματα πρόσβασης μόνο στα απολύτως απαραίτητα δεδομένα και υπηρεσίες του δικτύου.

Παράλληλα, η συμβατότητα μιας εφαρμογής πελάτη (client) με τα απαιτούμενα πρωτόκολλα ασφαλείας και η δικτυακή δραστηριότητα του χρήστη, αποτελεί έναν από τους παράγοντες που συνθέτουν το πλαίσιο συνεχούς ανάλυσης κινδύνου (Continuous Adaptive Risk Assessment), απαραίτητο για την επιτυχή εφαρμογή της αρχής Ελάχιστων Προνομίων (Least-Privilege Access) σε σύγχρονα περιβάλλοντα. Αξίζει να σημειωθεί ότι ο όσο το δυνατόν ευρύτερος κατακερματισμός και διαχωρισμός των διαθέσιμων δεδομένων και υπηρεσιών (micro-segmentation), αποτελεί τη βάση πάνω στην οποία δομείται πάντα η εφαρμογή της αρχής Ελάχιστων Προνομίων, καθώς είναι αυτός που ορίζει αρχικά το πεδίο αναφοράς της.

Η εφαρμογή της αρχής Ελάχιστων Προνομίων συχνά αποδεικνύεται απαιτητική διαδικασία ιδιαιτέρως σε πολυεπίπεδα εταιρικά δίκτυα με πολλαπλά επίπεδα πρόσβασης. Οι πολιτικές ασφαλείας θα πρέπει να αξιολογούνται και να προσαρμόζονται στις συνεχείς μεταβολές των απαιτήσεων της σύγχρονης δικτύωσης αρκετά τακτικά και κατά περίπτωση. Αντίθετα, ανεπιτυχής εφαρμογή της αρχής Ελάχιστων Προνομίων, είναι αρκετά πιθανό να οδηγήσει σε σημαντικά προβλήματα σχετικά με την εμπειρία χρήσης και τελικά τη συνολική επίδοση ενός δικτύου. Τα συστήματα ασφαλείας κάθε οργανισμού στις περισσότερες περιπτώσεις πρέπει να υποστούν ριζική αναμόρφωση και επαναθεμελίωση ώστε η αρχή Ελάχιστων Προνομίων να είναι δυνατόν να υιοθετηθεί με τρόπο αποτελεσματικό.

Οι δικτυακές απειλές που πηγάζουν από το εσωτερικό της "λογικής περιμέτρου" ενός δικτύου, είναι δυνατόν να περιοριστούν σημαντικά ως αποτέλεσμα της επιτυχημένης υιοθέτησης της αρχής Ελάχιστων Προνομίων, ενώ και η συμμόρφωση με εξωτερικούς κανονισμούς ή Service Level Agreements ανάμεσα σε εταιρίες, συχνά εκπληρώνεται με την εφαρμογή της αρχής. Με την εκχώρηση πρόσβασης μόνο στους απολύτως απαραίτητους πόρους, η αρχή Ελάχιστων Προνομίων κατορθώνει τελικά να μειώσει θεαματικά και τον αριθμό των σημείων πρόσβασης σε ένα δίκτυο, τα οποία μπορούν να καταστούν ευάλωτα σε επιθέσεις. Με αυτόν τον τρόπο η αρχή Ελάχιστων Προνομίων συρρικνώνει δραματικά

την "επιφάνεια" του δικτύου (attack-surface) που είναι δυνητικά ευάλωτη σε επιθέσεις. Η λήψη αποφάσεων σε πραγματικό χρόνο σχετικά με την εφαρμογή της αρχής Ελάχιστων Προνομίων, από σύνθετους αλγορίθμους αξιολόγησης κινδύνου φαίνεται πως θα αποτελέσει από τις κυρίαρχες τάσεις σχετικά με την Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης. Η προστασία των δεδομένων κατά τη μετάδοση τους επίσης αποτελεί πρόκληση που νεότεροι, ανθεκτικοί σε επιθέσεις κβαντικών υπολογιστών, κρυπτογραφικοί αλγόριθμοι θα πρέπει να υπερβούν. Οι πολιτικές ασφαλείας κάθε δικτύου πρέπει, όπως έχει προαναφερθεί, να υπόκεινται τακτικά σε προσαρμογές, οι οποίες θα ήταν δυνατόν να απλουστευτούν για τους διαχειριστές των δικτύων (network administrators) αλλά και να βελτιστοποιηθούν (optimization) συνολικά, με χρήση μηχανικής μάθησης και τεχνητής νοημοσύνης.

Συμπερασματικά, η αρχή Ελάχιστων Προνομίων αποτελεί ακρογωνιαίο λίθο της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης, προσφέροντας ένα ισχυρό πλαίσιο για την αντιμετώπιση των σύγχρονων δικτυακών απειλών. Καθώς τα δικτυακά περιβάλλοντα γίνονται ολοένα και πιο πολύπλοκα, η σημασία αυτής της αρχής της Ελάχιστων Προνομίων αναμένεται να εμφανίζεται συνεχώς αυξανόμενη, οδηγώντας σε πιο προσαρμοστικά και ανθεκτικά συστήματα ασφαλείας.

3.2 Κατάτμηση Πόρων (Resource Segmentation)

Η αρχή (principle) κατάτμησης των πόρων (resource segmentation) ενός δικτύου έχει στον πυρήνα της τη λογική διαχωρισμού των μεθόδων ελέγχου πρόσβασης και των πολιτικών πρόσβασης για κάθε εφαρμογή/υπηρεσία/πόρο του δικτύου, τα οποία ορίζει ως διακριτές μεταξύ τους οντότητες. Η αρχή της κατάτμησης των πόρων αποτελεί ιδιαίτερα σημαντικό δομικό στοιχείο του ευρύτερου και συνεχώς μεταβαλλόμενου σύγχρονου τοπίου της ασφάλειας δικτύων, αλλά και της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης ειδικότερα.

Η λεπτομερής και συνεχής παρακολούθηση της δικτυακής κίνησης (network traffic) ανάμεσα στις εφαρμογές και τους κόμβους ενός δικτύου που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης αποτελεί αναπόσπαστο κομμάτι της διαδικασίας προσαρμογής των πολιτικών πρόσβασης. Συστήματα Ανίχνευσης/Πρόβλεψης Εισβολών (Intrusion Detection/Prevention Systems), η δυναμική τροποποίηση της λειτουργίας των δικτύων μέσω λογισμικού (Software Defined Networking) αλλά και η ανάπτυξη συστημάτων διαχείρισης πρόσβασης ταυτοτήτων (IAM), συνιστούν εξαιρετικά χρήσιμα εργαλεία στη διάθεση των διαχειριστών δικτύων (network administrators) στη διαδικασία εφαρμογής σύνθετων και πολυεπίπεδων πολιτικών πρόσβασης και τελικά και της ενσωμάτωσης της αρχής της Κατάτμησης των Πόρων (resource segmentation principle) στη σχεδιαστική λογική και λειτουργία των δικτύων που υιοθετούν Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης [6].

Η εφαρμογή της αρχής Κατάτμησης των Πόρων (resource segmentation principle) στο άκρως απαιτητικό σκηνικό των σύγχρονων πολυεπίπεδων εταιρικών δικτύων μπορεί να αποδειχθεί αρκετά απαιτητική διαδικασία για τους σχεδιαστές των δικτυακών συστημάτων, όπως καταδυκνύουν και σύγχρονες ακαδημαϊκές εργασίες που θέτουν ως αντικείμενο τους το συγκεκριμένο θέμα. Η συνολικότερη δικτυακή κίνηση ενός χρήστη/εφαρμογής, τα χαρακτηριστικά τόσο του υλικού (hardware) όσο και του λογισμικού (software) όπως π.χ. το είδος ή η έκδοση του λειτουργικού συστήματος, η έκδοση μιας εφαρμογής πελάτη κλπ. αποτελούν παράγοντες που καθορίζουν σε σημαντικό βαθμό ένα ευρύτερο πλαίσιο

πρόσβασης το οποίο πρέπει να συνυπολογίζεται μαζί με την ταυτοποίηση των χρηστών και των εφαρμογών, ώστε η υιοθέτηση της αρχής Κατάτμησης των Πόρων (resource segmentation principle) να εντάσσεται σε ένα ευρύτερο πλαίσιο αποτελεσματικής υλοποίησης της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης. Επιπρόσθετα, καθίσταται ευδιάκριτο ότι διακριτές μεταξύ τους πολιτικές πρόσβασης και ταυτοποίησης θα πρέπει να διαμορφώνονται και να υιοθετούνται για κάθε μια από τις οντότητες ενός δικτύου Μηδενικής Εμπιστοσύνης, ακολουθώντας την αρχή της κατάτμησης των πόρων (resource segmentation). Οι εξαρτήσεις (dependencies) ανάμεσα στους διαφορετικούς πόρους (υπηρεσίες/δεδομένα) ενός δικτύου Μηδενικής Εμπιστοσύνης πρέπει επίσης να αποτελούν αντικείμενο συστηματικής μελέτης από τους διαχειριστές δικτύου (Network Administrators), ώστε να αποφεύγονται οι ανεπιθύμητοι (λόγω ανεπαρκούς σχεδιασμού) αποκλεισμοί πρόσβασης από αυστηρώς ζωτικούς, για τους χρήστες και τις εφαρμογές, πόρους και να διασφαλίζεται κατα τρόπο τέτοιο η εύρυθμη και αποτελεσματική συνολική λειτουργία των δικτύων, συμβάλλοντας έτσι και στην ευρύτερη αποδοτικότητα οργανισμών και εταιρειών [7].

Το ενδεχόμενο “πλευρικής” κίνησης (lateral movement) πιθανών επιτιθέμενων εντός ενός δικτύου, καθίσταται αρκετά δύσκολα υλοποιήσιμο με την εφαρμογή της αρχής Κατάτμησης των Πόρων, ενώ μειώνοντας τους πόρους στους οποίους έχει δικαίωμα πρόσβασης ένας χρήστης, ελαττώνονται θεαματικά και τα δυνητικά ευάλωτα σημεία πρόσβασης σε ένα δίκτυο (attack surface), που θα μπορούσαν πιθανώς να εκμεταλλευτούν κακόβουλοι χρήστες/λογισμικό. Βάσει της αρχής Κατάτμησης των Πόρων (resource segmentation) η ανάπτυξη εφαρμογών (π.χ. βάσεων δεδομένων) σε ίδιο φυσικό ή εικονικό περιβάλλον δεν συνεπάγεται πρακτικά οποιαδήποτε εγγύτητα των οντοτήτων αυτών σχετικά με την υπόσταση τους σε ένα δίκτυο που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης. Παράλληλα, η έννοια της μικρο-κατάτμησης εμβαθύνει περαιτέρω την αρχή Κατάτμησης των Πόρων σε επίπεδο ενδεχομένως ακόμα χαμηλότερο και από το επίπεδο των εφαρμογών, υιοθετώντας διακριτές πολιτικές πρόσβασης και ταυτοποίησης για διαφορετικά δεδομένα (workloads) που μπορεί να διαχειρίζεται ακόμα και η ίδια εφαρμογή [8].

Η γενική εικόνα ενός δικτύου σχετικά με την ασφάλεια είναι δυνατόν να διατηρηθεί ανεπηρέαστη, ανεξάρτητα από την προσθαφαίρεση νέων υπηρεσιών και εφαρμογών, ως αποτέλεσμα της εφαρμογής της αρχής Κατάτμησης Πόρων (resource segmentation principle). Ένα σημαντικό πλεονέκτημα της κατάτμησης εφαρμογών (application segmentation) είναι η βελτιωμένη ορατότητα και έλεγχος της κυκλοφορίας δεδομένων εντός του δικτύου. Έτσι επιτρέπεται η ταχύτερη ανίχνευση και αντιμετώπιση των απειλών καθώς και η πιο αποτελεσματική εφαρμογή των πολιτικών συμμόρφωσης. Επιπλέον, μέσω της κατάτμηση εφαρμογών διευκολύνεται η κλιμάκωση και ευελιξία των συστημάτων, καθώς δίνεται δυνατότητα προσθήκης ή αφαίρεσης εφαρμογών χωρίς να επηρεάζεται η συνολική ασφάλεια του δικτύου.

Ωστόσο, η εφαρμογή της κατάτμησης εφαρμογών αντιμετωπίζει και ορισμένες προκλήσεις. Μία από τις κύριες δυσκολίες στην εφαρμογή της αρχής είναι η πολυπλοκότητα διαχείρισης μεγάλου αριθμού λεπτομερών πολιτικών ασφαλείας. Για την αντιμετώπιση του παραπάνω ζητήματος, οι ερευνητές προτείνουν τη χρήση τεχνητής νοημοσύνης και μηχανικής μάθησης για την αυτοματοποίηση της διαχείρισης πολιτικών και την ανίχνευση ανωμαλιών [9]. Επιπλέον, η Κατάτμηση Πόρων απαιτεί σημαντικές αλλαγές στην υποδομή και τις διαδικασίες ασφαλείας των οργανισμών. Αυτό μπορεί να περιλαμβάνει την αναβάθμιση του υλικού δικτύωσης, την κατάλληλη εκπαίδευση του

προσωπικού, αλλά και την αναθεώρηση των επιχειρησιακών διαδικασιών. Ως εκ τούτου, η μετάβαση σε ένα μοντέλο Κατάτμησης Πόρων απαιτεί προσεκτικό σχεδιασμό και σταδιακή ενσωμάτωση.

Σε ό,τι αφορά τις μελλοντικές τάσεις, η έρευνα δείχνει ότι η Κατάτμηση Πόρων θα εξελιχθεί περαιτέρω με την ενσωμάτωση τεχνολογιών όπως το edge computing και το IoT [10]. Αυτές οι τεχνολογίες θα απαιτήσουν ακόμη πιο εξελιγμένες μεθόδους κατάτμησης και ελέγχου πρόσβασης, καθώς τα όρια μεταξύ των παραδοσιακών δικτυακών υποδομών και των νέων κατανεμημένων συστημάτων γίνονται ολοένα και πιο ασαφή.

Συμπερασματικά, η εφαρμογή της αρχής Κατάτμησης Πόρων στην Αρχιτεκτονική Δικτύωσης Μηδενικής Εμπιστοσύνης αποτελεί ένα ισχυρό εργαλείο για την ενίσχυση της ασφάλειας στο σύγχρονο απαιτητικό τοπίο της Ασφάλειας Δικτύων. Παρέχει ένα πλαίσιο για τον λεπτομερή έλεγχο πρόσβασης και κίνησης δεδομένων, μειώνοντας σημαντικά τους κινδύνους ασφαλείας.

3.3 Συνεχής Ταυτοποίηση (Continual Authentication)

Η συνεχής ταυτοποίηση και εξουσιοδότηση αποτελούν θεμελιώδεις αρχές της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης, οι οποίες στοχεύουν στη βελτίωση της ασφάλειας των δικτύων και των πληροφοριακών συστημάτων. Η έννοια της συνεχούς ταυτοποίησης περιλαμβάνει τη διαδικασία επαλήθευσης της ταυτότητας ενός χρήστη ή μιας συσκευής σε πραγματικό χρόνο, καθ' όλη τη διάρκεια της συνεδρίας τους. Αυτό σημαίνει ότι οι χρήστες δεν απαιτείται μόνο να ταυτοποιούνται κατά την είσοδό τους στο σύστημα, αλλά και σε τακτά χρονικά διαστήματα ή κάθε φορά που επιχειρούν να αποκτήσουν πρόσβαση σε ευαίσθητους πόρους. Αυτή η προσέγγιση ελαττώνει τον κίνδυνο μη εξουσιοδοτημένης πρόσβασης, καθώς αυξάνονται οι διαδικασίες ελέγχου πρόσβασης τις οποίες πιθανοί επιτιθέμενοι θα έπρεπε να παραβιάσουν ώστε να αποκτήσουν πρόσβαση σε πόρους ενός συστήματος.

Ένα από τα κύρια πλεονεκτήματα της συνεχούς ταυτοποίησης είναι η ικανότητά της να προσαρμόζεται σε μεταβαλλόμενες συνθήκες δικτύου. Για παράδειγμα, εάν μια συσκευή ή ένας χρήστης παρουσιάσει ύποπτη συμπεριφορά, τα συστήματα μπορούν να απαιτούν επιπλέον στοιχεία ταυτοποίησης ή να περιορίζουν την πρόσβαση σε συγκεκριμένους πόρους. Αυτό το επίπεδο ευελιξίας και προσαρμοστικότητας είναι κρίσιμο για την προστασία των οργανισμών από τις εξελισσόμενες απειλές που αντιμετωπίζουν οι δικτυακές υποδομές.

Η ενσωμάτωσή συνεχούς ταυτοποίησης στην Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης απαιτεί χρήση προηγμένων τεχνολογιών και εργαλείων. Οι αλγόριθμοι ανίχνευσης ανωμαλιών, οι οποίοι χρησιμοποιούν τεχνητή νοημοσύνη και μηχανική μάθηση, μπορούν να αναλύσουν τη συμπεριφορά των χρηστών και των συσκευών σε πραγματικό χρόνο, εντοπίζοντας πιθανές απειλές και προσαρμόζοντας τις πολιτικές ασφαλείας αναλόγως. Χρήση τέτοιων αλγορίθμων έχει φανεί ικανή να βελτιώσει σημαντικά την αποτελεσματικότητα των διαδικασιών ταυτοποίησης και εξουσιοδότησης.

Επιπλέον, εφαρμογή συνεχούς εξουσιοδότησης επιτρέπει σε οργανισμούς να εφαρμόζουν πολιτικές Ελάχιστων Προνομίων (least privilege access), παρέχοντας στους χρήστες μόνο τα απολύτως απαραίτητα δικαιώματα πρόσβασης για την εκτέλεση των καθηκόντων τους. Αυτή η προσέγγιση μπορεί να μειώσει τα πιθανώς εκτεθειμένα σε

απειλές σημεία ενός δικτύου (attack surface) και να περιορίσει τις δυνατότητες “πλευρικής” κίνησης των επιτιθέμενων εντός των λογικών ορίων του.

Σημαντική είναι επίσης η ανάγκη για συνεχή αξιολόγηση των κινδύνων και των απειλών που αντιμετωπίζουν τα δικτυακά συστήματα. Η εφαρμογή μοντέλων συνεχούς ταυτοποίησης απαιτεί τη συλλογή και ανάλυση δεδομένων σχετικά με τις δραστηριότητες χρηστών και συσκευών, προκειμένου να εντοπίζονται τυχόν ανωμαλίες ή ύποπτη δικτυακή κίνηση. Αυτή η διαδικασία μπορεί να περιλαμβάνει χρήση εργαλείων παρακολούθησης δικτύου και ανάλυσης δεδομένων σε πραγματικό χρόνο για την αποτίμηση της ασφάλειας μιας δικτυακής υποδομής.

Η υιοθέτηση συνεχούς ταυτοποίησης και εξουσιοδότησης στην Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης έχει αποδειχθεί ιδιαίτερα αποτελεσματική σε διάφορους τομείς, όπως η υγειονομική περίθαλψη, οι χρηματοπιστωτικές υπηρεσίες, αλλά και οι δημόσιες υπηρεσίες. Στις περιπτώσεις αυτές, η ανάγκη για προστασία ευαίσθητων δεδομένων είναι επιτακτική, καθιστώντας την εφαρμογή αυτών των αρχών ζωτικής σημασίας για τη συνολική ασφάλεια των συστημάτων.

Ωστόσο, υπάρχουν προκλήσεις στην υλοποίηση της συνεχούς ταυτοποίησης σε δικτυακά συστήματα. Η πολυπλοκότητα της διαχείρισης πολλαπλών πολιτικών ασφαλείας μπορεί να οδηγήσει σε δυσκολίες στην εφαρμογή τους, ενώ η ανάγκη για συνεχή παρακολούθηση μπορεί να απαιτήσει σημαντικούς πόρους και υποδομές. Επιπλέον, οι οργανισμοί πρέπει να διασφαλίσουν ότι οι διαδικασίες ταυτοποίησης δεν επηρεάζουν αρνητικά την εμπειρία του χρήστη.

Συμπερασματικά, η συνεχής ταυτοποίηση και εξουσιοδότηση αποτελεί κεντρική αρχή της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης που συμβάλλει καθοριστικά στην ενίσχυση της ασφάλειας των δικτυακών υπηρεσιών. Με τη χρήση προηγμένων τεχνολογιών και εργαλείων ανάλυσης δεδομένων, οι οργανισμοί μπορούν να προστατεύσουν πιο αποτελεσματικά τους πόρους τους από τις σύγχρονες, συνεχώς εξελισσόμενες απειλές. Η συνεχής επαλήθευση της ταυτότητας και η δυναμική διαχείριση των δικαιωμάτων πρόσβασης δημιουργούν ένα πιο ασφαλές περιβάλλον για την εκτέλεση επιχειρησιακών διαδικασιών, ενισχύοντας τη συνολική ανθεκτικότητα των οργανισμών απέναντι σε κυβερνοεπιθέσεις.

3.4 Κρυπτογράφηση Συνδέσεων

Θεμελιώδες συστατικό στοιχείο της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης αποτελεί αναμφίβολα η ασφάλεια των συνδέσεων μέσω της κρυπτογράφησης τους. Πολλαπλές προσεγγίσεις και λύσεις έχουν διαμορφωθεί μέσω της προόδου των πρωτοκόλλων ασφάλειας, αλλά και των μεθόδων κρυπτογράφησης. Αναπόσπαστο στοιχείο της σύγχρονης κρυπτογράφησης επικοινωνιών αποτελεί το πρωτόκολλο TLS. Η νεότερη έκδοση του πρωτοκόλλου (έκδοση 1.3) περιγράφηκε στο Request For Comments 8446 (RFC 8446) από τον σύμβουλο ασφάλειας δικτύων Eric Riscola [11]. Τόσο οι επιδόσεις όσο και η ασφάλεια των κρυπτογραφημένων συνδέσεων έχουν γνωρίσει αξιοσημείωτες βελτιώσεις με την εισαγωγή της νεότερης έκδοσης 1.3 του πρωτοκόλλου TLS αλλά και ευρύτερα με τη μετάβαση του πρωτοκόλλου από την αρχική έκδοση του (SSL) προς το TLS.

Εξίσου θεμελιώδες συστατικό στοιχείο της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης αποτελούν και οι ίδιοι οι αλγόριθμοι κρυπτογράφησης. Οι σύγχρονοι αλγόριθμοι κρυπτογράφησης αναλύονται λεπτομερώς από τον καθηγητή Dan Boneh και τον ερευνητή Victor Shoup στο σύγγραμμα "A Graduate Course in Applied Cryptography" που εκδόθηκε το 2017 [12]. Ενισχυμένη ασφάλεια για τις συνδέσεις δικτύου προσφέρεται με την υιοθέτηση εξελιγμένων κρυπτογραφικών αλγορίθμων, τόσο συμμετρικών όσο και μη συμμετρικών. Δύο περιπτώσεις αλγορίθμων κρυπτογράφησης που τυγχάνουν χρήσης σε πληθώρα εφαρμογών είναι ο RSA και ο AES-GCM. Ο αλγόριθμος RSA τυγχάνει εξαιρετικά ευρείας χρήσης μέσω των ψηφιακών πιστοποιητικών X.509 όπως αυτά έχουν περιγραφεί στο Request For Comments 5280 (RFC 5280) [13].

Παράλληλα, σημαντικό κομμάτι της διαδικασίας εγκατάστασης μιας ασφαλούς κρυπτογραφημένης δικτυακής σύνδεσης αποτελεί και ο καθορισμός των κρυπτογραφικών κλειδιών που χρησιμοποιούνται στις παραπάνω συνδέσεις. Για τον καθορισμό των κλειδιών που χρησιμοποιούνται για την κρυπτογράφηση των δικτυακών επικοινωνιών έχουν αναπτυχθεί κατάλληλα πρωτόκολλα. Τα παραπάνω πρωτόκολλα συνήθως αναφέρονται στη βιβλιογραφία ως πρωτόκολλα "ανταλλαγής" κρυπτογραφικών κλειδιών. Ίσως το πιο ευρέως χρησιμοποιούμενος αλγόριθμος για τον καθορισμό κλειδιών κρυπτογραφημένων επικοινωνιών δικτύου στο σύγχρονο σκηνικό της ασφάλειας δικτύων είναι ο Elliptic Curve Diffie-Hellman (ECDH), ως επέκταση του προϋπάρχοντος αλγορίθμου Diffie-Hellman.

Άξιο αναφοράς είναι ότι ακόμα και οι κρυπτογραφημένες δικτυακές επικοινωνίες δέχονται επιθέσεις. Ορισμένες από τις πιο συνήθεις είναι οι επιθέσεις επανάληψης, οι επιθέσεις υποβάθμισης πρωτοκόλλου καθώς και οι επιθέσεις Man-in-the-Middle. Αναλυτική έρευνα σχετικά με την αντοχή της νεότερης έκδοσης του πρωτοκόλλου TLS (version 1.3) παρουσιάζεται στη δημοσίευση με τίτλο "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", που δημοσιεύθηκε το 2017 [14]. Η ανάπτυξη κβαντικών υπολογιστών έχει καταστήσει τη μελλοντική επάρκεια των σύγχρονων κρυπτογραφικών αλγορίθμων αμφίβολη, αν δεν αναπτυχθούν νέοι ή δεν εξελιχθούν οι υπάρχοντες ώστε να είναι ανθεκτικοί σε επιθέσεις από κβαντικούς υπολογιστές.

Το επόμενο μεγάλο άλμα που καλείται να πραγματοποιήσει το ερευνητικό πεδίο της κρυπτογραφίας σκιαγραφείται στο άρθρο με τίτλο "Post-quantum cryptography" που δημοσιεύθηκε το 2017 από τον Daniel j. Bernstein και την Tanja Lange [15]. Οι νεότεροι κρυπτογραφικοί αλγόριθμοι είναι ικανοί να παρουσιάζουν αμελητέο αντίκτυπο στις επιδόσεις των δικτυακών επικοινωνιών συνδυάζοντας τον παράλληλα με αρκετά αυξημένο επίπεδο ασφάλειας. Έτσι κατορθώνουν ένα αξιοσημείωτο βήμα και προς την κατεύθυνση διαφύλαξης των προσωπικών δεδομένων των χρηστών του διαδικτύου, που τα τελευταία χρόνια φαίνεται να είναι από τους κυρίαρχους στόχους των δικτυακών επιθέσεων.

4. Μηχανισμοί εφαρμογής Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης

4.1 Σύγχρονοι μηχανισμοί Ταυτοποίησης

Οι σύγχρονοι μηχανισμοί ταυτοποίησης και η ενσωμάτωσή τους στην Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης αποτελούν κρίσιμα στοιχεία για την ασφάλεια των σύγχρονων πληροφοριακών συστημάτων. Η προσέγγιση της μηδενικής εμπιστοσύνης συμπυκνώνεται στη φράση "ποτέ μην εμπιστεύεσαι, πάντα επαλήθευε" (“never trust, always verify”), η οποία απαιτεί συνεχή και αυστηρή ταυτοποίηση για κάθε πρόσβαση σε πόρους του συστήματος.

Ένας από τους πιο σημαντικούς μηχανισμούς ταυτοποίησης που ενσωματώνονται σε υλοποιήσεις Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης είναι η πολυπαραγοντική ταυτοποίηση (MFA). Η MFA απαιτεί από τους χρήστες να παρέχουν δύο ή περισσότερα αποδεικτικά στοιχεία ταυτότητας πριν τους επιτραπεί η πρόσβαση σε ένα σύστημα. Αυτά τα στοιχεία μπορεί να περιλαμβάνουν κάτι που ο χρήστης γνωρίζει (π.χ. κωδικός πρόσβασης), κάτι που ο χρήστης διατηρεί στην κατοχή του (π.χ. μια συσκευή) ή κάποιο φυσικό χαρακτηριστικό του (βιομετρικά δεδομένα). Η ενσωμάτωση της MFA στην Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης ενισχύει σημαντικά την ασφάλεια των συστημάτων, καθιστώντας πολύ πιο δύσκολο για τους επιτιθέμενους να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση [16].

Μια καινοτόμος προσέγγιση στην ταυτοποίηση που συγκεντρώνει όλο και μεγαλύτερο ενδιαφέρον είναι η χρήση της τεχνολογίας blockchain. Η ενσωμάτωση του blockchain στους μηχανισμούς ταυτοποίησης προσφέρει αυξημένη ασφάλεια και διαφάνεια στη διαδικασία επαλήθευσης ταυτότητας. Ένα παράδειγμα αυτής της προσέγγισης είναι η χρήση αποδείξεων μηδενικής γνώσης (Zero-Knowledge Proofs - ZKP) βασισμένων σε blockchain για την επαλήθευση της γνώσης ενός κωδικού μιας χρήσης (One-Time Password - OTP) χωρίς να αποκαλύπτεται ο ίδιος ο κωδικός. Αυτή η μέθοδος όχι μόνο διασφαλίζει την αυθεντικότητα της απόδειξης, αλλά επίσης επιβεβαιώνει την ταυτότητα του αποδεικνύοντος.

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο της παρούσας Διπλωματικής Εργασίας αναπόσπαστο κομμάτι της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης αποτελεί επίσης η συνεχής ταυτοποίηση (Continuous Authentication), ιδιαίτερα στο πλαίσιο του Διαδικτύου των Πραγμάτων (IoT). Ένα παράδειγμα τέτοιου πρωτοκόλλου είναι το STCA (Stacked Token-based Continuous Authentication), το οποίο χρησιμοποιεί στοιβαγμένα tokens για να διασφαλίσει τη ταυτότητα των οντοτήτων καθ' όλη τη διάρκεια μιας συνεδρίας. Το STCA έχει αποδειχθεί ανθεκτικό σε διάφορες συνήθεις επιθέσεις και παρουσιάζει καλύτερες επιδόσεις σε σύγκριση με άλλα πρωτόκολλα συνεχούς ταυτοποίησης, καθιστώντας το ιδιαίτερα κατάλληλο για υλοποιήσεις Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης σε περιβάλλοντα IoT [17].

Στο πλαίσιο δικτύων IoT, η εφαρμογή ταυτοποίησης αντιμετωπίζει προκλήσεις λόγω της ποικιλομορφίας των συστημάτων, αλλά και του μεγάλου αριθμού των συσκευών. Οι

ερευνητές προτείνουν διάφορες λύσεις, συμπεριλαμβανομένης της χρήσης ελαφριάς κρυπτογραφίας, αμοιβαίας ταυτοποίησης ή και τεχνολογίας blockchain. Η αξιολόγηση των επιδόσεων των μεθόδων ταυτοποίησης σε δίκτυα IoT λαμβάνει υπόψη παράγοντες όπως η ασφάλεια, η αποδοτικότητα και η δυνατότητα κλιμάκωσης των συστημάτων [18].

Ένας άλλος καινοτόμος μηχανισμός ταυτοποίησης που ενσωματώνεται σε υλοποιήσεις Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης είναι το Decentralized Open Interoperable Security Protocol (dOISP). Το dOISP προσφέρει μια προσέγγιση ταυτοποίησης και κρυπτογράφησης σε επίπεδο 3 του μοντέλου OSI, η οποία είναι πλήρως αποκεντρωμένη, ανθεκτική σε κβαντικές επιθέσεις και λειτουργεί χωρίς την ανάγκη για αποθηκευμένες βάσεις δεδομένων, πιστοποιητικά ή κεντρική συνδεσιμότητα έπειτα από την αρχική εγκατάσταση. Αυτή η προσέγγιση επιτρέπει την ταχεία ενσωμάτωση διαφορετικών υποδομών δικτύου, συμπεριλαμβανομένων εκείνων που βασίζονται σε High Assurance Internet Protocol Encryptors (HAIPE) [19].

Η ενσωμάτωση τεχνητής νοημοσύνης (AI) και μηχανικής μάθησης (ML) στους μηχανισμούς ταυτοποίησης αποτελεί επίσης μια αναδυόμενη τάση στις εφαρμογές της Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης. Αυτές οι τεχνολογίες χρησιμοποιούνται για τη βελτίωση ανίχνευσης ανωμαλιών, την πρόβλεψη πιθανών απειλών και την προσαρμογή των πολιτικών ασφαλείας σε πραγματικό χρόνο. Για παράδειγμα, τεχνικές βαθιάς ενισχυτικής μάθησης (Deep Reinforcement Learning) έχουν χρησιμοποιηθεί για τη βελτιστοποίηση των πολιτικών ασφαλείας σε δυναμικά περιβάλλοντα IoT [20].

Ωστόσο, παρά τις σημαντικές προόδους στους μηχανισμούς ταυτοποίησης, εξακολουθούν να υφίστανται προκλήσεις στην ενσωμάτωση τους στην Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης. Από τις κύριες προκλήσεις συνιστά η διαχείριση της πολυπλοκότητας που ανακύπτει από την εφαρμογή πολλαπλών επιπέδων ταυτοποίησης και ελέγχου πρόσβασης. Παράλληλα, η χρυσή τομή συμβιβασμού μεταξύ ασφάλειας και χρηστικότητας παραμένει ένα κρίσιμο ζήτημα, καθώς οι πιο αυστηροί μηχανισμοί ταυτοποίησης μπορεί να επηρεάσουν αρνητικά την εμπειρία των χρηστών.

Συμπερασματικά, η υιοθέτηση σύγχρονων μηχανισμών ταυτοποίησης σε υλοποιήσεις Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης αποτελεί ένα δυναμικό και ταχέως εξελισσόμενο πεδίο έρευνας. Η συνεχής ανάπτυξη νέων τεχνολογιών και μεθόδων, όπως η blockchain-based ταυτοποίηση, η συνεχής ταυτοποίηση και η ενσωμάτωση AI/ML, υπόσχεται να ενισχύσει περαιτέρω την ασφάλεια των σύγχρονων πληροφοριακών συστημάτων. Ωστόσο, η αποτελεσματική εφαρμογή αυτών των μηχανισμών απαιτεί προσεκτικό σχεδιασμό και εξισορρόπηση μεταξύ ασφάλειας, επιδόσεων και χρηστικότητας.

4.1.1 Ψηφιακά Πιστοποιητικά X.509

Τα ψηφιακά πιστοποιητικά X.509 αποτελούν έναν θεμελιώδη μηχανισμό ασφαλείας στις σύγχρονες υποδομές δημόσιου κλειδιού (Public Key Infrastructure - PKI) και διαδραματίζουν κρίσιμο ρόλο στην εξασφάλιση ασφαλών επικοινωνιών μέσω του Διαδικτύου. Το πρότυπο X.509, που αναπτύχθηκε αρχικά από τη Διεθνή Ένωση Τηλεπικοινωνιών (ITU), καθορίζει τη δομή και το περιεχόμενο των ψηφιακών πιστοποιητικών, παρέχοντας ένα τυποποιημένο μέσο για την επαλήθευση της ταυτότητας οντοτήτων στο ψηφιακό περιβάλλον.

Τα ψηφιακά πιστοποιητικά X.509 περιέχουν βασικές πληροφορίες όπως το δημόσιο κλειδί του κατόχου, την ταυτότητα του κατόχου, την ταυτότητα της Αρχής Πιστοποίησης (Certificate Authority - CA) που εξέδωσε το πιστοποιητικό, την περίοδο ισχύος και άλλα σχετικά δεδομένα. Η ψηφιακή υπογραφή της CA στο πιστοποιητικό εγγυάται την ακεραιότητα και την αυθεντικότητά του. Αυτή η δομή επιτρέπει στα συστήματα να επαληθεύουν την ταυτότητα των οντοτήτων και να εδραιώνουν ασφαλείς συνδέσεις μέσω διαφόρων πρωτοκόλλων και εφαρμογών, όπως το HTTP Secure (HTTPS), το SSL/TLS και το Secure/Multipurpose Internet Mail Extensions (S/MIME).

Η χρήση των πιστοποιητικών X.509 εκτείνεται σε πολλούς τομείς της ψηφιακής ασφάλειας. Στο πλαίσιο του Διαδικτύου των Πραγμάτων (IoT), για παράδειγμα, τα πιστοποιητικά X.509 παίζουν καθοριστικό ρόλο στην εξασφάλιση ασφαλούς επικοινωνίας μεταξύ συσκευών. Ερευνητές έχουν προτείνει προσαρμοσμένες αρχιτεκτονικές υποδομών δημοσίου κλειδιού (PKI) που χρησιμοποιούν ειδικά σχεδιασμένα πιστοποιητικά X.509 για την αντιμετώπιση των μοναδικών προκλήσεων που θέτουν τα περιβάλλοντα IoT, όπως οι περιορισμοί πόρων και η ανάγκη για κλιμάκωση [21].

Ωστόσο, η εφαρμογή και η διαχείριση πιστοποιητικών X.509 αποτελεί διαδικασία που καλείται να αντιμετωπίσει αρκετές δυσκολίες. Η συμμόρφωση με τις προδιαγραφές του προτύπου X.509 είναι κρίσιμη για τη διασφάλιση της διαλειτουργικότητας και της ασφάλειας. Μια μελέτη που διεξήχθη σε πιστοποιητικά X.509 σε δίκτυα IPv6 αποκάλυψε ότι ένα σημαντικό ποσοστό πιστοποιητικών δεν συμμορφώνονταν πλήρως με τις προδιαγραφές, υπογραμμίζοντας την ανάγκη για συνεχή παρακολούθηση και βελτίωση των πρακτικών έκδοσης πιστοποιητικών [22].

Η ασφάλεια των πιστοποιητικών X.509 εξαρτάται σε μεγάλο βαθμό από την ακεραιότητα των Αρχών Πιστοποίησης (CAs) που τα εκδίδουν. Περιστατικά όπου οι CAs έχουν παραβιαστεί ή έχουν εκδώσει μη εξουσιοδοτημένα πιστοποιητικά έχουν οδηγήσει σε σημαντικές ανησυχίες αναφορικά με τη συνολική ασφάλεια των συστημάτων. Για την αντιμετώπιση των ζητημάτων αυτών, έχουν προταθεί καινοτόμες προσεγγίσεις, όπως η χρήση της τεχνολογίας blockchain για τη διαχείριση των πιστοποιητικών X.509 [23]. Αυτή η προσέγγιση προσφέρει πλεονεκτήματα όπως η εξάλειψη του μοναδικού σημείου αποτυχίας και η δυνατότητα ταχείας αντίδρασης σε περιπτώσεις παραβίασης των Αρχών Πιστοποίησης (CAs).

Επιπλέον, η εξέλιξη των υπολογιστικών δυνατοτήτων, ιδιαίτερα η ανάπτυξη των κβαντικών υπολογιστών, θέτει νέες προκλήσεις για την ασφάλεια των πιστοποιητικών X.509. Οι κρυπτογραφικοί αλγόριθμοι που χρησιμοποιούνται σήμερα για την υπογραφή και την επαλήθευση των πιστοποιητικών ενδέχεται να καταστούν ευάλωτοι σε επιθέσεις από κβαντικούς υπολογιστές. Ως εκ τούτου, η έρευνα στον τομέα της μετα-κβαντικής κρυπτογραφίας και η ενσωμάτωσή της στα πρότυπα X.509 αποτελεί ένα κρίσιμο πεδίο μελέτης για τη διασφάλιση της μακροπρόθεσμης ασφάλειας των ψηφιακών πιστοποιητικών.

Συμπερασματικά, τα ψηφιακά πιστοποιητικά X.509 παραμένουν ένα θεμελιώδες στοιχείο της σύγχρονης ψηφιακής ασφάλειας, παρέχοντας ένα αξιόπιστο μέσο για την επαλήθευση ταυτότητας και την εγκαθίδρυση ασφαλών επικοινωνιών. Ωστόσο, η συνεχής έρευνα και καινοτομία είναι απαραίτητες για την αντιμετώπιση των αναδυόμενων προκλήσεων και τη διασφάλιση της αποτελεσματικότητάς τους στο μεταβαλλόμενο τοπίο της ψηφιακής ασφάλειας. Η ενσωμάτωση νέων τεχνολογιών, όπως το blockchain και η μετα-κβαντική κρυπτογραφία, υπόσχεται να ενισχύσει περαιτέρω την ασφάλεια και την

αξιοπιστία των πιστοποιητικών X.509, διασφαλίζοντας τη συνεχή τους ενσωμάτωση στις σύγχρονες υποδομές ασφάλειας.

4.1.2 JSON Web Tokens

Τα Javascript Object Notation Web Tokens (JWT) αποτελούν ένα σύγχρονο και ευρέως διαδεδομένο πρότυπο για την ασφαλή μετάδοση πληροφοριών μεταξύ διαφόρων μερών ως αντικείμενα JSON. Τα JWT χρησιμοποιούνται εκτενώς σε εφαρμογές διαδικτύου και μικροϋπηρεσίες (microservices) για την ταυτοποίηση και εξουσιοδότηση χρηστών, καθώς και για την ασφαλή ανταλλαγή πληροφοριών.

Η δομή ενός JWT αποτελείται από τρία μέρη: την επικεφαλίδα (header), τα δεδομένα (payload) και την υπογραφή (signature). Η επικεφαλίδα περιέχει πληροφορίες σχετικά με τον τύπο του token και τον αλγόριθμο κρυπτογράφησης που χρησιμοποιείται. Τα δεδομένα (payload) περιλαμβάνουν τα πεδία "claims" σχετικά με την οντότητα (συνήθως τον χρήστη) και πρόσθετα μεταδεδομένα. Η υπογραφή χρησιμοποιείται για την επαλήθευση της ακεραιότητας του token και για να διασφαλίσει ότι δεν έχει παραποιηθεί [24].

Ένα από τα κύρια πλεονεκτήματα των JWT είναι η stateless φύση τους, που σημαίνει ότι ο διακομιστής (server) δεν χρειάζεται να αποθηκεύει πληροφορίες σχετικά τις συνεδρίες του χρήστη. Αυτό καθιστά τα JWT ιδιαίτερα χρήσιμα σε κατανεμημένα συστήματα και μικροϋπηρεσίες, όπου η κλιμάκωση και η διαλειτουργικότητα είναι κρίσιμες [25]. Η ασφάλεια των JWT εξαρτάται σε μεγάλο βαθμό από την ορθή εφαρμογή των κρυπτογραφικών αλγορίθμων και την προστασία των μυστικών κλειδιών που χρησιμοποιούνται για την υπογραφή των tokens. Επιπλέον, η αποθήκευση των JWT στον πελάτη (client) μπορεί να εγείρει ζητήματα ασφαλείας εάν δεν υλοποιηθεί με ασφάλεια [26].

Μια σημαντική πτυχή της ασφάλειας των JWT είναι η επιλογή του κατάλληλου αλγορίθμου κρυπτογράφησης. Οι παραδοσιακοί αλγόριθμοι που χρησιμοποιούνται συχνά, όπως ο RSA, ενδέχεται να καταστούν ευάλωτοι σε επιθέσεις από κβαντικούς υπολογιστές στο μέλλον. Για την αντιμετώπιση αυτής της απειλής, έχουν προταθεί κβαντικά ανθεκτικοί αλγόριθμοι βασισμένοι σε "πλέγματα" (lattice-based) για τη δημιουργία και επαλήθευση των υπογραφών JWT. Έρευνες έχουν δείξει ότι αλγόριθμοι όπως το DILITHIUM και το qTESLA παρουσιάζουν καλύτερη επίδοση από τον RSA σε ρυθμό αιτημάτων ανά δευτερόλεπτο και μέσο χρόνο απόκρισης, με το DILITHIUM να επιδεικνύει την καλύτερη συνολική επίδοση [27].

Επιπλέον, έχουν προταθεί καινοτόμες προσεγγίσεις για την ενίσχυση της ασφάλειας των JWT, όπως τα Biometric JSON Web Tokens (BJWT). Αυτή η προσέγγιση συνδυάζει τη βιομετρική ανταλλαγή κλειδιών με την ταυτοποίηση OTP-JWT για την ενίσχυση της ασφάλειας των προγραμματιστικών διεπαφών (Application Programming Interfaces – APIs). Η σύγκριση των JWT με άλλες τεχνολογίες ασφαλείας είναι επίσης ένα σημαντικό πεδίο έρευνας. Μια μελέτη συνέκρινε τις επιδόσεις και την ασφάλεια των JWT με τα Platform Agnostic Security Tokens (PASETO) σε Representational State Transfer (RESTful) APIs. Τα αποτελέσματα έδειξαν ότι τα PASETO έχουν μεγαλύτερο χρόνο δημιουργίας και μεταφοράς token σε σύγκριση με τα JWT, αλλά προσφέρουν υψηλότερο επίπεδο ασφάλειας, καθώς μπορούν να προστατεύσουν από τις τρεις κορυφαίες επιθέσεις του Open Web Application Security Project (OWASP) 2019 API [28].

Η χρήση των JWT σε συγκεκριμένα πλαίσια, όπως το Hadoop, έχει επίσης διερευνηθεί. Προτάθηκε ένας μηχανισμός ταυτοποίησης βασισμένος σε JWT για το Hadoop (JWTAMH), ο οποίος συνδυάζει το Kerberos με τα JWT για ασφαλή επικοινωνία μεταξύ των στοιχείων του Hadoop. Τα πειραματικά αποτελέσματα έδειξαν την αποτελεσματικότητα αυτής της προσέγγισης έναντι διαφόρων τύπων επιθέσεων [29].

Τέλος, η κατανόηση του κύκλου ζωής των JWT και των σχετικών απειλών είναι κρίσιμη για την ασφαλή ενσωμάτωση τους σε συστήματα ελέγχου πρόσβασης δικτυακών υποδομών. Μια ανάλυση 139 ευπαθειών σχετικών με την ταυτοποίηση βασισμένη σε JWT από τη γνωστή βάση δεδομένων ευπαθειών Common Vulnerabilities and Exposures (CVE) αποκάλυψε μια τάση αύξησης των ευπαθειών "Κρίσιμου" και "Υψηλού" επιπέδου σοβαρότητας. Με βάση αυτή την ανάλυση, προτάθηκε μια καθολική ταξινόμηση των απειλών ασφαλείας των JWT βασισμένη στον κύκλο ζωής τους, η οποία μπορεί να χρησιμοποιηθεί στο σχεδιασμό συστημάτων πληροφοριών, τη μοντελοποίηση απειλών, την αξιολόγηση ασφάλειας και τη διερεύνηση περιστατικών ασφαλείας [30].

Συμπερασματικά, τα JWT αποτελούν ένα ισχυρό εργαλείο για την ασφαλή μετάδοση πληροφοριών και την ταυτοποίηση χρηστών σε σύγχρονες εφαρμογές διαδικτύου. Ωστόσο, η αποτελεσματική και ασφαλής χρήση τους απαιτεί προσεκτική εφαρμογή, συνεχή έρευνα και προσαρμογή στις αναδυόμενες απειλές ασφαλείας.

4.1.3 Time-based One-Time Passwords (TOTP)

Οι Κωδικοί Πρόσβασης μίας χρήσης που βασίζονται στον τρέχοντα χρόνο (TOTP) αποτελούν μια σημαντική εξέλιξη στον τομέα της ασφάλειας πληροφοριακών συστημάτων και της ταυτοποίησης χρηστών. Πρόκειται για έναν μηχανισμό που παράγει δυναμικούς, εφήμερους κωδικούς πρόσβασης, χρησιμοποιώντας ως είσοδο την τιμή της τρέχουσας χρονικής στιγμής (epoch time, δηλαδή ο αριθμός δευτερολέπτων από την 1/1/1970) και ένα μυστικό κλειδί, τα οποία επεξεργάζονται μέσω μιας κρυπτογραφικής συνάρτησης κατακερματισμού [31].

Η χρήση των TOTP έχει αποδειχθεί ιδιαίτερα αποτελεσματική στην αντιμετώπιση των περιορισμών των παραδοσιακών συστημάτων κωδικών πρόσβασης μιας χρήσης, βελτιώνοντας σημαντικά την ασφάλεια μέσω της μείωσης των ευπαθειών σε επιθέσεις επαναχρησιμοποίησης κωδικών και υποκλοπής [31]. Η εφαρμογή τους εκτείνεται σε ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένων τραπεζικών εφαρμογών, ιστοσελίδων ηλεκτρονικού εμπορίου, συστημάτων ελέγχου πρόσβασης και συστημάτων υγειονομικής περίθαλψης [31].

Ένα σημαντικό πλεονέκτημα των Time-based OTPs είναι η δυνατότητα αποτελεσματικής χρήσης τους σε συσκευές με περιορισμένους πόρους, όπως αυτές που χρησιμοποιούνται σε δίκτυα IoT [31]. Αυτό επιτυγχάνεται μέσω χρήσης κρυπτογραφικών αλγορίθμων με μικρότερη επίδραση στις επιδόσεις, οι οποίοι εξασφαλίζουν την αποδοτική ενσωμάτωση μηχανισμών TOTP σε περιβάλλοντα με περιορισμένους υπολογιστικούς πόρους.

Ένα σημαντικό ζήτημα αναφορικά με τη χρήση TOTP είναι ο συγχρονισμός μεταξύ των συσκευών των πελατών και των διακομιστών αναφορικά με τον τρέχοντα χρόνο [31]. Επιπλέον, υπάρχει ένας συμβιβασμός μεταξύ της υπολογιστικής αποδοτικότητας και της

ισχύος της ασφάλειας, ο οποίος πρέπει να λαμβάνεται υπόψη κατά το σχεδιασμό και την εφαρμογή συστημάτων TOTP.

Η έρευνα στον τομέα των TOTP συνεχίζεται με στόχο την αντιμετώπιση αυτών των προκλήσεων και τη βελτίωση της ασφάλειας. Μια ενδιαφέρουσα εξέλιξη είναι η ανάπτυξη των Δυναμικών Ομαδικών Κωδικών Πρόσβασης μίας χρήσης βασισμένων στον τρέχοντα χρόνο (Dynamic Group TOTP - DGTOTP) [32]. Αυτή η προσέγγιση επιτρέπει την αποτελεσματική παραγωγή εφήμερων κωδικών πρόσβασης εκ μέρους μιας ομάδας, χωρίς να αποκαλύπτεται καμία πληροφορία σχετικά με την πραγματική ταυτότητα του χρήστη πέρα από το γεγονός της ότι αποτελεί μέρος της παραπάνω "ομάδας".

Επιπλέον, έχουν προταθεί καινοτόμες προσεγγίσεις για την ενίσχυση της ασφάλειας των TOTP. Μία τέτοια προσέγγιση είναι η χρήση βιομετρικών δεδομένων σε συνδυασμό με τους TOTP. Αυτή η μέθοδος, γνωστή ως Βιομετρική ταυτοποίηση συναλλαγών βασισμένη σε κινητό OTP, στοχεύει στην αντιμετώπιση των απειλών ασφαλείας στις διαδικτυακές συναλλαγές, όπως οι επιθέσεις phishing, επιθέσεις Man-in-the-Middle και οι επιθέσεις brute force.

Στο πλαίσιο του Metaverse, έχει προταθεί ένα σύστημα ταυτοποίησης βασισμένο σε TOTP για την προστασία των χρηστών από την κλοπή ταυτότητας (identity theft) [33]. Αυτή η προσέγγιση, γνωστή ως TOTPAuth, έχει σχεδιαστεί για να αντιμετωπίσει τις απειλές που σχετίζονται με την παρακολούθηση και την καταγραφή των κινήσεων των χρηστών κατά τη διαδικασία ταυτοποίησης. Τέλος, η έρευνα έχει επεκταθεί και στον τομέα της κβαντικής κρυπτογραφίας, με την ανάπτυξη Κβαντικών Κωδικών Πρόσβασης Μίας Χρήσης (Quantum OTP - QOTP) [34].

Συμπερασματικά, οι Κωδικοί Πρόσβασης Μίας Χρήσης που βασίζονται στον τρέχοντα χρόνο αποτελούν μια ισχυρή και ευέλικτη λύση για ταυτοποίηση σε ένα ευρύ φάσμα εφαρμογών. Καθώς η τεχνολογία εξελίσσεται και οι απειλές ασφαλείας γίνονται όλο και πιο εξελιγμένες, η συνεχής έρευνα και ανάπτυξη στον τομέα των TOTP παραμένει κρίσιμη για τη προστασία των ψηφιακών συστημάτων και των χρηστών τους από απειλές.

4.2 Mutual TLS

Η αμφίδρομη εφαρμογή του TLS αποτελεί ένα κρίσιμο στοιχείο της σύγχρονης ασφάλειας δικτύων, παρέχοντας ισχυρή προστασία για τις επικοινωνίες μεταξύ "πελατών" και διακομιστών. Σε αντίθεση με την τυπική υλοποίηση του πρωτοκόλλου, όπου μόνο ο διακομιστής ταυτοποιείται στον πελάτη, η εφαρμογή mTLS απαιτεί και από τις δύο πλευρές να αποδείξουν την ταυτότητά τους, προσφέροντας έτσι ένα υψηλότερο επίπεδο ασφάλειας.

Η εφαρμογή mTLS έχει αποδειχθεί ιδιαίτερα σημαντική σε περιβάλλοντα υψηλής ασφάλειας, όπως τα επιχειρηματικά δίκτυα και οι χρηματοπιστωτικές υπηρεσίες. Μία σημαντική πτυχή της υλοποίησης του mTLS είναι η διαχείριση των πιστοποιητικών των πελατών, η οποία μπορεί να είναι πολύπλοκη σε μεγάλης κλίμακας περιβάλλοντα. Επιπλέον, η ανάγκη για συμβατότητα με παλαιότερα συστήματα μπορεί να περιορίσει την υιοθέτηση των πιο πρόσφατων εκδόσεων του TLS.

Μια ενδιαφέρουσα εξέλιξη στον τομέα της αμοιβαίας TLS είναι η ανάπτυξη του πρωτοκόλλου LURK (Limited Use of Remote Keys). Το LURK επιτρέπει την ασφαλή ανάθεση της υλοποίησης συνδέσεων με χρήση του TLS σε ενδιάμεσους κόμβους,

διατηρώντας παράλληλα τον έλεγχο του διακομιστή TLS [35]. Αυτή η προσέγγιση προσφέρει μια ισορροπία μεταξύ ασφάλειας και επιδόσεων, ιδιαίτερα σε περιβάλλοντα όπου απαιτείται η χρήση ενδιάμεσων συσκευών για την παρακολούθηση και διαχείριση των συνδέσεων που χρησιμοποιούν TLS.

Στο πλαίσιο των ασύρματων ad hoc δικτύων, η χρήση mTLS παρουσιάζει πρόσθετες προκλήσεις λόγω της δυναμικής φύσης των δικτύων αυτών. Έρευνες έχουν προτείνει τη χρήση πρωτοκόλλων ταυτοποίησης βασισμένων σε ZKP σε συνδυασμό με mTLS για την αντιμετώπιση αυτών των προκλήσεων. Η ασφάλεια σε εταιρικά περιβάλλοντα (Enterprise Level Security - ELS) αποτελεί ένα άλλο πεδίο όπου το mTLS διαδραματίζει κρίσιμο ρόλο. Σε αυτό το πλαίσιο, η αμοιβαία ταυτοποίηση διακομιστή και πελάτη μέσω TLS χρησιμοποιείται σε συνδυασμό με κάρτες Personal Identity Verification (PIV) για την εξασφάλιση υψηλών επιπέδων ασφάλειας [36].

Συμπερασματικά, η υλοποίηση αμφίδρομης ταυτοποίησης μέσω mTLS αποτελεί ένα ισχυρό εργαλείο για την ασφάλεια των σύγχρονων δικτύων. Καθώς οι απειλές που αντιμετωπίζουν οι δικτυακές υπηρεσίες συνεχίζουν να εξελίσσονται, η συνεχής έρευνα και ανάπτυξη στον τομέα αυτό παραμένει κρίσιμη για τη διατήρηση της ασφάλειας των ψηφιακών επικοινωνιών.

4.3 Αλγόριθμος δημιουργίας κρυπτογραφικών κλειδιών Diffie-Hellman

Το πρωτόκολλο δημιουργίας κοινών κρυπτογραφικών κλειδιών Diffie-Hellman αποτελεί έναν θεμελιώδη μηχανισμό για την ασφαλή επικοινωνία στα σύγχρονα δίκτυα και πληροφοριακά συστήματα. Αναπτύχθηκε από τους Whitfield Diffie και Martin Hellman το 1976 και επιτρέπει σε δύο μέρη να δημιουργήσουν ένα κοινό μυστικό κρυπτογραφικό κλειδί ακόμα και μέσω μη ασφαλούς καναλιού επικοινωνίας, χωρίς να χρειάζεται να αποσταλεί απευθείας αυτούσιο το κλειδί που τελικά θα δημιουργηθεί.

Η λειτουργία του πρωτοκόλλου Diffie-Hellman βασίζεται στη δυσκολία επίλυσης του προβλήματος του διακριτού λογαρίθμου σε πεπερασμένο χρόνο. Αυτή η μαθηματική ιδιότητα επιτρέπει στα δύο μέρη να υπολογίσουν ένα κοινό ιδιωτικό κρυπτογραφικό κλειδί, χωρίς ένας τρίτος παρατηρητής να μπορεί εύκολα να το αποκρυπτογραφήσει, ακόμη και αν έχει πρόσβαση στις πληροφορίες που ανταλλάσσονται μεταξύ των δύο μερών κατά τη διαδικασία δημιουργίας του κρυπτογραφικού κλειδιού.

Ωστόσο, η κλασική υλοποίηση του πρωτοκόλλου Diffie-Hellman έχει ορισμένες αδυναμίες, ιδιαίτερα όσον αφορά την ευαισθησία του σε επιθέσεις τύπου "man-in-the-middle". Για την αντιμετώπιση αυτών των ζητημάτων, έχουν προταθεί διάφορες παραλλαγές και βελτιώσεις του αρχικού πρωτοκόλλου. Μία σημαντική εξέλιξη είναι η χρήση ελλειπτικών καμπυλών στο πρωτόκολλο Diffie-Hellman, γνωστή ως ECDH. Η υλοποίηση του ECDH προσφέρει ισοδύναμη ασφάλεια με το κλασικό Diffie-Hellman, αλλά με μικρότερο μέγεθος κλειδιού, καθιστώντας το πιο αποδοτικό για συσκευές με περιορισμένους πόρους [37].

Στον τομέα του Διαδικτύου των Πραγμάτων (IoT), η εφαρμογή του ECDH έχει αποδειχθεί ιδιαίτερα χρήσιμη. Έρευνες έχουν επιδείξει βελτιωμένα αποτελέσματα στην υλοποίηση του ECDH στο λειτουργικό σύστημα Contiki, χρησιμοποιώντας την καμπύλη

SECP160K1, με σημαντική μείωση του υπολογιστικού χρόνου σε σύγκριση με προηγούμενες εργασίες [37].

Μια άλλη ενδιαφέρουσα προσέγγιση είναι η ενσωμάτωση του Diffie-Hellman σε πιο πολύπλοκα κρυπτογραφικά συστήματα. Για παράδειγμα, ερευνητές έχουν προτείνει ένα σύστημα κρυπτογράφησης εικόνας που συνδυάζει το τροποποιημένο πρωτόκολλο ανταλλαγής κλειδιών ελλειπτικής καμπύλης Diffie-Hellman με τον κρυπτογραφικό αλγόριθμο Hill Cipher [38]. Αυτή η προσέγγιση αποσκοπεί στην ενίσχυση της ασφάλειας ψηφιακών εικόνων, οι οποίες συχνά περιέχουν ευαίσθητες πληροφορίες σε επιχειρηματικά, ιατρικά και στρατιωτικά περιβάλλοντα.

Η βελτιστοποίηση του πρωτοκόλλου Diffie-Hellman αποτελεί επίσης ένα ενεργό πεδίο έρευνας. Μια πρόσφατη μελέτη παρουσίασε μια νέα έκδοση του πρωτοκόλλου, γνωστή ως Optimized Diffie-Hellman Key Exchange (ODHKE), η οποία βασίζεται σε βελτιστοποιημένες συναρτήσεις διακριτού λογαρίθμου και χρησιμοποιεί μια γραφική μέθοδο για τον υπολογισμό του κοινού μυστικού κλειδιού [39]. Αυτή η προσέγγιση υπόσχεται αυξημένη ασφάλεια σε σύγκριση με την αρχική εκδοχή του πρωτοκόλλου Diffie-Hellman.

Στον τομέα των ασύρματων δικτύων αισθητήρων (Wireless Sensors Networks - WSN), έχει προταθεί μια τροποποιημένη έκδοση του Diffie-Hellman για τη βελτίωση της ασφάλειας των κόμβων αισθητήρων. Η προσέγγιση αυτή, γνωστή ως Cryptanalysis with Modified Diffie-Hellman key exchange (CMDH), χρησιμοποιεί κατακερματισμό (hashing) για κάθε τιμή που αποστέλλεται μέσω του δικτύου, ενισχύοντας έτσι την αντίσταση σε επιθέσεις [40].

Η ασφάλεια του πρωτοκόλλου Diffie-Hellman έχει επίσης ενισχυθεί μέσω της ενσωμάτωσής του σε πιο πολύπλοκα συστήματα ασφαλείας. Για παράδειγμα, ερευνητές έχουν προτείνει ένα πρωτόκολλο ομαδικής ταυτοποίησης και δημιουργίας συμμετρικού κρυπτογραφικού κλειδιού που βασίζεται στο ECDH (Group Security Authentication and Key Agreement, GSAKA-ECDH Key Exchange) για επικοινωνίες LTE με στρατιωτική χρήση [41]. Αυτό το πρωτόκολλο αντιμετωπίζει τις αδυναμίες του πρωτοκόλλου Evolved Packet System based Authentication and Key Agreement (EPS-AKA) και παρέχει προστασία από διάφορες γνωστές επιθέσεις ασφαλείας. Τέλος, αξίζει να σημειωθεί ότι η έρευνα στον τομέα της κβαντικής υπολογιστικής έχει εγείρει ανησυχίες σχετικά με τη μακροπρόθεσμη ασφάλεια του Diffie-Hellman και άλλων κλασικών κρυπτογραφικών αλγορίθμων. Ως εκ τούτου, η ανάπτυξη κβαντικά ανθεκτικών εκδόσεων του Diffie-Hellman αποτελεί ένα σημαντικό πεδίο τρέχουσας έρευνας.

Συμπερασματικά, το πρωτόκολλο ανταλλαγής κλειδιών Diffie-Hellman παραμένει ένα θεμελιώδες εργαλείο στην κρυπτογραφία και την ασφάλεια των δικτυακών επικοινωνιών. Οι συνεχείς έρευνες και βελτιώσεις του πρωτοκόλλου αντανακλούν τη σημασία του στην αντιμετώπιση των αναδυόμενων προκλήσεων ασφαλείας στο πεδίο των δικτυακών υπηρεσιών.

5. Η πλατφόρμα Δικτύωσης Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti

5.1 Βασικά στοιχεία της πλατφόρμας Δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti

Το OpenZiti (“open-source Zero-Trust”) αποτελεί μια πλατφόρμα ανοιχτού κώδικα για την υλοποίηση Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης. Τα συστατικά στοιχεία του OpenZiti μπορούν να κατηγοριοποιηθούν σε δύο βασικές ομάδες, τις λογικές οντότητες και τα στοιχεία λογισμικού (deployed components), τα οποία και παρουσιάζονται στις δύο επόμενες υποενότητες της παρούσας Διπλωματικής Εργασίας.

5.1.1 Λογικές οντότητες της πλατφόρμας Δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti

Στην κατηγορία των λογικών οντοτήτων, το OpenZiti περιλαμβάνει τρεις βασικές έννοιες: τις Ταυτότητες, τις Υπηρεσίες και τις Πολιτικές. Οι Ταυτότητες (identities) αντιπροσωπεύουν κάθε διακριτό χρήστη/εφαρμογή/συσκευή στα πλαίσια ενός δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti. Οι Υπηρεσίες (Services) αποτελούν αφαιρετικές αναπαραστάσεις πόρων που θα μπορούσαν να προσπελαστούν από μια εφαρμογή πελάτη σε ένα οποιοδήποτε δίκτυο. Η πρόσβαση στις παραπάνω υπηρεσίες των δικτύων Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti αντί να ορίζονται με βάση την υποκείμενη δικτυακή υποδομή, όπως διευθύνσεις IP ή ονόματα DNS (Domain Name System), ορίζονται με βάση παραμετροποιήσεις που ορίζει το εκάστοτε δίκτυο OpenZiti.

Οι διαμορφώσεις (configurations) των υπηρεσιών των δικτύων Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti διακρίνονται σε δύο είδη, intercept και host. Οι διαμορφώσεις (configurations) καθορίζουν τον συνδυασμό πρωτοκόλλου στρώματος μεταφοράς (Transport Control Protocol - TCP/User Datagram Protocol - UDP), idn hostname/address και θύρας τον οποίο οι εφαρμογές χρησιμοποιούν προκειμένου να αποκτήσουν πρόσβαση σε μια υπηρεσία OpenZiti. Οι διαμορφώσεις (configurations) που συνδέονται με τις υπηρεσίες OpenZiti χρησιμοποιούνται ως διακριτικά από τις εφαρμογές πελάτη και διακομιστή για να ενημερώνονται σχετικά με τους δρομολογητές “άκρου” με τους οποίους πρέπει να συνδεθούν για dial/bind κάθε υπηρεσίας OpenZiti. Με αυτό τον τρόπο καθορίζουν και ποια δικτυακή κίνηση πρέπει να δρομολογηθεί μέσω δικτύου Μηδενικής Εμπιστοσύνης OpenZiti και ποια όχι. Στο πλαίσιο των πρωτοκόλλων, οι ρυθμίσεις τύπου intercept καθορίζουν συνήθως το TCP ως πρωτόκολλο στρώματος μεταφοράς. Ωστόσο, υπάρχει ευελιξία για την υποστήριξη πρωτοκόλλου UDP ανάλογα με τις απαιτήσεις της εκάστοτε υλοποίησης.

Αξίζει να σημειωθεί ότι οι παραπάνω διαμορφώσεις (configurations) υφίστανται με τη λογική τα αντίστοιχα idn hostnames να είναι αναλύσιμα μόνο από ένα δίκτυο OpenZiti και τυχόν εφαρμογές tunnel (ή άλλα στοιχεία δικτύου) που περιλαμβάνει, καθιστώντας κατ’ αυτόν τον τρόπο τις υπηρεσίες ενός δικτύου Μηδενικής Εμπιστοσύνης OpenZiti αόρατες (“dark” όπως αναφέρεται στο documentation της πλατφόρμας) και κατ’ επέκταση απροσπέλαστες από μη ταυτοποιημένους χρήστες/εφαρμογές/συσκευές, προσφέροντας έτσι ένα ιδιωτικό ταυτοποιημένο DNS. Ως τμήμα κάθε διαμόρφωσης (configuration) (intercept/host) μπορεί να αναφερθεί και η αντίστοιχη “διεύθυνση” (intercept/host address).

Στην εικόνα 5.1 παρατίθεται παράδειγμα διαμόρφωσης τύπου “intercept”, υπηρεσίας δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti, σε μορφή JSON.

Στις περιπτώσεις όπου η εφαρμογή που λειτουργεί ως διακομιστής δεν ενσωματώνει κώδικα των SDKs της πλατφόρμας Δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti η διαμόρφωση (configuration) τύπου “host” χρησιμοποιείται για την ενημέρωση του τερματικού στοιχείου (δρομολογητής ή εφαρμογή tunnel) του δικτύου αναφορικά με το με ποια διεύθυνση IP και θύρα θα πρέπει να πραγματοποιήσει σύνδεση TCP. Οι παραπάνω συνδέσεις TCP χρησιμοποιούνται για τη μεταφορά των αιτημάτων (requests) από το εκάστοτε δίκτυο OpenZit προς τις εφαρμογές που λειτουργούν ως διακομιστές και αντίστοιχα των αποκρίσεών τους προς την εκάστοτε εφαρμογή πελάτη, μέσω του δικτύου OpenZiti.

```
{
    "protocols": [
        "tcp"
    ],
    "addresses": [
        "acme.example.ziti"
    ],
    "portRanges": [
        {
            "low": 5000,
            "high": 5000
        }
    ]
}
```

Εικόνα 5.1: Παράδειγμα διαμόρφωσης (configuration) υπηρεσίας OpenZiti

Στην περίπτωση που μια εφαρμογή που λειτουργεί ως διακομιστής, ενσωματώνει κώδικα βιβλιοθήκης της πλατφόρμας Μηδενικής Εμπιστοσύνης OpenZiti, η διεύθυνση “host” χρησιμοποιείται για τη δημιουργία mTLS σύνδεσης με κατάλληλο δρομολογητή “άκρου” του δικτύου OpenZiti, για τη συγκεκριμένη υπηρεσία στην οποία αντιστοιχεί. Η διενέργεια της παραπάνω σύνδεσης από την πλευρά της εφαρμογής που λειτουργεί ως διακομιστής, εξαλείφει την ανάγκη για κάποιο δημοσιοποιημένο, αναλύσιμο idn hostname και είναι το δεύτερο σκέλος (μαζί με τις διευθύνσεις τύπου “intercept” όπως αναλύθηκαν παραπάνω) που καθιστά τις υπηρεσίες OpenZiti αόρατες (“dark”), αφού δεν υφίσταται ανάγκη για κάποια ανοικτή θύρα στην οποία να “ακούει” η εφαρμογή, η οποία θα την καθιστούσε δυνητικά ευάλωτη σε επιθέσεις.

Οι Πολιτικές (Policies) καθορίζουν ποιες ταυτότητες ενός δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti έχουν δικαίωμα πρόσβασης είτε ως "πελάτες" είτε ως διακομιστές, σε ποιες υπηρεσίες και μέσω ποιων δρομολογητών. Οι Πολιτικές περιλαμβάνουν διάφορους υποτύπους. Τα "χαρακτηριστικά ρόλων" (Role Attributes) είναι απλές συμβολοσειρές που μπορούν να αποδοθούν σε λογικές οντότητες ενός δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti όπως ταυτότητες, υπηρεσίες και δρομολογητές. Οι Πολιτικές Υπηρεσιών (Service Policies) καθορίζουν τα δικαιώματα πρόσβασης των ταυτοτήτων των δικτύων Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti στις υπηρεσίες του αντίστοιχου δικτύου. Οι Πολιτικές των δρομολογητών "άκρου" (Edge Router Policies) διαχειρίζονται τον έλεγχο πρόσβασης των ταυτοτήτων στους δρομολογητές που έχουν ενεργοποιημένη τη σχετική λειτουργία ("edge") και κατ' επέκταση μπορούν να αποτελούν σημείο εισόδου στο δίκτυο (δηλαδή επιτρέπουν συνδέσεις από εφαρμογές πελάτη αν υπάρχει κατάλληλη εξουσιοδότηση - authorization). Τέλος, οι Πολιτικές Υπηρεσιών των δρομολογητών "άκρου" (Service Edge Router Policies) καθορίζουν ποιοι εξ αυτών μπορούν να χρησιμοποιούνται για να δρομολογούν κίνηση συγκεκριμένων υπηρεσιών του δικτύου.

Οι συνεδρίες υπηρεσιών (service sessions) αποτελούν ένα ακόμα κρίσιμο στοιχείο της Αρχιτεκτονικής Ασφαλείας του OpenZiti. Πρόκειται για λογικές οντότητες που αντιπροσωπεύουν την εξουσιοδότηση μιας εφαρμογής πελάτη να αποκτήσει πρόσβαση σε μια συγκεκριμένη υπηρεσία ενός δικτύου OpenZiti. Όταν μια εφαρμογή πελάτη επιθυμεί να συνδεθεί σε μια υπηρεσία, πρέπει πρώτα να δημιουργήσει μια συνεδρία υπηρεσίας. Αυτή η διαδικασία περιλαμβάνει την αποστολή ενός αιτήματος στον ελεγκτή (controller) του δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti, ο οποίος στη συνέχεια αξιολογεί εάν η εφαρμογή πελάτη έχει τα απαραίτητα δικαιώματα για να αποκτήσει πρόσβαση στην υπηρεσία. Εάν η εξουσιοδότηση είναι επιτυχής, ο ελεγκτής δημιουργεί μια συνεδρία υπηρεσίας και επιστρέφει ένα διακριτικό για τη συνεδρία υπηρεσίας (session token) στην εφαρμογή πελάτη.

Το διακριτικό (token) μιας συνεδρίας υπηρεσίας είναι ένα κρίσιμο στοιχείο ασφάλειας. Αντιστοιχεί στην ταυτότητα που χρησιμοποιεί η εφαρμογή πελάτη, την υπηρεσία στην οποία επιτρέπεται η πρόσβαση και τυχόν περιορισμούς ή πρόσθετες παραμέτρους που σχετίζονται με τη συνεδρία. Αυτό το διακριτικό χρησιμοποιείται στη συνέχεια από την εφαρμογή πελάτη για να αποδείξει την εξουσιοδότησή του κατά τη σύνδεση με την υπηρεσία μέσω των δρομολογητών "άκρου" ενός δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti.

Ένα σημαντικό χαρακτηριστικό των συνεδριών υπηρεσιών είναι η περιορισμένη διάρκεια ζωής τους. Κάθε συνεδρία έχει ένα προκαθορισμένο χρόνο λήξης, μετά τον οποίο παύει να ισχύει. Αυτό ενισχύει την ασφάλεια του συστήματος, καθώς περιορίζει το χρονικό περιθώριο κατά το οποίο ένα κλεμμένο ή παραβιασμένο διακριτικό συνεδρίας θα μπορούσε να χρησιμοποιηθεί από έναν κακόβουλο χρήστη. Επιπλέον, οι συνεδρίες υπηρεσιών στα δίκτυα Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti υποστηρίζουν δυνατότητα ανάκλησης. Αυτό σημαίνει ότι ο ελεγκτής (controller) του δικτύου μπορεί να ακυρώσει μια ενεργή συνεδρία ανά πάσα στιγμή, εάν ανιχνευθεί ύποπτη δραστηριότητα ή εάν υπάρξει σχετική αλλαγή στις πολιτικές πρόσβασης. Αυτή η δυνατότητα προσφέρει ένα επιπλέον επίπεδο ελέγχου και ασφάλειας στα δίκτυα OpenZiti.

Η αρχιτεκτονική των συνεδριών υπηρεσιών του OpenZiti είναι σχεδιασμένη να υποστηρίζει υψηλή κλιμάκωση και επιδόσεις. Οι πληροφορίες συνεδρίας αποθηκεύονται κατανεμημένα σε όλο το δίκτυο, επιτρέποντας γρήγορη επαλήθευση και αποτελεσματική

διαχείριση ακόμη και σε περιβάλλοντα με μεγάλο αριθμό ταυτόχρονων συνδέσεων. Ένα άλλο σημαντικό χαρακτηριστικό των συνεδριών υπηρεσιών στο OpenZiti είναι η υποστήριξη για πολλαπλές ταυτόχρονες συνδέσεις. Μια εφαρμογή πελάτη μπορεί να χρησιμοποιήσει το ίδιο διακριτικό συνεδρίας για να δημιουργήσει πολλαπλές συνδέσεις με την ίδια υπηρεσία, χωρίς να χρειάζεται να επαναλάβει τη διαδικασία εξουσιοδότησης για κάθε σύνδεση. Η δυνατότητα αυτή βελτιώνει σημαντικά τις επιδόσεις και την αποτελεσματικότητα του συστήματος.

Οι συνεδρίες υπηρεσιών παίζουν επίσης σημαντικό ρόλο στην υλοποίηση της αρχής Ελάχιστων Προνομίων (principle of least privilege) στο OpenZiti. Κάθε συνεδρία παρέχει πρόσβαση μόνο στη συγκεκριμένη υπηρεσία για την οποία έχει εξουσιοδοτηθεί, περιορίζοντας έτσι το εύρος της πρόσβασης σε περίπτωση παραβίασης.

Στο πλαίσιο της αρχιτεκτονικής ασφάλειας της πλατφόρμας OpenZiti, τα χαρακτηριστικά ρόλων (role attributes) λειτουργούν ως δυναμικοί μηχανισμοί καθορισμού και εφαρμογής πολιτικών εξουσιοδότησης, με στόχο την αυστηρή τήρηση των αρχών του μοντέλου Μηδενικής Εμπιστοσύνης. Κάθε χαρακτηριστικό ρόλου ορίζεται μέσω ενός συνόλου ιδιοτήτων που αποτελούν κριτήρια για την αντιστοίχιση ταυτοτήτων (identities) σε υπηρεσίες ή πόρους, με βάση συγκεκριμένες συνθήκες και παραμέτρους.

Τα χαρακτηριστικά ρόλων συνδέονται άμεσα με τις πολιτικές εξουσιοδότησης (authorization policies), οι οποίες επιβάλλουν μια σχέση μεταξύ ταυτοτήτων και πόρων, με βάση λογικούς συνδυασμούς ιδιοτήτων. Για παράδειγμα, μια ταυτότητα μπορεί να αποκτήσει πρόσβαση σε μια υπηρεσία μόνο εάν διαθέτει συγκεκριμένες ετικέτες (tags) ή αν πληροί προκαθορισμένες συνθήκες, όπως π.χ. η χρήση ταυτοποίησης πολλαπλών παραγόντων (multi-factor authentication).

Η διαδικασία αυτή βασίζεται στην αρχή Ελάχιστων Προνομίων (least privilege access), όπου κάθε οντότητα λαμβάνει μόνο τα δικαιώματα που είναι απολύτως απαραίτητα για την εκπλήρωση των λειτουργιών της, χωρίς επιπλέον δικαιώματα πρόσβασης που θα μπορούσαν να αυξήσουν τον κίνδυνο επιθέσεων. Παράλληλα, η χρήση κατάτμησης πόρων ενισχύεται μέσω της αυστηρής διαχωριστικής λογικής των χαρακτηριστικών ρόλων, περιορίζοντας την κίνηση εντός του δικτύου μόνο σε εξουσιοδοτημένες αλληλεπιδράσεις και απομονώνοντας κρίσιμους πόρους από μη εξουσιοδοτημένα αιτήματα.

Συνοπτικά, τα χαρακτηριστικά ρόλων στο OpenZiti δεν αποτελούν απλές στατικές ρυθμίσεις, αλλά δυναμικά στοιχεία που ενσωματώνουν συνεχή αξιολόγηση και προσαρμογή. Μέσω αυτών, εξασφαλίζεται ότι οι πολιτικές πρόσβασης παραμένουν ευέλικτες και ανταποκρίνονται σε πραγματικούς κινδύνους, διατηρώντας παράλληλα υψηλά επίπεδα κλιμακωσιμότητας και δυνατότητας διαχείρισης σε πολύπλοκα, κατανεμημένα περιβάλλοντα.

5.1.2 Στοιχεία λογισμικού της πλατφόρμας Δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti

Όσον αφορά τα στοιχεία λογισμικού και ανάπτυξης, το OpenZiti περιλαμβάνει τρία βασικά συστατικά: τον Ελεγκτή (controller), τους Δρομολογητές (Routers), τους "πελάτες άκρου" (Edge Clients). Ο Ελεγκτής (Controller) εκτελεί τις πιο κεντρικές και ζωτικές λειτουργίες ενός δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti. Είναι

υπεύθυνος για τη διαχείριση των υπηρεσιών και των ταυτοτήτων που χρησιμοποιούνται από χρήστες, εφαρμογές και συσκευές του δικτύου. Επιπλέον, ο ελεγκτής είναι υπεύθυνος για την ταυτοποίηση και εξουσιοδότηση κάθε σύνδεσης που πραγματοποιείται στα πλαίσια ενός δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti. Προς επίτευξη του παραπάνω χρησιμοποιεί υποδομή δημόσιου κλειδιού (PKI) για την έκδοση πιστοποιητικών X.509 με σκοπό τη δημιουργία ασφαλών, αμοιβαία ταυτοποιημένων συνδέσεων TLS μεταξύ οποιωνδήποτε δύο κόμβων του δικτύου. Διατηρεί τοπική βάση δεδομένων βασισμένη στη bbolt για την αποθήκευση των λογικών οντοτήτων που σχετίζονται με το δίκτυο Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης και η λειτουργία του καθορίζεται από ένα αρχείο YAML.

Οι Δρομολογητές (Routers) αποτελούν επίσης θεμελιώδη δομικά στοιχεία κάθε δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti. Είναι υπεύθυνοι για την ασφαλή και αξιόπιστη προώθηση της κίνησης από έναν κόμβο δικτύου προς τον προορισμό της. Οι Δρομολογητές των δικτύων OpenZiti συνδέονται μεταξύ τους για να σχηματίσουν ένα "πλέγμα" δικτύου (mesh network), το οποίο παρακολουθείται συνεχώς για καθυστερήσεις, επιτρέποντας τη χρήση των ταχύτερων διαδρομών και την ενεργή εναλλαγή σε περίπτωση προβλήματος σε οποιονδήποτε κόμβο. Όσοι εκ των δρομολογητών ενός δικτύου Μηδενικής Εμπιστοσύνης OpenZiti έχουν τη λειτουργία "edge" ενεργοποιημένη, ονομάζονται δρομολογητές "άκρου" ("Edge-Routers") και αποτελούν αρκετά κρίσιμο δομικό στοιχείο, αφού αποτελούν δυνητικά σημείο εισόδου το δίκτυο και επιτρέπουν συνδέσεις από τους "πελάτες άκρου" (Edge-Clients), που περιγράφονται παρακάτω.

Οι "πελάτες άκρου" (Edge Clients) αποτελούν στοιχεία λογισμικού απαραίτητα για τη σύνδεση σε δίκτυα OpenZiti. Μπορούν να διαχωριστούν σε δύο κατηγορίες, τις εφαρμογές πελάτη ή διακομιστή που ενσωματώνουν κώδικα των SDKs της πλατφόρμας OpenZiti και τις έτοιμες λύσεις εφαρμογών tunnel που προσφέρει η πλατφόρμα. Τα SDKs που προσφέρει η πλατφόρμα καλύπτουν διάφορες γλώσσες προγραμματισμού και περιβάλλοντα εκτέλεσης για να παρέχουν γρήγορη, αξιόπιστη και ασφαλή συνδεσιμότητα. Για υπάρχουσες λύσεις, που δεν καθίσταται δυνατόν να τροποποιηθούν η πλατφόρμα OpenZiti προσφέρει τις προαναφερθείσες εφαρμογές tunnel, οι οποίες παρέχουν ασφαλή συνδεσιμότητα χωρίς να απαιτούνται αλλαγές στις εφαρμογές πελάτη ή/και διακομιστή.

Οι εφαρμογές tunnel αποτελούν ένα σημαντικό συστατικό της πλατφόρμας OpenZiti για την υιοθέτηση Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης. Ο βασικός στόχος των εφαρμογών αυτών είναι να παρεμβάλλονται στη δικτυακή κίνηση που προορίζεται για υπηρεσίες ενός δικτύου OpenZiti και να την προωθούν μέσω του δικτύου OpenZiti και όχι μέσω της υποκείμενης δικτυακής υποδομής. Η λειτουργία των εφαρμογών tunnel βασίζεται στη διαμόρφωση ενός διακομιστή ονομάτων DNS (nameserver) για την επίλυση (resolve) των διευθύνσεων των υπηρεσιών OpenZiti. Ένα σημαντικό χαρακτηριστικό των εφαρμογών tunnel είναι ότι επιτρέπουν σε όλες τις διεργασίες του συστήματος να έχουν πρόσβαση στις υπηρεσίες OpenZiti που είναι διαθέσιμες στις ταυτότητες που χρησιμοποιεί η εφαρμογή tunnel (προσφέρουν δηλαδή host-access).

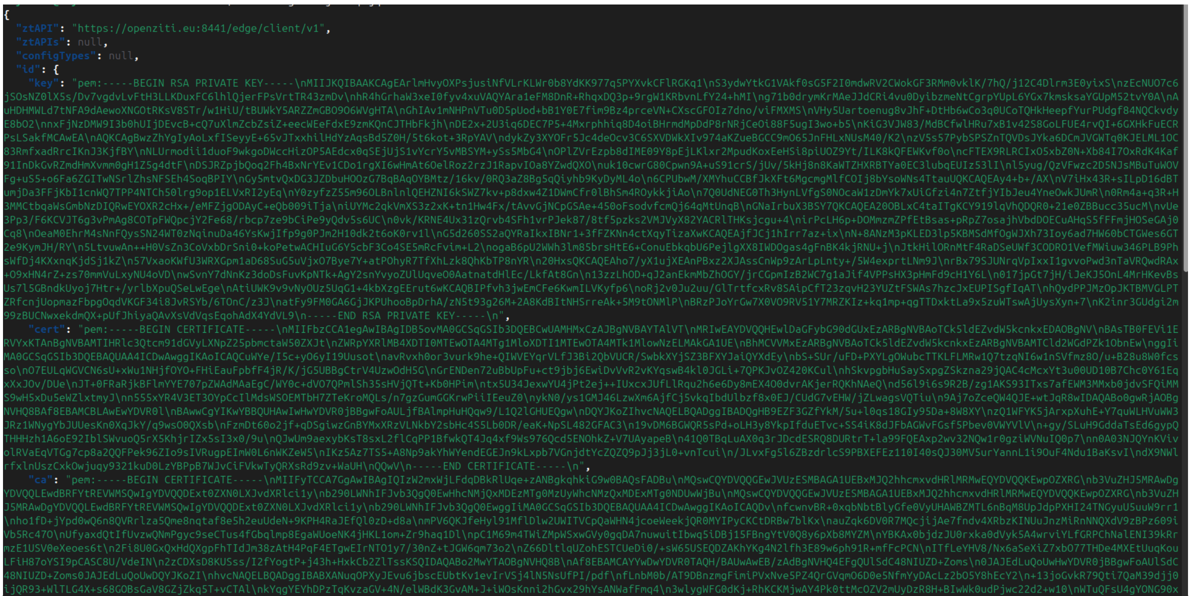
Οι εφαρμογές tunnel του OpenZiti υποστηρίζουν διάφορα λειτουργικά συστήματα και πλατφόρμες, συμπεριλαμβανομένων των Windows, macOS, Linux και iOS. Κάθε εφαρμογή tunnel είναι προσαρμοσμένη στις ιδιαιτερότητες του εκάστοτε λειτουργικού συστήματος, αλλά όλες λειτουργούν με παρόμοιο τρόπο για να επιτύχουν τον ίδιο στόχο. Κοινό χαρακτηριστικό όλων των εφαρμογών tunnel της πλατφόρμας OpenZiti αποτελεί η δημιουργία εικονικών διεπαφών δικτύου και προσάρτηση σε αυτές ιδιωτικών διευθύνσεων IP. Πιο συγκεκριμένα οι διευθύνσεις που αντιστοιχούν σε υπηρεσίες ενός δικτύου

Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti αναλύονται στις παραπάνω διευθύνσεις IP μέσω των οποίων η εκάστοτε εφαρμογή tunnel προωθεί την κίνηση προς το δίκτυο.

Για λειτουργικά συστήματα Linux, η εφαρμογή tunnel χρησιμοποιεί το πρωτόκολλο TUN/TAP για να δημιουργήσει μια εικονική διεπαφή δικτύου. Αυτό επιτρέπει στην εφαρμογή να παρεμβάλλεται στην κίνηση δικτύου σε επίπεδο πυρήνα (kernel) και να τη δρομολογεί μέσω δικτύου OpenZiti. Ένα κοινό χαρακτηριστικό όλων των εφαρμογών tunnel της πλατφόρμας OpenZiti είναι η ικανότητά τους να διαχειρίζονται πολλαπλές ταυτότητες. Αυτό σημαίνει ότι κάθε εφαρμογή tunnel μπορεί δυνητικά να παρέχει πρόσβαση σε πολλαπλές υπηρεσίες δικτύων OpenZiti, με διαφορετικές πολιτικές ασφαλείας και δικαιώματα πρόσβασης μεταξύ τους.

Προκειμένου να χρησιμοποιεί μια εφαρμογή tunnel (ή και ένας SDK-enabled “πελάτης άκρου” γενικότερα) μια οποιαδήποτε ταυτότητα ενός δικτύου OpenZiti, χρησιμοποιεί κατάλληλο αρχείο “ταυτότητας” (identity file). Τα αρχεία “ταυτότητας” της πλατφόρμας Δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti αποτελούνται από ορισμένα κρίσιμα πεδία. Κατ’ αρχάς περιέχουν το Uniform Resource Locator (URL) το οποίο οι αντίστοιχες εφαρμογές χρησιμοποιούν για επικοινωνία με τον ελεγκτή (controller) του δικτύου Μηδενικής Εμπιστοσύνης OpenZiti με σκοπό την ταυτοποίηση του χρήστη ή της εφαρμογής και τη μετέπειτα χρήση του client API ή του API διαχείρισης.

Παράλληλα, περιέχεται το ψηφιακό πιστοποιητικό τύπου X.509 και το ιδιωτικό κλειδί του χρήστη τον οποίο ταυτοποιεί το εκάστοτε αρχείο ταυτότητας, αλλά και έμπιστα πιστοποιητικά της Αρχής Πιστοποίησης (CA) του δικτύου OpenZiti, τα δημόσια κλειδιά των οποίων χρησιμοποιούνται για την επικύρωση των πιστοποιητικών τόσο του ίδιου του ελεγκτή (controller) του δικτύου, όσο και οποιουδήποτε δρομολογητή “άκρου” με τον οποίο θα επιδιώξει να συνδεθεί ο εκάστοτε “πελάτης άκρου” (Edge-Client). Στην εικόνα 5.2 παρουσιάζεται ενδεικτικά το περιεχόμενο ενός αρχείου ταυτότητας OpenZiti.



Εικόνα 5.2: Παράδειγμα αρχείου ταυτότητας OpenZiti

Οι εφαρμογές tunnel της πλατφόρμας OpenZiti παρέχουν ένα ευέλικτο και ισχυρό μέσο για την ενσωμάτωση υπαρχόντων εφαρμογών και συστημάτων σε ένα ασφαλές “επικαλύπτον” δίκτυο (overlay network), προσφέροντας παράλληλα προηγμένες

δυνατότητες ασφάλειας και ελέγχου πρόσβασης, όπως τη χρήση εφήμερων κωδικών (TOTP). Η ικανότητά τους να υποστηρίζουν πολλαπλές εφαρμογές τις καθιστά ιδανικές για υλοποιήσεις Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης σε ποικίλα περιβάλλοντα και σενάρια χρήσης.

Συνοψίζοντας, το OpenZiti παρέχει ένα ολοκληρωμένο σύνολο λογικών οντοτήτων και στοιχείων λογισμικού/ανάπτυξης που επιτρέπουν την υιοθέτηση Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης. Ο συνδυασμός αυτών των δομικών στοιχείων επιτρέπει τη δημιουργία ενός ασφαλούς, ευέλικτου και επεκτάσιμου δικτύου που βασίζεται στις θεμελιώδεις αρχές της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης.

5.2 Δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti

Τα δίκτυα OpenZiti αποτελούν μια σύγχρονη προσέγγιση στην υλοποίηση δικτυακών υποδομών, βασισμένη στην έννοια του "επικαλύπτοντος δικτύου" (network overlay). Χρησιμοποιώντας τις λογικές οντότητες και τα εργαλεία λογισμικού που περιγράφονται στις δύο προηγούμενες υποενότητες, η αρχιτεκτονική εισάγει ένα νέο επίπεδο αφαίρεσης πάνω από τα παραδοσιακά δικτυακά στρώματα, προσφέροντας ένα καινοτόμο σύνολο εργαλείων και μεθόδων για τον σχεδιασμό και την υλοποίηση λογισμικού και συστημάτων.

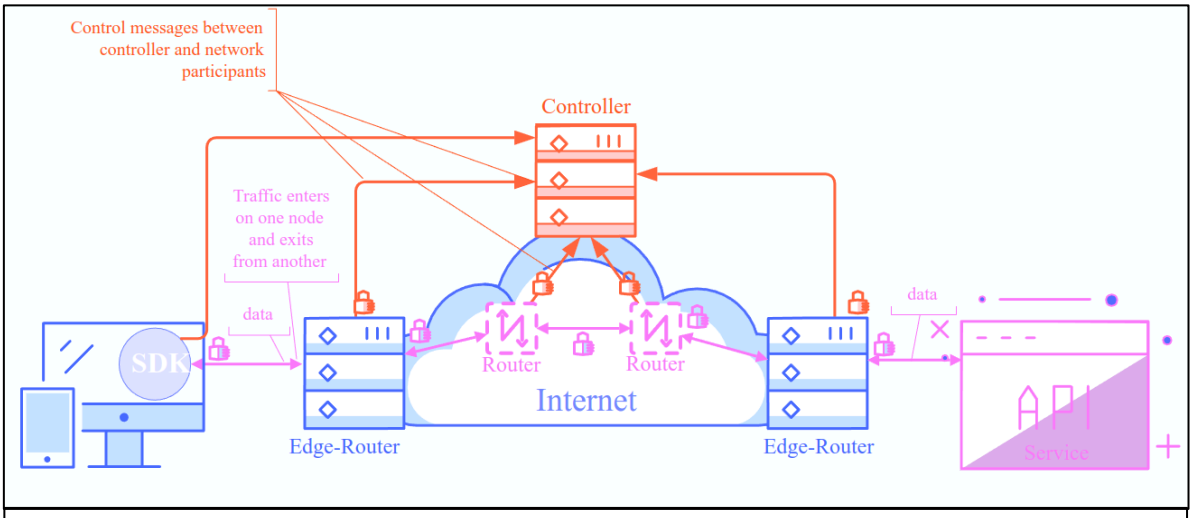
Το δίκτυο OpenZiti λειτουργεί ως ένα στρώμα αφαίρεσης που αποκρύπτει την πολυπλοκότητα των υποκείμενων δικτυακών επιπέδων. Αυτή η προσέγγιση επιτρέπει στους προγραμματιστές να επικεντρωθούν στη λογική υψηλού επιπέδου των εφαρμογών τους, χωρίς να χρειάζεται να ασχοληθούν με τις λεπτομέρειες χαμηλού επιπέδου της δικτυακής επικοινωνίας. Η αφαίρεση αυτή είναι ιδιαίτερα σημαντική στο σύγχρονο περιβάλλον ανάπτυξης λογισμικού, όπου η πολυπλοκότητα των δικτυακών υποδομών συνεχώς αυξάνεται.

Ένα από τα κύρια πλεονεκτήματα του επικαλύπτοντος (overlay) δικτύου OpenZiti είναι η δυνατότητα που παρέχει στους προγραμματιστές να εστιάσουν στη συνδεσιμότητα μεταξύ των διαφόρων στοιχείων μιας εφαρμογής ή ενός συστήματος. Αυτό επιτυγχάνεται χωρίς να απαιτείται βαθιά γνώση ή διαχείριση των λεπτομερειών χαμηλού επιπέδου που αφορούν τον τρόπο με τον οποίο επιτυγχάνεται αυτή η συνδεσιμότητα. Η προσέγγιση αυτή όχι μόνο απλοποιεί τη διαδικασία ανάπτυξης, αλλά επίσης επιτρέπει τη δημιουργία πιο ευέλικτων και προσαρμόσιμων εφαρμογών.

Η αρχιτεκτονική του επικαλύπτοντος δικτύου ενσωματώνει προηγμένες τεχνικές διαχείρισης δικτύου, όπως δυναμική δρομολόγηση, αυτόματη εξισορρόπηση φορτίου (load balancing) και προσαρμοστική βελτιστοποίηση των επιδόσεων. Αυτές οι λειτουργίες εκτελούνται αυτόματα στο παρασκήνιο, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στη λογική των εφαρμογών τους, χωρίς να ανησυχούν για τις τεχνικές λεπτομέρειες της δικτυακής υποδομής.

Επιπλέον, το δίκτυο OpenZiti παρέχει ενισχυμένη ασφάλεια και ιδιωτικότητα. Ενσωματώνει προηγμένες τεχνικές κρυπτογράφησης και ελέγχου πρόσβασης, οι οποίες εφαρμόζονται σε όλο το εύρος του δικτύου. Αυτό το χαρακτηριστικό είναι ιδιαίτερα σημαντικό στο σύγχρονο περιβάλλον, όπου η ασφάλεια των δεδομένων και η προστασία της ιδιωτικότητας αποτελούν κρίσιμες προτεραιότητες.

Η ευελιξία των δικτύων OpenZiti επεκτείνεται και στην υποστήριξη διαφόρων τύπων δικτυακών τοπολογιών και αρχιτεκτονικών. Μπορεί να προσαρμοστεί εύκολα σε διάφορα σενάρια, από παραδοσιακά κεντρικοποιημένα δίκτυα μέχρι σύγχρονες κατανεμημένες και υβριδικές αρχιτεκτονικές. Αυτή η προσαρμοστικότητα καθιστά το OpenZiti ιδανικό για ένα ευρύ φάσμα εφαρμογών, από μικρές έως μεγάλες επιχειρήσεις και πολύπλοκα συστήματα διαδικτύου των πραγμάτων (IoT). Στην εικόνα 5.3 παρουσιάζεται σχηματικά ενδεικτική τοπολογία δικτύου OpenZiti.



Εικόνα 5.3: Παράδειγμα τοπολογίας δικτύου που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης χρησιμοποιώντας την πλατφόρμα OpenZiti

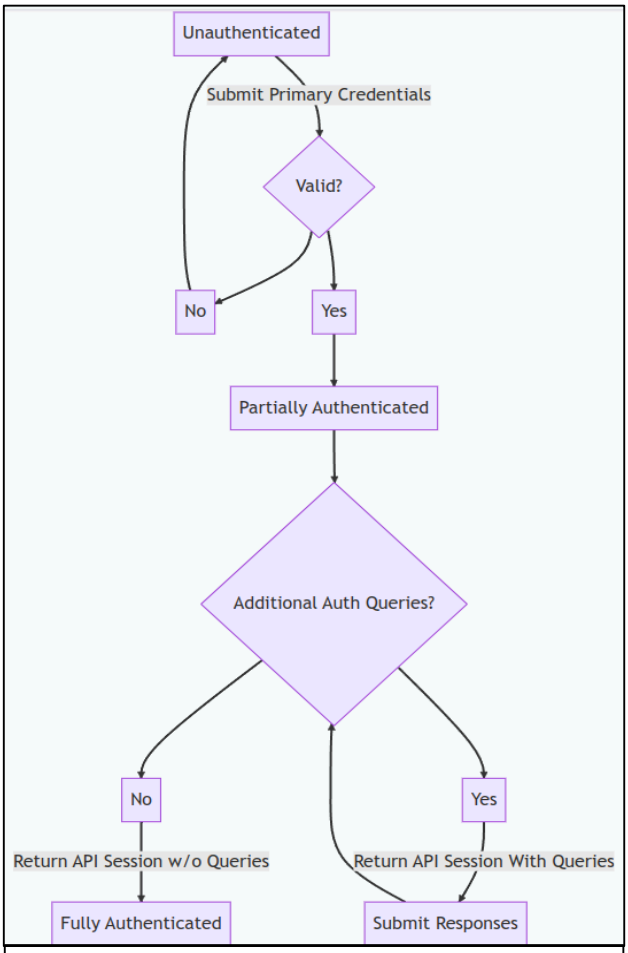
Συμπερασματικά, τα δίκτυα OpenZiti, μέσω της υλοποίησης του "επικαλύπτοντος δικτύου" (overlay network), προσφέρουν μια ισχυρή και ευέλικτη πλατφόρμα για την ανάπτυξη σύγχρονων δικτυακών εφαρμογών. Παρέχοντας ένα υψηλό επίπεδο αφαίρεσης, επιτρέπουν στους προγραμματιστές να επικεντρωθούν στη λειτουργία των εφαρμογών τους, ενώ παράλληλα εξασφαλίζουν υψηλές επιδόσεις, ασφάλεια και προσαρμοστικότητα. Αυτή η προσέγγιση αντιπροσωπεύει ένα σημαντικό βήμα προς την κατεύθυνση της απλοποίησης και βελτιστοποίησης της ανάπτυξης και διαχείρισης σύγχρονων δικτυακών υποδομών.

5.3 Ταυτοποίηση σε δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti

Οι εφαρμογές πελάτη μπορούν να χρησιμοποιήσουν τρεις μηχανισμούς για να πραγματοποιήσουν ταυτοποίηση χρήστη σε δίκτυα Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti: ψηφιακά πιστοποιητικά X.509, JWTs και συνδυασμούς username/password. Ένας ταυτοποιημένος χρήστης/εφαρμογή σε ένα δίκτυο που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης μέσω της πλατφόρμας OpenZiti αντιπροσωπεύεται από μια δομή δεδομένων που αναπαρίσταται ως ένα JSON object και ονομάζεται συνεδρία (session) API. Η διαδικασία ταυτοποίησης ξεκινά με την αποστολή στον ελεγκτή του δικτύου OpenZiti, αιτήματος ταυτοποίησης (authentication request), από τον εκάστοτε "πελάτη άκρου" (Edge-Client), το οποίο ενσωματώνει τα κατάλληλα credentials που αναφέρθηκαν παραπάνω.

Μια συνεδρία API μπορεί να αντιπροσωπεύει ένα χρήστη/εφαρμογή που έχει ταυτοποιηθεί για να χρησιμοποιεί κάποια από τις προγραμματιστικές διεπαφές

(client/management API) του ελεγκτή (controller) ενός δικτύου OpenZiti. Στην εικόνα 5.4 παρουσιάζεται το διάγραμμα ροής της διαδικασίας ταυτοποίησης σε ένα δίκτυο OpenZiti για χρήστη/εφαρμογή που επιδιώκει να ταυτοποιηθεί, ώστε να μπορεί να χρησιμοποιεί είτε το client API είτε το API διαχείρισης του δικτύου. Η παραπάνω εξουσιοδότηση μπορεί να είναι πλήρης (full API access) ή περιορισμένη (limited API access) στην περίπτωση που απαιτούνται και δευτερεύοντα στοιχεία ταυτοποίησης (JWT/TOTP) τα οποία δεν έχουν ακόμη υποβληθεί στον ελεγκτή (controller) του δικτύου OpenZiti.



Εικόνα 5.4: Διάγραμμα ροής της διαδικασίας ταυτοποίησης σε δίκτυα Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti


```
{
    "createdAt":
"2022-05-20T14:02:53.359Z",
    "id": "default",
    "tags": {},
    "updatedAt":
"2022-05-20T14:02:53.359Z",
    "name": "Default",
    "primary": {
        "cert": {
            "allowExpiredCerts": true,
            "allowed": true
        },
        "extJwt": {
            "allowed": true,
            "allowedSigners": null
        },
        "updb": {
            "allowed": true,
            "lockoutDurationMinutes": 0,
            "maxAttempts": 0
        }
    },
    "secondary": {
        "requireTotp": false,
        "requireExtJwt": ""
    }
}
```


Εικόνα 5.5: Παράδειγμα πολιτικής ταυτοποίησης σε δίκτυο Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti (αναπαρίσταται ως JSON object)

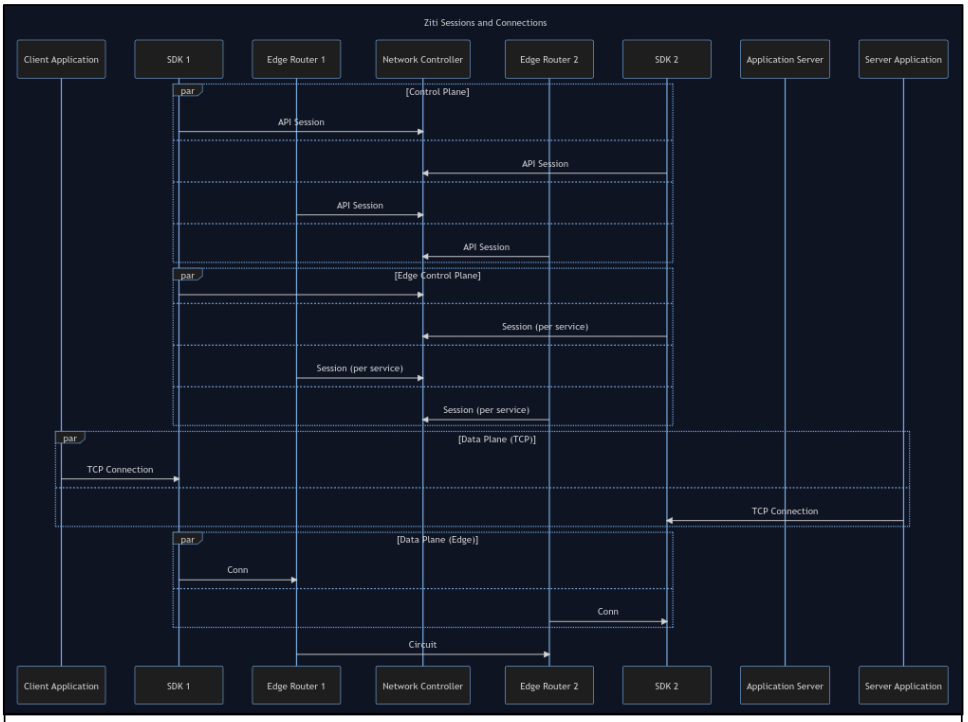
Η ταυτοποίηση των χρηστών/εφαρμογών σε ένα δίκτυο OpenZiti καθορίζεται συγκεκριμένα για κάθε ταυτότητα από την πολιτική ταυτοποίησης (authentication policy) με την οποία έχει αντιστοιχηθεί. Οι πολιτικές αυτές ορίζουν τους μηχανισμούς κύριας ταυτοποίησης (primary authentication), αλλά μπορούν να επιβάλλουν και τη χρήση επιπρόσθετων παραγόντων ταυτοποίησης (secondary authentication). Οι πολιτικές ταυτοποίησης αναπαρίστανται (όπως σχεδόν όλες οι λογικές οντότητες ενός δικτύου OpenZiti) ως JSON objects. Στην εικόνα 5.5 παρουσιάζεται παράδειγμα μιας πολιτικής ταυτοποίησης δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti.

5.4 Είδη Συνδέσεων σε δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti

Όπως έχει ήδη αναφερθεί και στις προηγούμενες υποενότητες της παρούσας Διπλωματικής Εργασίας υπάρχουν διαφορετικά είδη συνδέσεων και "συνεδριών" (sessions) που χρησιμοποιούνται σε ένα δίκτυο που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης μέσω της πλατφόρμας OpenZiti. Οι παραπάνω συνδέσεις ταξινομούνται ακροθιγώς σε δύο αφηρημένα "στρώματα" στα πλαίσια ενός δικτύου OpenZiti, το επίπεδο ελέγχου (Control Plane) και το επίπεδο δεδομένων (Data Plane).

5.4.1 Επίπεδο Ελέγχου δικτύων Μηδενικής Εμπιστοσύνης OpenZiti

Η δημιουργία συνεδρίας (session) API είναι το πρώτο χρονικά αλλά και ουσιαστικότερο συνολικά σημείο της διαδικασίας χρήσης ενός δικτύου OpenZiti. Μια συνεδρία API εν ισχύ είναι απαραίτητη για να αποτελεί ένας κόμβος τμήμα του "επικαλύπτοντος" δικτύου (Overlay Network) και δημιουργείται έπειτα από την αμφίδρομη ταυτοποίηση τόσο της εφαρμογής που χρησιμοποιεί το δίκτυο ως πελάτη όσο και του ίδιου του ελεγκτή (controller) του δικτύου.



Εικόνα 5.6: Αλληλουχία συνδέσεων και είδη συνεδριών (sessions) στα δίκτυα Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti

Στη συνέχεια δημιουργείται μια "συνεδρία" (session) η οποία αναφέρεται σε κάποια συγκεκριμένη υπηρεσία του δικτύου (OpenZiti service) για την οποία έχει κατάλληλα δικαιώματα (permissions) η εφαρμογή πελάτη και εξουσιοδοτείται μέσω της ήδη υπάρχουσας συνεδρίας (session) API που ταυτοποιεί την εφαρμογή πελάτη. Το JSON object που αναπαριστά τη "συνεδρία" (session) περιλαμβάνει πληροφορίες σχετικά με τις πολιτικές εξουσιοδότησης (authorization policies) που σχετίζονται με την υπηρεσία

(OpenZiti service) στην οποία επιδιώκει να αποκτήσει πρόσβαση η εφαρμογή πελάτη, το διακριτικό της εφαρμογής στα πλαίσια του δικτύου OpenZiti και άλλα.

Στη συνέχεια δημιουργούνται συνδέσεις ανάμεσα στην εφαρμογή πελάτη και κάθε δρομολογητή "άκρου" που (βάσει των πολιτικών) μπορεί να χρησιμοποιηθεί από τη συγκεκριμένη ταυτότητα για δρομολόγηση δικτυακής κίνησης για τη συγκεκριμένη υπηρεσία. Οι καθυστερήσεις (latency) των συνδέσεων αυτών αλλά και ολόκληρου του "μονοπατιού" (path) έως τον διακομιστή (server) μετρώνται συνεχώς ώστε να επιλέγεται πάντα το ταχύτερο. Όπως έχει ήδη αναφερθεί στην παρούσα Διπλωματική Εργασία, οι δρομολογητές ενός δικτύου OpenZiti δημιουργούν συνδέσεις μεταξύ τους διαμορφώνοντας ένα "πλέγμα" δικτύου (mesh Network), το οποίο έχει διττή υπόσταση, αφού αποτελεί τμήμα τόσο του επιπέδου ελέγχου, όσο και του επιπέδου δεδομένων που θα περιγραφεί στη συνέχεια.

5.4.2 Επίπεδο Δεδομένων δικτύων Μηδενικής Εμπιστοσύνης OpenZiti

Το επίπεδο δεδομένων ενός δικτύου που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης χρησιμοποιώντας την πλατφόρμα OpenZiti, περιλαμβάνει διαφορετικές συνδέσεις. Αν σε μια τοπολογία χρησιμοποιούνται εφαρμογές tunneling ή δρομολογητές σε λειτουργία tunnel είτε στην πλευρά της εφαρμογής πελάτη, είτε στην πλευρά της εφαρμογής που λειτουργεί ως διακομιστής, τότε πραγματοποιούνται και οι αντίστοιχες συνδέσεις TCP από τις εφαρμογές πελάτη προς την εφαρμογή tunnel ή τον δρομολογητή "άκρου" και αντίστροφα από τα αντίστοιχα στοιχεία τους δικτύου προς τις εφαρμογές που λειτουργούν ως διακομιστές.

Σε περίπτωση που είτε η εφαρμογή πελάτη είτε η εφαρμογή που λειτουργεί ως διακομιστής, ενσωματώνουν κώδικα βιβλιοθήκης της πλατφόρμας OpenZiti μέσω του αντίστοιχου SDK, τότε οι παραπάνω συνδέσεις περιττεύουν και δεν απαιτούνται για την επικοινωνία με ένα δίκτυο Μηδενικής Εμπιστοσύνης OpenZiti. Οι συνδέσεις ανάμεσα σε κάποιο τερματικό σημείο δικτύου (είτε από την πλευρά της εφαρμογής πελάτη, είτε από την πλευρά του διακομιστή) με τον δρομολογητή "άκρου" που αποτελεί σημείο εισόδου για την αντίστοιχη εφαρμογή προς το δίκτυο Μηδενικής Εμπιστοσύνης OpenZiti, ονομάζονται συνδέσεις "άκρου" (Edge Connections) και έχουν τη δυνατότητα να ενθυλακώνουν πολλαπλές συνδέσεις για διαφορετικές υπηρεσίες του εκάστοτε δικτύου.

Το επίπεδο δεδομένων ενός δικτύου Μηδενικής Εμπιστοσύνης OpenZiti αποτελείται και από τα "κυκλώματα" συνδέσεων ανάμεσα στους δρομολογητές. Το λογικό σημείο εκκίνησης των "κυκλωμάτων" τοποθετείται στο δρομολογητή που αποτελεί σημείο "εισόδου" στο δίκτυο Μηδενικής Εμπιστοσύνης για τον εκάστοτε "πελάτη άκρου" (Edge-Client) και φτάνουν έως και τον δρομολογητή που θα αναλάβει τη σύνδεση, είτε με το τερματικό σημείο του δικτύου, είτε και με την ίδια την εφαρμογή που λειτουργεί ως διακομιστής, στην περίπτωση που αυτή ενσωματώνει κώδικα βιβλιοθήκης της πλατφόρμας OpenZiti (μέσω του αντίστοιχου SDK).

Τέλος, ένα "κύκλωμα" (circuit) δρομολογητών μπορεί να αποτελείται από έναν ή περισσότερους δρομολογητές και αντίστοιχα από καμία ή περισσότερες συνδέσεις ανάμεσα τους. Στην εικόνα 5.6 απεικονίζονται σχηματικά οι διαφορετικές συνδέσεις που αναλύθηκαν στις δύο τελευταίες υποενότητες, ενώ ταξινομούνται και στα αντίστοιχα επίπεδα ελέγχου και δεδομένων όπως περιγράφηκαν προηγουμένως.

5.5 Πρόσβαση σε υπηρεσίες μέσω δικτύων Μηδενικής Εμπιστοσύνης OpenZiti

Η διαδικασία πρόσβασης σε μια υπηρεσία μέσω ενός δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti είναι μια πολύπλοκη αλλά καλά οργανωμένη διαδικασία που εξασφαλίζει την ασφάλεια και αποτελεσματικότητα της επικοινωνίας. Για το σκοπό αυτό τα βήματα, τα οποία περιλαμβάνει η παραπάνω διαδικασία πρέπει να εκτελούνται με συγκεκριμένη σειρά. Αρχικά, η εφαρμογή πελάτη πρέπει να ταυτοποιείται στον ελεγκτή (controller) του δικτύου OpenZiti. Αυτό το βήμα είναι κρίσιμο για την ασφάλεια του συστήματος, καθώς εξασφαλίζει ότι μόνο ταυτοποιημένες εφαρμογές μπορούν να αποκτήσουν πρόσβαση στο δίκτυο. Το αποτέλεσμα μιας επιτυχημένης ταυτοποίησης είναι η δημιουργία μιας συνεδρίας (session) API, η οποία λειτουργεί ως ένα είδος διαπιστευτηρίου για τις επόμενες αλληλεπιδράσεις με το δίκτυο.

Μετά την ταυτοποίηση, η εφαρμογή πελάτη πρέπει να δημιουργήσει μια συνεδρία (Session) για την επιθυμητή υπηρεσία. Κατά τη δημιουργία της συνεδρίας αυτής, ο ελεγκτής (controller) επαληθεύει ότι η ταυτότητα που χρησιμοποιεί η εφαρμογή πελάτη έχει δικαίωμα πρόσβασης στη συγκεκριμένη υπηρεσία. Επιπλέον, ο ελεγκτής προσδιορίζει ποιοι δρομολογητές εξ όσων έχουν ενεργοποιημένη τη σχετική λειτουργία “άκρου” (edge routers) μπορούν να χρησιμοποιηθούν για την πρόσβαση στην υπηρεσία από τη συγκεκριμένη ταυτότητα. Οι πληροφορίες για τους διαθέσιμους δρομολογητές “άκρου” επιστρέφονται από τον ελεγκτή προς την εφαρμογή πελάτη μαζί με το διακριτικό (token) της συνεδρίας.

Στη συνέχεια, η εφαρμογή πελάτη συνδέεται με έναν ή περισσότερους από τους διαθέσιμους δρομολογητές “άκρου”. Αξίζει να σημειωθεί ότι οι συνδέσεις με τους δρομολογητές “άκρου” δεν είναι μοναδικές ανά υπηρεσία, αλλά έχουν τη δυνατότητα να ενθυλακώνουν συνδέσεις για όλες τις υπηρεσίες στις οποίες έχει δικαίωμα πρόσβασης η εκάστοτε ταυτότητα (identity) που χρησιμοποιείται. Η σύνδεση πραγματοποιείται με αμφίδρομη εφαρμογή TLS (mTLS), ενώ απαιτείται επίσης από την εφαρμογή πελάτη αποστολή του token της σχετικής συνεδρίας API στους δρομολογητές “άκρου” για περαιτέρω επαλήθευση της εξουσιοδότησης της σύνδεσης και κατ’ επέκταση της πρόσβασης στην εκάστοτε υπηρεσία.

Αφού συνδεθεί με έναν δρομολογητή “άκρου”, η εφαρμογή πελάτη στέλνει ένα αίτημα κλήσης (Dial request) για την επιθυμητή υπηρεσία, μαζί με το διακριτικό της συνεδρίας (session token). Ο επιλεγμένος δρομολογητής “άκρου” (initiating router) μεταφράζει αυτό το αίτημα σε ένα αίτημα "δημιουργίας κυκλώματος" (circuit request) προς τον ελεγκτή (controller). Ο ελεγκτής, λαμβάνοντας το αίτημα δημιουργίας “κυκλώματος” (circuit request), επιλέγει τη βέλτιστη διαδρομή μέσω του δικτύου. Η επιλογή αυτή βασίζεται σε διάφορους παράγοντες, συμπεριλαμβανομένου του κόστους (συνολική καθυστέρηση) των διαθέσιμων διαδρομών για τη συγκεκριμένη υπηρεσία. Ο τελευταίος δρομολογητής στη διαδρομή ονομάζεται δρομολογητής τερματισμού.

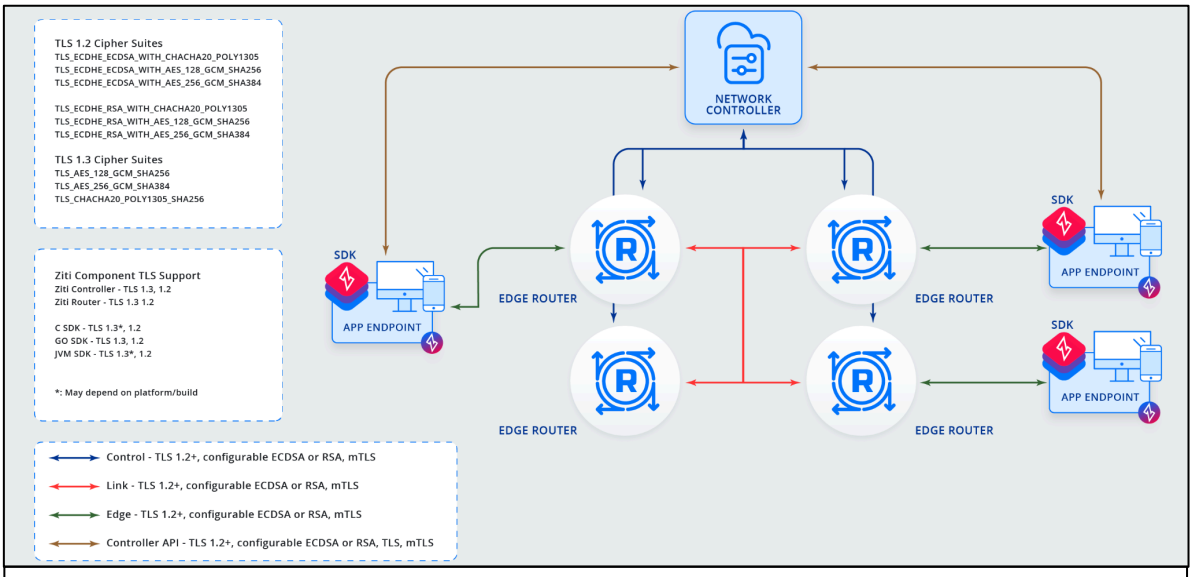
Μετά την επιλογή της διαδρομής, ο ελεγκτής ενημερώνει τους δρομολογητές στη διαδρομή, ώστε να μπορούν να επικαιροποιήσουν τους πίνακες δρομολόγησής τους ή/και να δημιουργήσουν TCP/UDP συνδέσεις με εφαρμογές που λειτουργούν ως διακομιστές, όπως απαιτείται στην περίπτωση τερματικού δρομολογητή που πρέπει να συνδεθεί με εφαρμογή που δεν ενσωματώνει δυνατότητα σύνδεσης με δρομολογητές OpenZiti. Μόλις δημιουργηθούν οι επιμέρους συνδέσεις μεταξύ των δρομολογητών (link connections) και η σύνδεση με την εφαρμογή που λειτουργεί ως διακομιστής, το κύκλωμα ολοκληρώνεται. Ο

ελεγκτής ειδοποιεί τον δρομολογητή εκκίνησης (initiating router) και πλέον τα δεδομένα μπορούν να μεταφέρονται μεταξύ της εφαρμογής πελάτη και της εφαρμογής που λειτουργεί ως διακομιστής.

Αυτή η διαδικασία εξασφαλίζει ότι κάθε σύνδεση σε μια υπηρεσία μέσω δικτύων Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti είναι ασφαλής, ταυτοποιημένη και βελτιστοποιημένη από άποψη επιδόσεων. Η χρήση της συνεδρίας API και των συνεδριών υπηρεσιών επιτρέπει τη συνεχή επαλήθευση της ταυτότητας και των δικαιωμάτων πρόσβασης, ενώ η δυναμική επιλογή διαδρομής (smart-routing) εξασφαλίζει την αποτελεσματική χρήση των πόρων του δικτύου. Επιπλέον, η χρήση αμφίδρομου (mutual) TLS και κρυπτογραφημένων συνδέσεων σε όλα τα επίπεδα, προσφέρει υψηλή ασφάλεια για όλες τις επικοινωνίες στα πλαίσια του δικτύου.

5.6 Ασφάλεια Συνδέσεων σε δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti

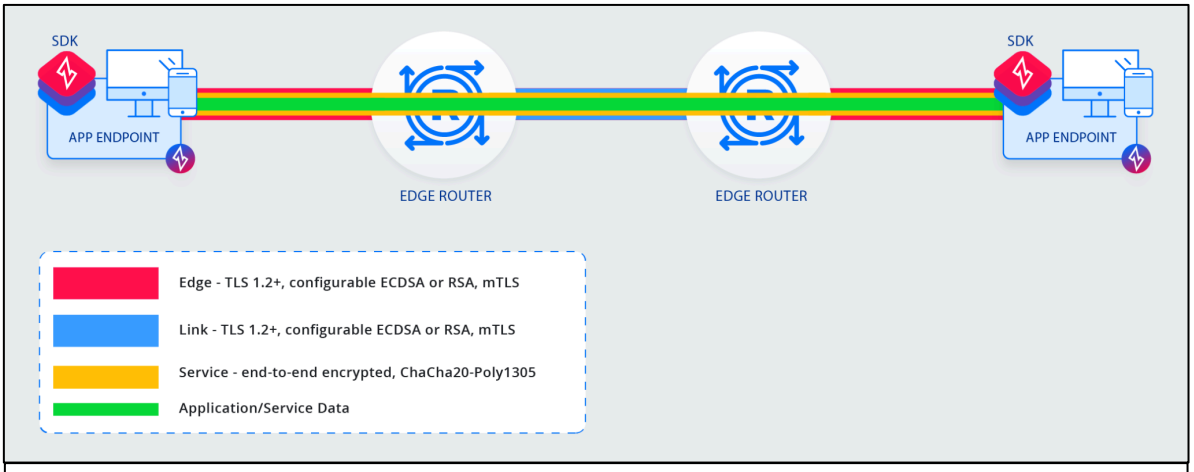
Σύγχρονοι μηχανισμοί ασφαλείας και κρυπτογραφικοί αλγόριθμοι αποτελούν ακρογωνιαίο λίθο της λειτουργίας των δικτύων Μηδενικής Εμπιστοσύνης OpenZiti. Οι μηχανισμοί ασφαλείας αυτοί έχουν επιλεχθεί για κάθε στοιχείο των δικτύων OpenZiti, ανάλογα με τον σκοπό του. Στα δίκτυα OpenZiti πραγματοποιείται ένα σύνολο από συνδέσεις διαφορετικών ειδών.



Εικόνα 5.7: Εκδόσεις πρωτοκόλλου TLS και κρυπτογραφικοί αλγόριθμοι ανά είδος σύνδεσης, στα δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti

Οι συνδέσεις μεταξύ ενός ελεγκτή δικτύου (network controller) και ενός δρομολογητή (router) για τη διαχείριση της λειτουργίας ενός δικτύου Μηδενικής Εμπιστοσύνης OpenZiti ονομάζονται συνδέσεις ελέγχου (control connections). Οι συνδέσεις μεταξύ των δρομολογητών (routers) διαμορφώνουν ουσιαστικά ένα "πλέγμα" δικτύου και ονομάζονται συνδέσεις "link". Οι συνδέσεις που υλοποιούν τελικά την επικοινωνία μεταξύ της εφαρμογής πελάτη (client) και του εκάστοτε διακομιστή (server) ονομάζονται συνδέσεις υπηρεσιών (service connections).

Οι συνδέσεις υπηρεσιών (service connections) κρυπτογραφούνται από άκρο σε άκρο χρησιμοποιώντας τις συναρτήσεις της βιβλιοθήκης κρυπτογραφίας δημόσιου κλειδιού libsodium. Οι συνδέσεις ανάμεσα στους "πελάτες άκρου" (Edge-Clients) των δικτύων Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti και τους δρομολογητές των δικτύων που έχουν ενεργοποιημένη τη λειτουργία "άκρου" ονομάζονται συνδέσεις "άκρου" (edge connections).



Εικόνα 5.8: Ενθυλάκωση συνδέσεων στα δίκτυα Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti

Οι συνδέσεις αυτές ουσιαστικά ενθυλακώνουν στα δεδομένα (payload) που μεταφέρονται συνδέσεις υπηρεσιών (service connections). Κάθε σύνδεση "άκρου" (edge connection) έχει τη δυνατότητα να ενθυλακώνει περισσότερες από μια συνδέσεις υπηρεσιών (service connections) στα δεδομένα (payload) που μεταφέρει, έχει δηλαδή δυνατότητα πολυπλεξίας των παραπάνω συνδέσεων. Οι συνδέσεις ανάμεσα στους "πελάτες άκρου" (Edge Clients) και τον ελεγκτή (controller) ενός δικτύου OpenZiti ονομάζονται συνδέσεις API ελεγκτή (controller API connections).

Οι συνδέσεις αυτές χρησιμοποιούν πρωτόκολλο HTTPS (HTTP over TLS) ή/και ασφαλή websockets. Σε όλες τις παραπάνω συνδέσεις χρησιμοποιείται αμφίδρομη υλοποίηση TLS έκδοσης 1.2 ή και 1.3, εκτός από περιπτώσεις συνδέσεων που αφορούν πολύ συγκεκριμένα μη κρίσιμα endpoints του API του ελεγκτή (controller) ενός δικτύου OpenZiti, χρήση των οποίων μπορεί να καταστεί απαραίτητη πριν την ταυτοποίηση ενός χρήστη/εφαρμογής/συσκευής (π.χ. ανάκτηση της έκδοσης του controller ώστε να συνταχθεί κατάλληλο authentication request). Στις περιπτώσεις αυτές εφόσον η εφαρμογή πελάτη δεν έχει ακόμα ταυτοποιηθεί και κατ' επέκταση δεν είναι δυνατή η χρήση αμφίδρομου TLS, χρησιμοποιείται το πρωτόκολλο TLS μη αμφίδρομα, για την ταυτοποίηση δηλαδή αποκλειστικά του ελεγκτή του δικτύου OpenZiti. Αξίζει να σημειωθεί, ότι στις περιπτώσεις που ο χρήστης/εφαρμογή/συσκευή δεν έχει στη διάθεση του "long-lived" ψηφιακό πιστοποιητικό X.509 και η ταυτοποίηση έχει πραγματοποιηθεί με διαφορετικό μηχανισμό (JWT ή συνδυασμό username/password), είναι δυνατή η έκδοση εφήμερων πιστοποιητικών X.509 (με χρονικό εύρος εγκυρότητας ίδιο με της αντίστοιχης συνεδρίας API στα πλαίσια της οποίας δημιουργήθηκαν) ώστε να μπορούν να πραγματοποιηθούν συνδέσεις mTLS με τον ελεγκτή ή τους δρομολογητές ενός δικτύου OpenZiti.

Στην εικόνα 5.7 απεικονίζεται σχηματικά η τοπολογία ενός δικτύου Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης OpenZiti και τα επιμέρους είδη συνδέσεων ανάμεσα στα διάφορα στοιχεία λογισμικού που το αποτελούν. Στην εικόνα 5.7 δεν απεικονίζονται οι συνδέσεις υπηρεσιών (service connections) που ενθυλακώνονται στα δεδομένα (payload) των συνδέσεων "άκρου" (edge connections). Για αυτό τον λόγο έχει παρατεθεί και η εικόνα 5.8, όπου απεικονίζεται η ενθυλάκωση των συνδέσεων υπηρεσιών (service connections) στο ωφέλιμο φορτίο των συνδέσεων "άκρου" (edge connections) και των συνδέσεων "link" αντίστοιχα καθώς και τα πρωτόκολλα ασφαλείας που χρησιμοποιούνται από τους διαφορετικούς τύπους συνδέσεων.

5.7 Εγγραφή (enrollment) ταυτοτήτων σε δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti

Η διαδικασία "εγγραφής" (enrollment) ταυτοτήτων αποτελεί θεμελιώδη λειτουργία των δικτύων Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti. Η εγγραφή ταυτοτήτων οδηγεί στη δημιουργία και σύνδεση διαπιστευτηρίων (credentials) ασφαλείας με μια προ-υπάρχουσα ταυτότητα ενός δικτύου OpenZiti. Κατά τη διάρκεια της εγγραφής, οι εφαρμογές ενημερώνονται για τη διεύθυνση και τη θύρα μέσω των οποίων μπορούν να επικοινωνούν με τον ελεγκτή του δικτύου, καθιστώντας απαραίτητη την επανεγγραφή ή την αντικατάσταση της ταυτότητας σε περίπτωση αλλαγής της κοινοποιημένης διεύθυνσης ή θύρας του ελεγκτή. Η διαδικασία εγγραφής είναι ζωτικής σημασίας για τη δυνατότητα πραγματοποίησης αμοιβαία ταυτοποιημένων συνδέσεων TLS, οι οποίες αποτελούν ακρογωνιαίο λίθο της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης.

Η πιο συνηθισμένη μέθοδος εγγραφής ταυτοτήτων σε δίκτυα Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti είναι με χρήση ενός διακριτικού μίας χρήσης (One Time Token - OTT). Με τη χρήση αυτής της μεθόδου, παράγεται ένα διακριτικό (token) μίας χρήσης κατά τη διαδικασία δημιουργίας της ταυτότητας. Αυτό το διακριτικό (token) υποβάλλεται αργότερα στον ελεγκτή του δικτύου OpenZiti ως μέρος της διαδικασίας "εγγραφής" της ταυτότητας. Μόλις μια ταυτότητα εγγραφεί επιτυχώς, το διακριτικό (token) μίας χρήσης δεν είναι πλέον έγκυρο και δεν μπορεί να χρησιμοποιηθεί για την εγγραφή της ίδιας ταυτότητας ξανά.

Τα διακριτικά (tokens) μίας χρήσης δημιουργούνται από τον ελεγκτή ως JWT (JSON Web Token) και λήγουν 24 ώρες μετά τη δημιουργία της ταυτότητας. Το JWT μπορεί να ληφθεί μέσω του γραφικού περιβάλλοντος διαχείρισης δικτύων OpenZiti ή μέσω της αντίστοιχης διεπαφής γραμμής εντολών. Η διαδικασία εγγραφής με χρήση OTT περιλαμβάνει τη δημιουργία ενός ιδιωτικού κλειδιού και ενός αιτήματος υπογραφής ψηφιακού πιστοποιητικού (Certificate Signing Request - CSR), το οποίο υποβάλλεται στον ελεγκτή του δικτύου.

Η ασφάλεια της διαδικασίας εγγραφής ταυτοτήτων είναι κρίσιμη για τη διατήρηση της ακεραιότητας ενός δικτύου Μηδενικής Εμπιστοσύνης OpenZiti. Αξίζει να σημειωθεί ότι έχουν προταθεί επιπλέον μηχανισμοί για την ασφαλή παράδοση διακριτικών (tokens) εγγραφής, συμπεριλαμβανομένης της χρήσης τεχνολογίας blockchain και Non-Fungible Tokens (NFTs) [42]. Αυτές οι προσεγγίσεις στοχεύουν στην ενίσχυση της ασφάλειας της διαδικασίας διανομής OTT.

Συμπερασματικά, η εγγραφή ταυτότητας στα δίκτυα μηδενικής εμπιστοσύνης OpenZiti αποτελεί μια πολύπλευρη και κρίσιμη διαδικασία. Παρέχει το θεμέλιο για την αμοιβαία επαλήθευση ταυτότητας και την ασφαλή επικοινωνία εντός του δικτύου. Καθώς η τεχνολογία εξελίσσεται, αναμένεται ότι θα αναπτυχθούν νέες και πιο προηγμένες μέθοδοι εγγραφής ταυτοτήτων, ενισχύοντας περαιτέρω την ασφάλεια και την αποτελεσματικότητα των δικτύων Μηδενικής Εμπιστοσύνης.

6. Υλοποίηση στα πλαίσια της παρούσας Διπλωματικής Εργασίας

6.1 Στόχοι Υλοποίησης της παρούσας Διπλωματικής Εργασίας

Κεντρικό ζητούμενο της παρούσας Διπλωματικής Εργασίας ήταν χρήση του project ανοικτού κώδικα cURL (“client URL”) [43] με σκοπό την επίτευξη επικοινωνίας (connectivity) με instances της εφαρμογής διαχείρισης περιεχομένου βάσεων δεδομένων FIWARE Orion Context Broker [44], μέσω δικτύων που υιοθετούν Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης χρησιμοποιώντας την πλατφόρμα ανοικτού κώδικα OpenZiti [45]. Προκειμένου να παρουσιαστεί η παραπάνω υλοποίηση είναι σκόπιμο να εξεταστούν τα παρακάτω:

1. Το οικοσύστημα των ποικίλων στοιχείων λογισμικού ανοικτού κώδικα της πλατφόρμας FIWARE, η οποία έχει αναπτύξει και την εφαρμογή Orion Context Broker, που χρησιμοποιείται ως διακομιστής (server) στην υλοποίηση της παρούσας διπλωματικής εργασίας [46]. Με τη μελέτη του οικοσυστήματος εργαλείων λογισμικού της πλατφόρμας FIWARE συνολικά, μπορούμε να αποκτήσουμε μια σφαιρική θεώρηση σχετικά με την ουσία και τη σημασία του θέματος της παρούσας Διπλωματικής Εργασίας, και της υλοποίησης που θα παρουσιαστεί, μέσα στο ευρύτερο πλαίσιο της έρευνας και του επιστημονικού πεδίου της ασφάλειας δικτύων και των Εφαρμογών Διαδικτύου, αλλά και για προεκτάσεις που είναι δυνατόν να προκύψουν από άλλες παρόμοιες εφαρμογές που μπορούν να εχουν υλοποιήσεις που ομοιάζουν με αυτή που παρουσιάζεται στην παρούσα Διπλωματική Εργασία.

2. Η εφαρμογή Orion Context Broker και ο τρόπος με τον οποίο δομείται η φιλοσοφία βάσει της οποίας αναπαρίστανται από την εφαρμογή οι λογικές οντότητες που διαχειρίζεται (entities, attributes κλπ.).

3. Το project cURL και η βασική του βιβλιοθήκη libcurl() [47], που αποτελεί ακρογωνιαίο λίθο του project cURL συνολικά και δομικό στοιχείο στην υλοποίηση που πραγματοποιήθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας.

4. Η χρήση του project cURL για την αποστολή αιτημάτων HTTP στην εφαρμογή FIWARE Orion Context Broker.

5. Συνοπτική παρουσίαση της λειτουργίας της εφαρμογής γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας.

Σύμφωνα με τους παραπάνω κεντρικούς άξονες αναπτύσσεται η δομή του παρόντος κεφαλαίου σε υποενότητες.

6.2 Εφαρμογές πελάτη και διακομιστή που χρησιμοποιήθηκαν στην παρούσα Εργασία

6.2.1 Η πλατφόρμα εργαλείων λογισμικού ανοικτού κώδικα FIWARE

Η FIWARE αποτελεί μια ανοιχτή πλατφόρμα η οποία προωθεί την ανάπτυξη καινοτόμων έξυπνων εφαρμογών ανοιχτού κώδικα (open-source smart applications) με έμφαση στον τομέα των "Έξυπνων Πόλεων" (Smart Cities), του Διαδικτύου των Πραγμάτων (IoT) και άλλων πεδίων που αξιοποιούν διαδικτυακές υπηρεσίες νέας γενιάς. Εμφανίστηκε αρχικά ως πρωτοβουλία της Ευρωπαϊκής Ένωσης με στόχο την επιτάχυνση της υιοθέτησης ανοιχτών προτύπων και την ενίσχυση ενός οικοσυστήματος ελεύθερα διαθέσιμων στοιχείων λογισμικού, προσφέροντας έτσι μια κοινή βάση για την ανάπτυξη καινοτόμων υπηρεσιών. Η φιλοσοφία της πλατφόρμας εστιάζει στη διευκόλυνση των μηχανικών λογισμικού, οργανισμών και πόλεων, επιτρέποντάς τους να αξιοποιούν επαναχρησιμοποιήσιμα δομικά στοιχεία, πρότυπα δεδομένων και διεπαφές προγραμματισμού εφαρμογών (APIs) για την υλοποίηση προηγμένων λύσεων.

Η αρχιτεκτονική της FIWARE βασίζεται σε μια σειρά από Ενεργά Δομικά Συστατικά (Generic Enablers), τα οποία είναι ανοιχτού κώδικα (open-source) και παίζουν τον ρόλο "δομικών στοιχείων" πάνω στα οποία μπορούν να στηριχθούν πολύπλοκες εφαρμογές. Μέσω της χρήσης των "δομικών στοιχείων" αυτών, επιτυγχάνεται διαλειτουργικότητα, επεκτασιμότητα και ευκολία ενσωμάτωσης σε ποικιλία από περιβάλλοντα, είτε αυτά αφορούν υλοποιήσεις σχετικά με την αστική κινητικότητα, τη διαχείριση αποβλήτων, τους ελέγχους περιβαλλοντικών δεικτών, την παρακολούθηση ενεργειακής κατανάλωσης ή άλλες "έξυπνες" λειτουργίες. Η κεντρική συνιστώσα της πλατφόρμας εργαλείων λογισμικού ανοικτού κώδικα FIWARE (Orion Context Broker), που χρησιμοποιείται ως διακομιστής (server) και στην παρούσα διπλωματική εργασία, έχει σχεδιαστεί να παρέχει μια συνολική "άποψη" των δεδομένων (context) μιας πόλης ή οποιασδήποτε άλλης ανάλογης υποδομής, επιτρέποντας σε εφαρμογές και υπηρεσίες να ανταλλάσσουν δεδομένα για αισθητήρες, συσκευές, χρήστες, και συμβάντα σε πραγματικό χρόνο.

Η υποστήριξη προτύπων όπως το Next Generation Service Interface Linked Data (NGSI-LD, NGSIv2 κλπ.) ένα μοντέλο δεδομένων και διεπαφής προγραμματισμού εφαρμογών (API) για την αναπαράσταση και διαχείριση πληροφοριών σε πραγματικό χρόνο, δίνει στην πλατφόρμα FIWARE τη δυνατότητα να προσφέρει υψηλό επίπεδο διαλειτουργικότητας μεταξύ ετερογενών πηγών δεδομένων (data sources) και συστημάτων. Η χρήση τέτοιων προτύπων καθιστά εφικτή την ανάπτυξη λύσεων που δεν εξαρτώνται από συγκεκριμένους προμηθευτές ή πλατφόρμες κλειστού κώδικα, ενώ ταυτόχρονα προάγει την ωρίμανση ενός βιώσιμου οικοσυστήματος καινοτομίας.

Ένα ακόμα σημαντικό πλεονέκτημα της πλατφόρμας εργαλείων λογισμικού ανοικτού κώδικα FIWARE είναι η ευκολία με την οποία προσαρμόζεται σε διαφορετικούς τομείς, πέραν των έξυπνων πόλεων: από τη γεωργία ακριβείας, στη Βιομηχανία 4.0 μέχρι και τις υπηρεσίες υγείας, η πλατφόρμα επιτρέπει την ανταλλαγή δεδομένων με τρόπο ασφαλή και επεκτάσιμο. Η ευελιξία αυτή οφείλεται στην υποστήριξη ανοιχτών διεπαφών προγραμματισμού εφαρμογών (APIs), όπως περιγράφηκε και προηγουμένως, αλλά και στη δυνατότητα ενσωμάτωσης πρόσθετων υπηρεσιών ανάλυσης δεδομένων, διαχείρισης ταυτοτήτων, ασφάλειας και ιδιωτικότητας.

Η πλατφόρμα FIWARE έχει αναδειχθεί σε μια από τις πιο ευρέως αποδεκτές πλατφόρμες για έξυπνες λύσεις, συμβάλλοντας στη δημιουργία προτύπων δεδομένων (data models) που μπορούν να υιοθετηθούν σε ολόκληρη την Ευρώπη και διεθνώς. Η βασισμένη σε ανοικτά πρότυπα προσέγγισή της υποστηρίζει πλήρως την ιδέα ενός αποκεντρωμένου, συνεργατικού οικοσυστήματος, όπου πολλαπλοί φορείς — δήμοι, εταιρείες τεχνολογίας, ακαδημαϊκά, ερευνητικά ιδρύματα — συμμετέχουν από κοινού στη δημιουργία καινοτόμων υπηρεσιών. Η παγκόσμια κοινότητα της πλατφόρμας FIWARE, με ενεργή συμμετοχή φορέων από όλο τον κόσμο, συνεχίζει να εμπλουτίζει τις λειτουργικότητες της πλατφόρμας, να βελτιώνει τη διαλειτουργικότητα και να διευρύνει το πεδίο εφαρμογών.

Συνοψίζοντας, η FIWARE αποτελεί μια ανοιχτού κώδικα πλατφόρμα που έχει τύχει ευρείας αποδοχής από ποικίλους δρώντες στο πεδίο της διαχείρισης δεδομένων σε πραγματικό χρόνο. Η προσέγγισή της, βασισμένη σε ανοικτά πρότυπα, επαναχρησιμοποιήσιμα εργαλεία λογισμικού, αλλά και η ευέλικτη αρχιτεκτονική της, την καθιστούν ένα σημαντικό σημείο αναφοράς για την ανάπτυξη έξυπνων εφαρμογών και υπηρεσιών, επηρεάζοντας σημαντικά τον τρόπο με τον οποίο οργανισμοί και πόλεις διαχειρίζονται, αξιοποιούν και ανταλλάσσουν δεδομένα.

6.2.2 FIWARE Orion Context Broker

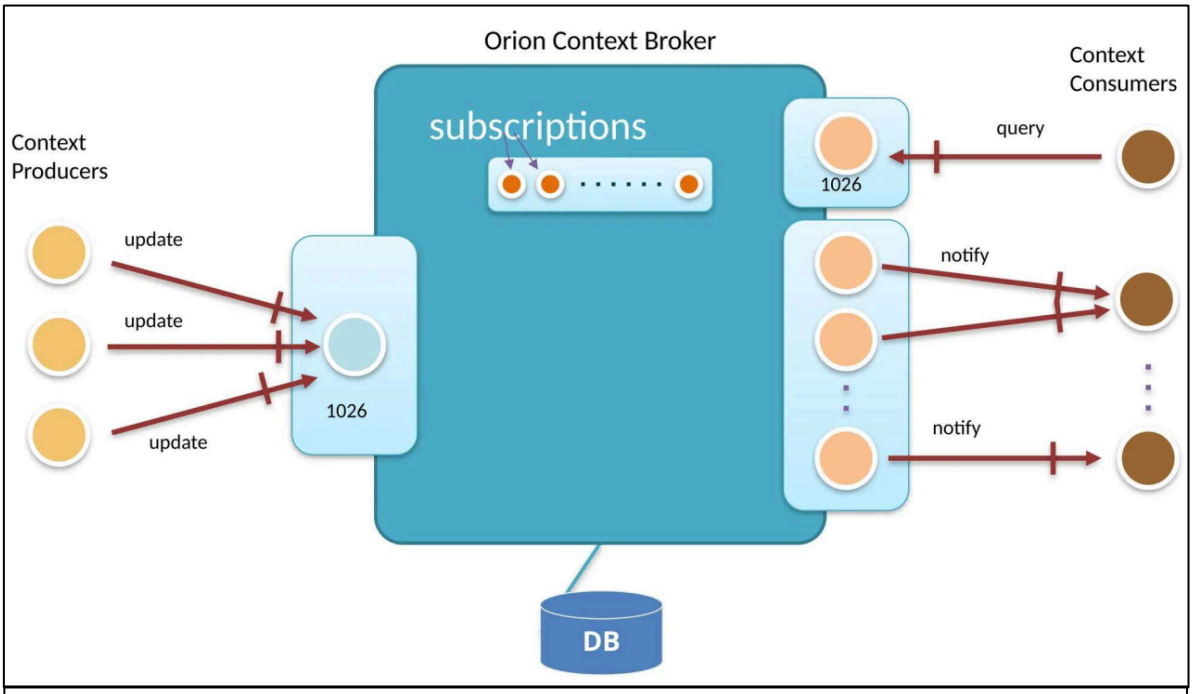
Ο FIWARE Orion Context Broker αποτελεί ένα θεμελιώδες στοιχείο (Generic Enabler) του οικοσυστήματος της πλατφόρμας εργαλείων λογισμικού ανοικτού κώδικα FIWARE. Έχει σχεδιαστεί με τέτοιο τρόπο ώστε να επιτρέπει τη διαχείριση, αποθήκευση και ανάκτηση δεδομένων (context information) σε πραγματικό χρόνο. Λειτουργώντας ως κεντρικό εργαλείο διαχείρισης δεδομένων (context broker), ο Orion επιτρέπει σε εφαρμογές, συσκευές, αισθητήρες και υπηρεσίες να μοιράζονται και να ενημερώνουν δεδομένα σχετικά με την κατάσταση του περιβάλλοντος στο οποίο δρουν, αποτελώντας εν τέλει κομμάτι μιας δυναμικής, βασισμένης σε συμβάντα (event-driven), προσαρμογής της λειτουργίας τους. Η χρήση του Orion Context Broker ως κεντρικό σημείο πρόσβασης σε δεδομένα συμβάλλει ιδιαίτερα σε εφαρμογές Έξυπνων Πόλεων (Smart Cities), Βιομηχανίας 4.0, δικτύων Ενέργειας (Energy Grids), και Διαδικτύου των Πραγμάτων (IoT), όπου η διαχείριση μεγάλου όγκου ετερογενών δεδομένων καθίσταται κρίσιμη.

Οι λειτουργίες του Orion Context Broker βασίζονται στο πρότυπο NGSI, αρχικά NGSI v2 και πλέον NGSI-LD, που προσφέρουν έναν ομοιόμορφο τρόπο αναπαράστασης, δημοσίευσης και κατανάλωσης δεδομένων. Μέσω των RESTful APIs, οι προγραμματιστές μπορούν εύκολα να δημιουργούν, να ενημερώνουν, να αναζητούν ή να διαγράφουν δεδομένα (entities, attributes, subscriptions), υποστηρίζοντας λειτουργίες όπως ενημέρωση δεδομένων σε πραγματικό χρόνο (real-time data updates), ερωτήματα (queries) και ειδοποιήσεις (notifications) προς συνδρομητές. Η κεντρική φιλοσοφία της αρχιτεκτονικής του Orion είναι η παροχή ενός ενιαίου σημείου αλήθειας (single source of truth) για δεδομένα τα οποία μεταβάλλονται συνεχώς, αξιοποιώντας την επικαιροποίηση της πληροφορίας και την ενεργό συμμετοχή πολλαπλών στοιχείων ενός οικοσυστήματος.

Σε επίπεδο επίδοσης, έχει καταστεί εμφανής η ικανότητα του Orion Context Broker να χειρίζεται μεγάλο όγκο δεδομένων, υψηλό ρυθμό αιτημάτων (requests) και συχνές ενημερώσεις δεδομένων, διατηρώντας χαμηλή καθυστέρηση (latency) και υψηλή διαθεσιμότητα (high resource-availability). Παράλληλα, έχει αναλυθεί η δυνατότητά του να προσφέρει επεκτασιμότητα (scalability) και ανθεκτικότητα (resilience) σε περιβάλλοντα παραγωγής, καθώς και να λειτουργεί αποτελεσματικά σε συνδυασμό με άλλα δομικά

στοιχεία της πλατφόρμας εργαλείων λογισμικού ανοικτού κώδικα FIWARE (π.χ. IoT Agents, Data/Context Processing Enablers). Αυτές οι ιδιότητες έχουν καταστήσει τον Orion Context Broker κομβικό εργαλείο για τη δημιουργία ενοποιημένων, δυναμικών υποδομών δεδομένων, ιδίως σε περιπτώσεις όπου απαιτείται χρήση δεδομένων από πολλαπλές πηγές (π.χ. αισθητήρες, υπηρεσίες cloud, εφαρμογές analytics).

Η μετάβαση στο NGSI-LD, μια πιο εκφραστική και διασυνδεδεμένη προσέγγιση αναπαράστασης δεδομένων βασισμένη στις αρχές του Linked Data, διευρύνει τις δυνατότητες του Orion Context Broker. Πλέον, οι οντότητες (entities) μπορούν να μοντελοποιηθούν με γνώμονα τις σχέσεις τους (relationships) και τα συμφραζόμενα δεδομένα (context data) μετατρέπονται σε γράφους δεδομένων, επιτρέποντας πιο λεπτομερή ερωτήματα (queries) και αποκρίσεις (responses). Αυτό το βήμα διευκολύνει την υλοποίηση έξυπνων εφαρμογών που απαιτούν αποθήκευση και επεξεργασία των δεδομένων, βελτιώνοντας την ικανότητα ανάκτησης, συσχέτισης περιεχομένου ετερογενών δεδομένων.

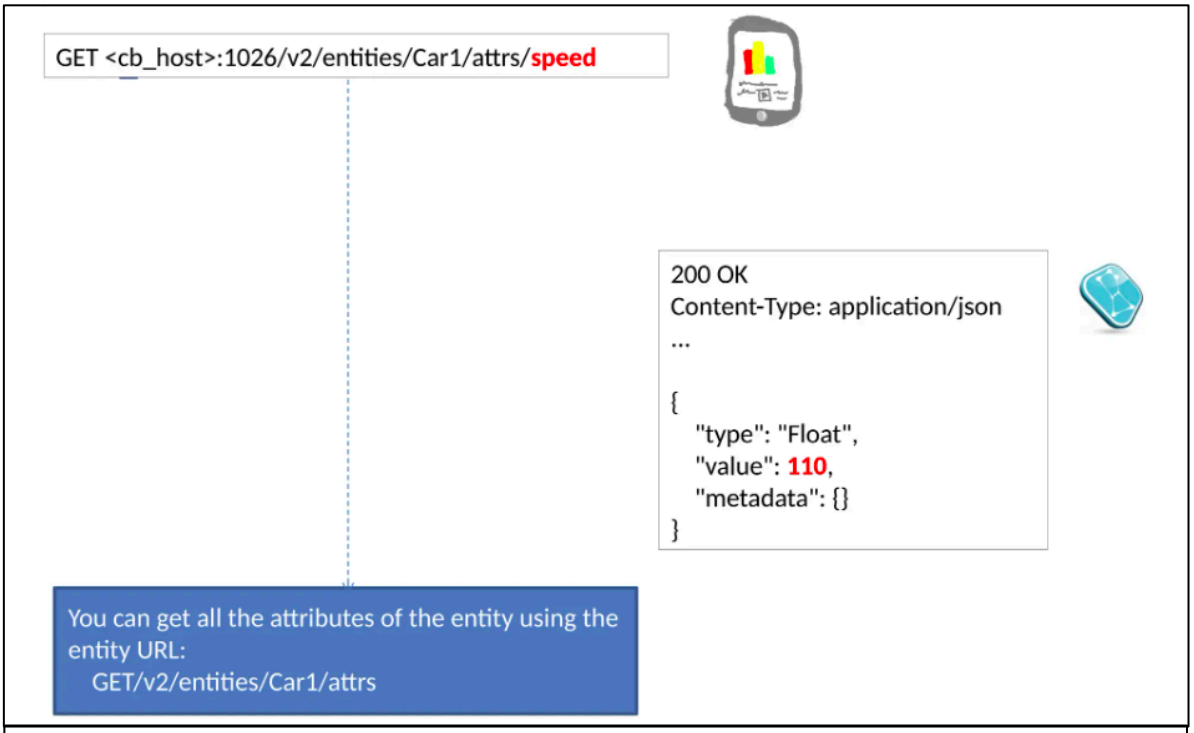


Εικόνα 6.1: Παράδειγμα τοπολογίας που περιλαμβάνει την εφαρμογή διαχείρισης περιεχομένου βάσεων δεδομένων FIWARE Orion Context Broker

Πέραν της τεχνικής πλευράς, αξιοσημείωτος είναι ο ρόλος του Orion Context Broker στην προώθηση ανοικτών προτύπων και οικοσυστημάτων εργαλείων λογισμικού ανοιχτού κώδικα, με γνώμονα τη διαλειτουργικότητα και την αποφυγή εξάρτησης από συγκεκριμένους παρόχους βάσεων δεδομένων (vendor lock-in). Η υποστήριξη διεθνών προτύπων και η παροχή συνεχώς εξελισσόμενων διεπαφών προγραμματισμού εφαρμογών (APIs) διευκολύνει την υιοθέτηση της τεχνολογίας διαχείρισης δεδομένων σε πραγματικό χρόνο από διάφορους κλάδους και οργανισμούς, επιταχύνοντας τη δημιουργία μιας "έξυπνης" οικονομίας δεδομένων και υπηρεσιών.

Αναφορικά με τις τοπολογίες που περιλαμβάνουν την εφαρμογή διαχείρισης βάσεων δεδομένων FIWARE Orion Context Broker σημαντικό ρόλο διαδραματίζουν οι "context producers", οι εφαρμογές/χρήστες δηλαδή που έχουν τη δυνατότητα/δικαιοδοσία να

δημιουργούν, να ενημερώνουν και να διαγράφουν περιεχόμενο των εκάστοτε βάσεων δεδομένων με τις οποίες επικοινωνούν τα instances της εφαρμογής Orion Context Broker, χρησιμοποιώντας τις POST, PUT, PATCH και DELETE, HTTP Methods αντίστοιχα. Παράλληλα, σε μια τοπολογία που περιλαμβάνει την εφαρμογή Orion Context Broker είναι δυνατό να περιλαμβάνονται χρήστες/εφαρμογές (content consumers) που έχουν τη δυνατότητα αποκλειστικά να ανακτούν το περιεχόμενο (το σύνολο ή μέρος αυτού) της βάσης δεδομένων με την οποία επικοινωνεί η εφαρμογή, χωρίς δυνατότητα δημιουργίας, ενημέρωσης/επεξεργασίας ή και διαγραφής περιεχομένου από τη βάση δεδομένων, ενώ τα παραπάνω σενάρια μπορούν και να συνδυάζονται μεταξύ τους.



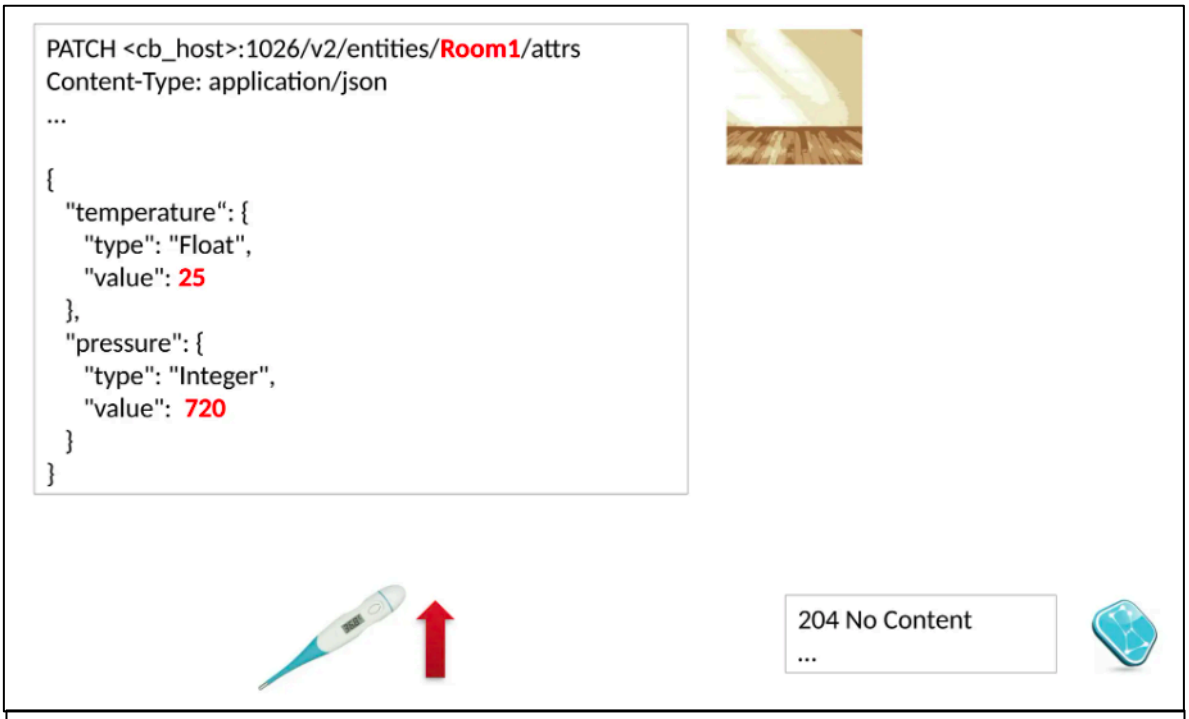
Εικόνα 6.2: Παράδειγμα χρήσης της εφαρμογής FIWARE Orion Context Broker για την ανάκτηση attribute οντότητας(entity)

Τέλος, μια τοπολογία που περιλαμβάνει την εφαρμογή Orion Context Broker είναι δυνατόν να επιτρέπει σε χρήστες/εφαρμογές, ανεξαρτήτως των δικαιωμάτων τους, να διενεργούν πράξεις (operations) με αιτήματα HTTP, όπως περιγράφηκαν παραπάνω, και να λαμβάνουν ειδοποιήσεις για συγκεκριμένες αλλαγές στο περιεχόμενο της βάσης δεδομένων με την οποία επικοινωνεί η εφαρμογή διαχείρισης περιεχομένου Orion Context Broker. Παράδειγμα τοπολογίας που περιλαμβάνει την εφαρμογή διαχείρισης περιεχομένου βάσεων δεδομένων Orion Context Broker απεικονίζεται στην εικόνα 6.1.

Στις εικόνες 6.2-6.4 απεικονίζονται διαφορετικές περιπτώσεις αιτημάτων HTTP προς την εφαρμογή διαχείρισης περιεχομένου βάσεων δεδομένων Orion Context Broker, αλλά και οι αντίστοιχες αποκρίσεις (responses) από την εφαρμογή. Πιο συγκεκριμένα, απεικονίζονται σχηματικά περιπτώσεις αιτημάτων HTTP χρησιμοποιώντας διαφορετικές HTTP Methods (GET, POST, PATCH), για τη δημιουργία, ανάκτηση και ενημέρωση περιεχομένου στη βάση δεδομένων με την οποία επικοινωνεί η εφαρμογή Orion Context Broker, ενώ τα παραδείγματα ενσωματώνονται σε ένα πλαίσιο πραγματικών σεναρίων

χρήσης, ώστε να γίνονται αντιληπτό το πιθανό πλαίσιο χρήσης της εφαρμογής σε σύνθετες υλοποιήσεις δικτύων IoT και εταιρικών/βιομηχανικών διατάξεων ευρύτερα.

Η εφαρμογή Orion Context Broker προσφέρει πληθώρα από επιλογές αναφορικά με τον τρόπο με τον οποίο τίθεται σε λειτουργία (deployment). Αρχικά, προσφέρει μια σειρά από επιλογές για ανάπτυξη ως docker container, δημιουργώντας έτσι ένα περιβάλλον ανεξάρτητο από εξωτερικές εξαρτήσεις αναφορικά με το λογισμικό (system dependencies), δίνοντας κατ' αυτόν τον τρόπο δυνατότητα χρήσης της εφαρμογής σε ποικιλία από διαφορετικά περιβάλλοντα και λειτουργικά συστήματα χωρίς να απαιτούνται ιδιαίτερες προσαρμογές. Επιπλέον, εκτελέσιμο αρχείο της εφαρμογής μπορεί να δημιουργηθεί και μεταγλωττίζοντας τον πηγαίο κώδικα (source-code) της εφαρμογής, μια διαδικασία ωστόσο που απαιτεί κατάλληλο απαραίτητο λογισμικό (software dependencies) στη διάθεση του μεταγλωττιστή, ώστε η διαδικασία να μπορεί να ολοκληρωθεί με επιτυχία.



Εικόνα 6.3: Παράδειγμα χρήσης της εφαρμογής FIWARE Orion Context Broker για την ενημέρωση οντότητας(entity)

Στα πλαίσια της παρούσας Διπλωματικής Εργασίας έγινε χρήση της εφαρμογής διαχείρισης περιεχομένου βάσεων δεδομένων FIWARE Orion Context Broker ως docker-compose container. Παράλληλα σε άλλο docker-compose container του ίδιου δικτύου containers τέθηκε σε λειτουργία βάση δεδομένων MongoDB. Η ανάπτυξη του παραπάνω δικτύου containers πραγματοποιείται με χρήση κατάλληλου αρχείου configuration (επέκτασης ".yml"). Στο εν λόγω αρχείο καθορίζονται η έτοιμη ανεπτυγμένη έκδοση (image) της κάθε εφαρμογής που θα ανακτηθεί, οι θύρες τις οποίες θα χρησιμοποιεί η εφαρμογή FIWARE Orion Context Broker για να δέχεται αιτήματα (requests) HTTP για τις περιπτώσεις που αυτά προέρχονται εσωτερικά ή και εξωτερικά του docker-compose container, αλλά και διαφορετικά options/flags που καθορίζουν τη λειτουργία των δύο εφαρμογών.

Παραδείγματος χάριν, για την εφαρμογή Orion Context Broker πρέπει να ορίζεται το Unique Resource Identifier το οποίο θα χρησιμοποιεί για να επικοινωνεί με instance βάσης δεδομένων MongoDB. Ένα τέτοιο Unique Resource Identifier πρέπει να ορίζει το πρωτόκολλο και τη διεύθυνση IP ή το idn hostname τα οποία θα χρησιμοποιεί η εφαρμογή διαχείρισης περιεχομένου Orion Context Broker για να επικοινωνεί με τη βάση δεδομένων MongoDB. Στην εικόνα 6.5 φαίνεται το περιεχόμενο του αρχείου επέκτασης ".yml" που χρησιμοποιήθηκε στα πλαίσια της παρούσας εργασίας, για την ανάπτυξη της εφαρμογής FIWARE Orion Context Broker και βάσης δεδομένων MongoDB με την οποία επικοινωνεί, ως δίκτυο containers docker-compose.

Εικόνα 6.4: Παράδειγμα χρήσης της εφαρμογής Orion Context Broker για τη δημιουργία οντότητας(entity)

Εν κατακλείδι, η εφαρμογή διαχείρισης περιεχομένου FIWARE Orion Context Broker αποτελεί ένα πολύτιμο εργαλείο για την ανάπτυξη καινοτόμων λύσεων που αξιοποιούν δεδομένα σε πραγματικό χρόνο. Η υποστήριξη προτύπων όπως το NGSI-LD ενισχύει τη διαλειτουργικότητα, καθιστώντας τον Orion Context Broker ιδανικό για την ανάπτυξη καινοτόμων λύσεων σε τομείς όπως οι έξυπνες πόλεις, το Διαδίκτυο των Πραγμάτων και η Βιομηχανία 4.0. Η ευελιξία, η επεκτασιμότητα και η ανοιχτή φύση του συμβάλλουν στη δημιουργία ενός δυναμικού οικοσυστήματος καινοτομίας, επιτρέποντας στους οργανισμούς να αναπτύξουν προηγμένες εφαρμογές και υπηρεσίες που ανταποκρίνονται στις σύγχρονες προκλήσεις της ψηφιακής εποχής.

```
version: "3"

services:
  orion:
    image: telefonicaiot/fiware-orion
    ports:
      - "1026:1026"
    depends_on:
      - mongo
    command: -dbURI mongodb://mongo

  mongo:
    image: mongo:6.0
    command: --nojournal
```

Εικόνα 6.5: Παράδειγμα αρχείου "docker-compose.yml" για την εκτέλεση της εφαρμογής Orion Context Broker ως docker container

6.2.3 cURL project

Το πλαίσιο και η ευρεία χρήση του project cURL – εργαλείο γραμμής εντολών και βιβλιοθήκη για τη μεταφορά δεδομένων με χρήση ποικίλων πρωτοκόλλων που χειρίζονται URLs – έχει αποτελέσει εδώ και δεκαετίες πολύτιμο εργαλείο στη διάθεση μηχανικών λογισμικού, μηχανικών και διαχειριστών δικτύων και δικτυακών υπηρεσιών που ασχολούνται με την ασφάλεια δικτύων και την ανάπτυξη αυτόματων διεργασιών για εφαρμογές και υπηρεσίες. Η πολυχρηστικότητα, η σταθερότητα και η διευρυμένη υποστήριξη πρωτοκόλλων όπως HTTP, HTTPS, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), και πολλών άλλων, που προσφέρει το project cURL το καθιστούν ένα θεμελιώδες εργαλείο στην υποδομή ανάπτυξης, δοκιμών, ενορχήστρωσης και αυτοματοποίησης υπηρεσιών διαδικτύου.

Το project cURL αποτελεί σταθερά αναπόσπαστο τμήμα διαδικασιών αξιολόγησης της επίδοσης συστημάτων, καθώς επιτρέπει την αυτοματοποιημένη λήψη δεδομένων από δικτυακές πηγές (network sources), που μπορούν αξιοποιηθούν για περαιτέρω ανάλυση. Η παραπάνω δυνατότητα χρησιμοποιείται συχνά για συγκριτικές μελέτες διαδικτυακών υπηρεσιών, πειραματικές προσομοιώσεις φόρτου (stress test under peak load), ανθεκτικότητας υποδομών, και ελέγχους συμβατότητας υπηρεσιών σε ένα μεγάλο εύρος από διαφορετικά περιβάλλοντα. Σε ορισμένες περιπτώσεις, η χρήση του project cURL σε συνδυασμό με εργαλεία διαχείρισης συστημάτων (system administration tools) επιτρέπει τη δημιουργία αρχιτεκτονικών συνεχούς ενσωμάτωσης (continuous integration architecture) και συνεχούς παράδοσης (continuous delivery architecture), συμβάλλοντας έτσι στην ομαλή ενσωμάτωση και ανάπτυξη κώδικα σε κλίμακα (at-scale), πριν από τη διάθεση των αντίστοιχων εφαρμογών σε περιβάλλοντα "παραγωγής" (production environment), όπου είναι πολύ πιο κρίσιμη η διασφάλιση υψηλών επιπέδων αξιοπιστίας στις υπηρεσίες που παρέχουν οι εν λόγω εφαρμογές.

Επιπλέον, σε μελέτες ασφάλειας δικτύων, το project cURL αξιοποιείται για αυτοματοποιημένους ελέγχους ευπαθειών (automated vulnerability checks), ελέγχους

διείσδυσης (penetration testing) και ανάκτηση δεδομένων προς ανάλυση προτύπων επιθέσεων (attack-categorization), ανεξάρτητα από την πολυπλοκότητα του υποκείμενου δικτυακού πρωτοκόλλου. Αυτή η ικανότητα επιτρέπει την ενσωμάτωση του project cURL σε δέσμες εντολών (scripts) για την ελεγχόμενη συλλογή δεδομένων, την παρακολούθηση αποκρίσεων σε αιτήματα HTTP και την ανάλυση της συμπεριφοράς εφαρμογών σε πραγματικό χρόνο (real-time application behaviour analysis). Ενδεικτικό παράδειγμα χρήσης του εργαλείου γραμμής εντολών του cURL project για την αποστολή αιτήματος HTTP στην εφαρμογή διαχείρισης βάσεων δεδομένων Orion Context Broker (που χρησιμοποιήθηκε ως διακομιστής στις τοπολογίες της παρούσας Διπλωματικής Εργασίας) παρατίθεται στην εικόνα 6.6.

```
curl <orion_host>:1026/v2/entities -s -S --header 'Content-Type: application/json' \
    -X POST -d @- <<EOF
{
  "id": "Room2",
  "type": "Room",
  "temperature": {
    "value": 23,
    "type": "Number"
  },
  "pressure": {
    "value": 720,
    "type": "Number"
  }
}
EOF
```

Εικόνα 6.6: Παράδειγμα χρήσης του εργαλείου γραμμής εντολών του cURL project για τη δημιουργία οντότητας(entity) στη βάση δεδομένων με την οποία επικοινωνεί η εφαρμογή Orion Context Broker

Το cURL project, ως βιβλιοθήκη (libcurl), ενσωματώνεται συχνά σε εφαρμογές λογισμικού για να παρέχει ολοκληρωμένες λειτουργίες δικτυακής επικοινωνίας, χωρίς να απαιτείται η ανάπτυξη προσαρμοσμένης υλοποίησης πρωτοκόλλων. Σε υλοποιήσεις που εστιάζουν στην ανάπτυξη δικτύων IoT, το project cURL διευκολύνει αποφασιστικά την επικοινωνία μεταξύ συσκευών, αισθητήρων και υπηρεσιών “νέφους” (cloud). Παράλληλα, στην ανάπτυξη ιστοσελίδων, RESTful APIs και μικροϋπηρεσιών (microservices), το project cURL λειτουργεί ως βασικό εργαλείο για τη δοκιμή της ορθότητας των τερματικών σημείων των δικτύων (endpoints), λειτουργώντας κατα αυτόν τον τρόπο σαν ένα σταθερό, απλό και διαχρονικό εργαλείο δομικής διεπαφών (API testing tool).

Η δυνατότητα του project cURL να υποστηρίζει εκδόσεις του πρωτοκόλλου TLS και σύγχρονα κρυπτογραφικά πρότυπα του προσδίδει ιδιαίτερα σημαντικά χαρακτηριστικά όσον αφορά την ασφάλεια της μετάδοσης δεδομένων, επιτρέποντας την επαλήθευση πιστοποιητικών (certificate validation), τον έλεγχο ανθεκτικότητας σε ενδιάμεσους επιτιθέμενους (man-in-the-middle attack durability), και την ενσωμάτωση σε συστήματα ανάλυσης ασφαλείας. Έτσι το project cURL αποτελεί ένα από τα σημαντικότερα εργαλεία για τη δημιουργία και διατήρηση ασφαλών καναλιών επικοινωνίας, αλλά και για την αυτοματοποίηση του ελέγχου και της επαλήθευσης της ασφαλούς μεταφοράς δεδομένων.

Κλείνοντας, είναι σημαντικό να υπογραμμιστεί ότι το cURL, αν και συχνά θεωρείται εργαλείο χαμηλού επιπέδου, μπορεί να εμφανίζεται σε πληθώρα υλοποιήσεων επιπέδου “παραγωγής” (production level) ως κρίσιμο δομικό στοιχείο, ιδιαίτερα όταν απαιτείται η αλληλεπίδραση με απομακρυσμένους διακομιστές (servers), APIs, ή η αξιολόγηση υποδομών δικτύων (network infrastructure assessment). Η ικανότητα του project cURL να παραμένει αξιόπιστο, φορητό και ευέλικτο το καθιστά ένα σταθερό σημείο αναφοράς σε συστήματα λογισμικού, διαδικτυακές υπηρεσίες και την ασφαλή επικοινωνία εφαρμογών.

6.2.4 Η βιβλιοθήκη του cURL project - libcurl()

Η βιβλιοθήκη libcurl() αποτελεί έναν από τους ακρογωνιαίους λίθους του project cURL, προσφέροντας ένα ισχυρό και ευέλικτο σύνολο εργαλείων για τη μεταφορά δεδομένων μέσω διαφόρων πρωτοκόλλων. Παρέχει δύο διεπαφές προγραμματισμού, τη διεπαφή “Easy” και τη διεπαφή “Multi”. Η διεπαφή προγραμματισμού “Multi” παρέχει δυνατότητα πραγματοποίησης πολλαπλών μεταφορών δεδομένων ανά thread και προσφέρεται επίσης για ασύγχρονο προγραμματισμό (non-blocking). Η διεπαφή "Easy" της libcurl() είναι σχεδιασμένη για να παρέχει μια απλοποιημένη προσέγγιση στη χρήση των λειτουργιών της βιβλιοθήκης, καθιστώντας την προσιτή σε προγραμματιστές διαφόρων επιπέδων. Η libcurl() έχει καθιερωθεί ως μια από τις πιο ευρέως χρησιμοποιούμενες βιβλιοθήκες για μεταφορά δεδομένων στο λογισμικό ανοιχτού κώδικα.

Η διεπαφή "Easy" της libcurl() παρέχει ένα σύνολο συναρτήσεων που επιτρέπουν στους προγραμματιστές να εκτελούν λειτουργίες μεταφοράς δεδομένων με σχετικά απλό τρόπο. Η βασική ροή χρήσης της διεπαφής "Easy" περιλαμβάνει τα εξής βήματα:

1. Αρχικοποίηση ενός “easy handle”, ενός τύπου μεταβλητής δηλαδή που χρησιμοποιείται ως identifier από τις διάφορες συναρτήσεις της βιβλιοθήκης για τον χειρισμό κάθε μεταφοράς δεδομένων
2. Ρύθμιση επιλογών για τη μεταφορά δεδομένων
3. Εκτέλεση της μεταφοράς δεδομένων
4. Καθαρισμός και απελευθέρωση πόρων

Στο παράδειγμα χρήσης του “Easy” API της βιβλιοθήκης libcurl() που παρουσιάζεται στην εικόνα 6.7, η συνάρτηση curl_easy_init() δημιουργεί ένα νέο “easy handle”, η curl_easy_setopt() ρυθμίζει τις επιλογές (στην προκειμένη περίπτωση, τη διεύθυνση URL), η curl_easy_perform() εκτελεί τη μεταφορά, και η curl_easy_cleanup() απελευθερώνει τους πόρους.

Η ευελιξία της libcurl() επεκτείνεται πέρα από τις βασικές λειτουργίες του πρωτοκόλλου HTTP. Υποστηρίζει ένα ευρύ φάσμα πρωτοκόλλων, συμπεριλαμβανομένων των FTP, Secure Shell File Transfer Protocol (SFTP), Secure Copy Protocol (SCP), TELNET και πολλών άλλων. Αυτή η πολυμορφία την καθιστά ιδιαίτερα χρήσιμη σε ποικίλα περιβάλλοντα ανάπτυξης και εφαρμογών. Η ασφάλεια είναι ένας κρίσιμος παράγοντας στη μεταφορά δεδομένων, και η libcurl() παρέχει ισχυρή υποστήριξη για ασφαλείς συνδέσεις. Υποστηρίζει πρωτόκολλα όπως το TLS, επιτρέποντας κρυπτογραφημένες συνδέσεις.

Η διεπαφή "Easy" της libcurl() προσφέρει επίσης προηγμένες λειτουργίες, όπως η δυνατότητα χειρισμού πολλαπλών μεταφορών ταυτόχρονα, η διαχείριση cookies, η υποστήριξη ταυτοποίησης και πολλά άλλα. Αυτές οι λειτουργίες μπορούν να ρυθμιστούν μέσω της συνάρτησης curl_easy_setopt(), η οποία δέχεται ένα ευρύ φάσμα επιλογών. Η απλότητα και η αποτελεσματικότητα της διεπαφής "Easy" έχουν συμβάλλει σημαντικά στην ευρεία υιοθέτηση της libcurl(). Ωστόσο, για πιο σύνθετες εφαρμογές, η βιβλιοθήκη προσφέρει, όπως αναφέρθηκε συνοπτικά παραπάνω, μια πιο προηγμένη διεπαφή, γνωστή ως "Multi".

```c
#include <curl/curl.h>

int main(void)
{
    CURL *curl;
    CURLcode res;

    curl = curl_easy_init();
    if(curl) {
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");
        res = curl_easy_perform(curl);
        if(res != CURLE_OK)
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(res));
        curl_easy_cleanup(curl);
    }
    return 0;
}
```

Εικόνα 6.7: Συνοπτικό παράδειγμα χρήσης του “Easy” Application Programming Interface της βιβλιοθήκης libcurl()

Συμπερασματικά, η διεπαφή "Easy" της libcurl() αποτελεί ένα ισχυρό εργαλείο για τους προγραμματιστές που επιθυμούν να ενσωματώσουν λειτουργίες μεταφοράς δεδομένων στις εφαρμογές τους. Η απλότητα χρήσης της, σε συνδυασμό με την ευελιξία και την ισχυρή υποστήριξη ασφαλείας, την καθιστούν μια αξιόπιστη επιλογή για ένα ευρύ φάσμα εφαρμογών, από απλά scripts μέχρι σύνθετα συστήματα επιχειρήσεων.

6.2.5 Εργαλεία ανάπτυξης λογισμικού της πλατφόρμας OpenZiti για τη γλώσσα προγραμματισμού C

Το SDK της πλατφόρμας OpenZiti για τη γλώσσα προγραμματισμού C αποτελεί ένα ισχυρό εργαλείο για προγραμματιστές που επιθυμούν να δημιουργήσουν εφαρμογές που να μπορούν να ενσωματώνονται σε δίκτυα που υιοθετούν Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης, χρησιμοποιώντας την πλατφόρμα ανοικτού κώδικα OpenZiti. Τα εν λόγω SDKs αξιοποιούν αυτή τη δύναμη μέσω APIs “υψηλού επιπέδου” (“high-level APIs”) που επιτρέπουν στους προγραμματιστές να ενσωματώσουν τη δυνατότητα σύνδεσης

με δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti στις ίδιες τις εφαρμογές τους, εξαλείφοντας την ανάγκη πρόσβασης ενός κόμβου δικτύου συνολικά σε ένα δίκτυο Μηδενικής Εμπιστοσύνης και όποιους πιθανούς κινδύνους εγκυμονούν τέτοιες προσεγγίσεις στο πεδίο της Ασφάλειας.

Το συγκεκριμένο SDK παρέχει μια σειρά από κρίσιμες λειτουργίες. Σε πρώτο επίπεδο αναλαμβάνει την αρχικοποίηση του περιβάλλοντος "πελάτη άκρου" (Edge-Client) στα πλαίσια ενός δικτύου Μηδενικής Εμπιστοσύνης OpenZiti, θέτοντας έτσι τη βάση ώστε να είναι δυνατόν οι εφαρμογές είτε να έχουν πρόσβαση σε υπηρεσίες ως "πελάτες" είτε να "προσφέρουν" (host) οι ίδιες υπηρεσίες ως διακομιστές. Προς επίτευξη των παραπάνω το SDK προσφέρει διεπαφή προγραμματισμού υψηλού επιπέδου (high-level API) για ταυτοποίηση χρηστών/εφαρμογής με χρήση ψηφιακών πιστοποιητικών τύπου X.509.

Παράλληλα, κάθε instance "πελάτη άκρου" (Edge-Client) μπορεί να συλλέγει και να υποβάλλει στον ελεγκτή (controller) του δικτύου Μηδενικής Εμπιστοσύνης OpenZiti, πληροφορίες σχετικά με διάφορες παραμέτρους που σχετίζονται με τη συσκευή στην οποία εκτελείται (π.χ. Λειτουργικό Σύστημα, Έκδοση Λειτουργικού Συστήματος, διεύθυνση MAC της κάρτας δικτύου που χρησιμοποιείται για επικοινωνία με το δίκτυο OpenZiti και άλλα). Τέλος υφίστανται και οι κατάλληλες διεπαφές (high-level interfaces) για την πρόσβαση σε υπηρεσίες ενός δικτύου Μηδενικής Εμπιστοσύνης OpenZiti ως πελάτη (client) αλλά και για την "παροχή" (hosting) υπηρεσιών ως διακομιστής (server) μέσω sockets.

Ως καλύτερη μέθοδος ενσωμάτωσης ενσωμάτωσης της βιβλιοθήκης της πλατφόρμας OpenZiti για τη γλώσσα προγραμματισμού C προκρίνεται η συμπερίληψη της βιβλιοθήκης μέσω του δημοφιλούς εργαλείου ανάπτυξης λογισμικού CMake, που χρησιμοποιήθηκε και στα πλαίσια της παρούσας Διπλωματικής Εργασίας. Ενδεικτικό παράδειγμα περιεχομένου αρχείου "CMakeLists.txt", για την χρήση του SDK μέσω του εργαλείου ανάπτυξης λογισμικού cmake, παρατίθεται στην εικόνα 6.8.

```
FetchContent_Declare(ziti-sdk-c
        GIT_REPOSITORY https://github.com/openziti/ziti-sdk-c.git
        GIT_TAG ${LATEST_ZITI_RELEASE}
        )
FetchContent_MakeAvailable(ziti-sdk-c)

# ...
# ...

add_executable(most-secure-app-ever ${my_sources})
target_link_libraries(most-secure-app-ever PRIVATE ziti)
```

Εικόνα 6.8: Παράδειγμα περιεχομένου αρχείου "CMakeLists.txt" για την ενσωμάτωση του OpenZiti SDK-C σε cmake build

Το SDK της πλατφόρμας Μηδενικής Εμπιστοσύνης OpenZiti για τη γλώσσα προγραμματισμού C βασίζεται και σε άλλες βιβλιοθήκες για να επιτελεί τις λειτουργίες του. Πιο συγκεκριμένα χρησιμοποιείται η βιβλιοθήκη "libuv" για τον χειρισμό βρόχων

συμβάντων (event loop), η βιβλιοθήκη “openssl” για την υλοποίηση συνδέσεων πρωτοκόλλου TLS, η βιβλιοθήκη “zlib” για τη συμπίεση των αιτημάτων HTTP, η βιβλιοθήκη “llhttp” για την ανάλυση των αιτημάτων HTTP και τέλος η βιβλιοθήκη libsodium για εφαρμογή κρυπτογράφησης από άκρο σε άκρο (E2EE). Κεντρικό χαρακτηριστικό των διεπαφών του SDK, το οποίο έδωσε και τη δυνατότητα για από κοινού χρήση με τη βιβλιοθήκη του cURL project, είναι ο χειρισμός των συνδέσεων (είτε ως πελάτης, είτε ως διακομιστής) μέσω file descriptors που αναφέρονται σε sockets.

Η αρχικοποίηση και ο τερματισμός ενός instance “πελάτη άκρου” (edge-client) πραγματοποιείται με κλήση τριών βασικών συναρτήσεων. Η συνάρτηση “Ziti_lib_init()” αρχικοποιεί το περιβάλλον του “πελάτη άκρου” (edge-client) και τον αντίστοιχο βρόχο συμβάντων στο παρασκήνιο (background event loop) που εκτελεί όλες τις λειτουργίες του. Η κλήση της συνάρτησης “Ziti_load_context()” φορτώνει μια ταυτότητα δικτύου Μηδενικής Εμπιστοσύνης OpenZiti από το αντίστοιχο αρχείο (το περιεχόμενο και η λειτουργία των αρχείων ταυτότητας OpenZiti περιγράφονται στην ενότητα 5.1.1).

Τέλος η συνάρτηση “Ziti_lib_shutdown()” τερματίζει ομαλά όλες τις λειτουργίες του “πελάτη άκρου” (Edge-Client) και τερματίζει τον βρόχο συμβάντων (event loop). Στην εικόνα 6.9 παρουσιάζεται απόσπασμα κώδικα που παρουσιάζει ακροθιγώς την αρχικοποίηση, ταυτοποίηση χρήστη και τον τερματισμό ενός “πελάτη άκρου” (Edge-Client) μέσω του SDK της πλατφόρμας δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti για τη γλώσσα προγραμματισμού C.

```c
#include <ziti/zitilib.h>

int main(int argc, char *argv[]) {

    const char *identity_file = process_args(argc, argv);

    Ziti_lib_init();
    ziti_context ztx = Ziti_load_context(identity_file);

    ...


    Ziti_lib_shutdown();
}
```

Εικόνα 6.9: Παράδειγμα χρήσης της βιβλιοθήκης zitilib() για την αρχικοποίηση, τον τερματισμό πελάτη “άκρου”(Edge-Client) και την ταυτοποίηση χρήστη

Μόλις φορτωθεί η δομή δεδομένων “ziti_context”, μπορεί να χρησιμοποιηθεί για πρόσβαση σε μια υπηρεσία δικτύου OpenZiti ως πελάτης (client) ή την παροχή της ως διακομιστής, με την προϋπόθεση φυσικά ότι η ταυτότητα που χρησιμοποιείται έχει τα κατάλληλα δικαιώματα πρόσβασης. Προκειμένου να επιτευχθεί πρόσβαση σε υπηρεσίες δικτύου OpenZiti ως πελάτης μπορούν να χρησιμοποιηθούν οι συναρτήσεις

“Ziti_connect()” ή “Ziti_connect_addr()”. Παράδειγμα χρήσης του high-level API του SDK της πλατφόρμας OpenZiti για πρόσβαση σε υπηρεσίες ως πελάτης παρουσιάζεται στην εικόνα 6.10.

Το OpenZiti C SDK παρέχει ένα ισχυρό σύνολο εργαλείων για την ανάπτυξη εφαρμογών που ενσωματώνουν Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης. Με την παροχή απλών και οικείων διεπαφών προγραμματισμού, επιτρέπει τη δημιουργία ασφαλών εφαρμογών δικτύου με σχετική ευκολία, ενώ παράλληλα εκμεταλλεύεται τα προηγμένα χαρακτηριστικά ασφαλείας της πλατφόρμας OpenZiti.

```
ziti_socket_t sock = Ziti_socket(SOCK_STREAM);
int error = Ziti_connect(sock, ztx, "my-secure-service", NULL);

// use sock as normal socket
do {
    write(sock, ...);
    read(sock, ...);

} while (!done);

close(sock);
```

Εικόνα 6.10: Παράδειγμα χρήσης της βιβλιοθήκης zitilib() για δημιουργία socket και σύνδεση του με δρομολογητή “άκρου” δικτύου OpenZiti

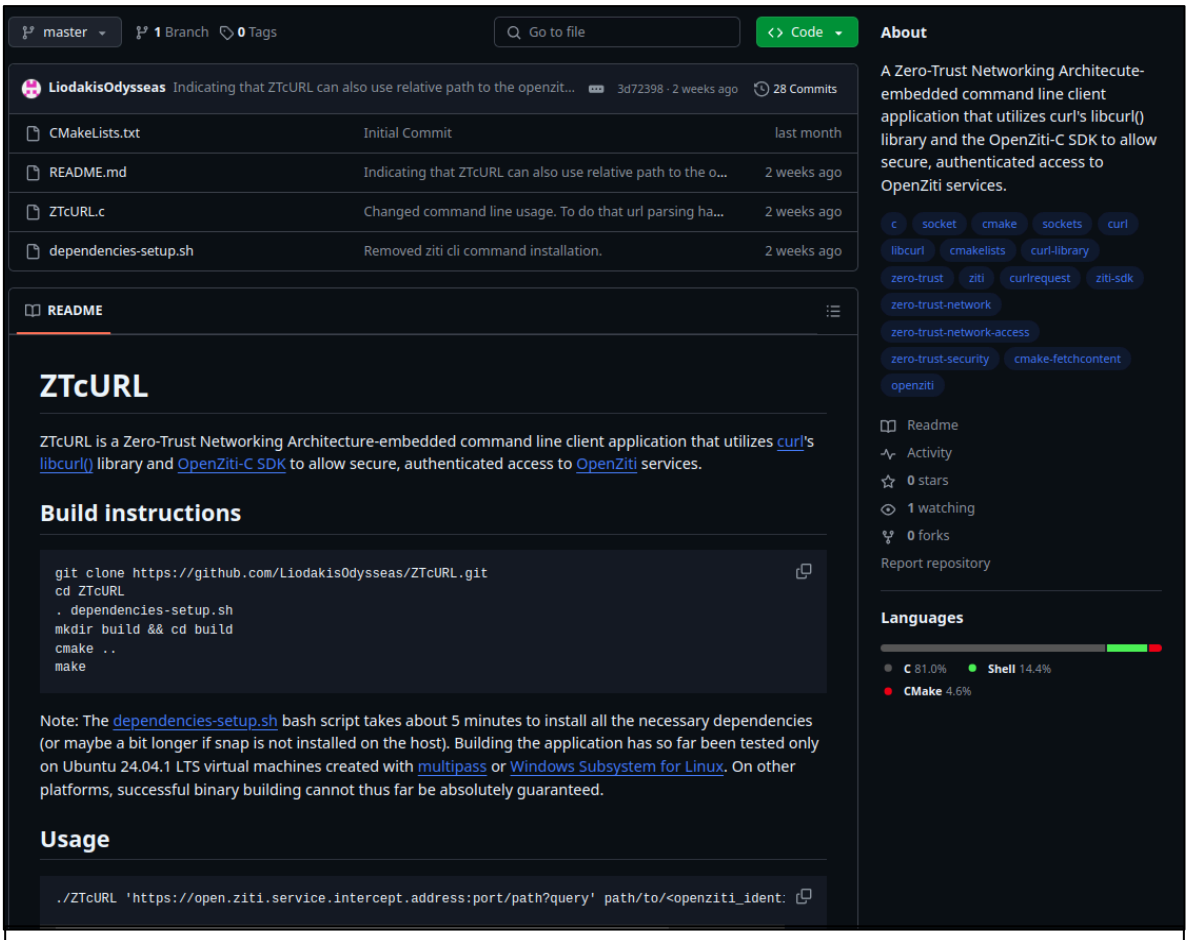
6.3 Η εφαρμογή γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας

Στα πλαίσια της παρούσας διπλωματικής εργασίας και σύμφωνα με το κεντρικό ζητούμενο της, δηλαδή τη χρήση του project ανοικτού κώδικα cURL κατά τέτοιο τρόπο ώστε να έχει τη δυνατότητα να αποστέλλει αιτήματα (requests) HTTP στην εφαρμογή διαχείρισης περιεχομένου βάσεων δεδομένων Orion Context Broker της πλατφόρμας FIWARE, μέσω δικτύων που ενσωματώνουν Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης, χρησιμοποιώντας την πλατφόρμα OpenZiti, αναπτύχθηκε εφαρμογή γραμμής εντολών που ονομάστηκε “ZTcURL” (Zero-Trust cURL).

Η εφαρμογή που αναπτύχθηκε έχει τη δυνατότητα να ενσωματώνεται σε δίκτυα που υιοθετούν Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης μέσω του SDK που προσφέρει η πλατφόρμα OpenZiti. Πιο συγκεκριμένα χρησιμοποιήθηκαν οι συναρτήσεις (functions), οι τύποι μεταβλητών και οι δομές δεδομένων (data-structures) που προσφέρει η πλατφόρμα OpenZiti για τη γλώσσα προγραμματισμού C. Ταυτόχρονα η εφαρμογή γραμμής εντολών που αναπτύχθηκε κάνει χρήση της βασικής βιβλιοθήκης συναρτήσεων και τύπων μεταβλητών του project cURL, της libcurl().

Αξιοποιώντας το SDK η εφαρμογή που αναπτύχθηκε καταφέρνει να μεταφέρει στους χρήστες της μια εμπειρία ασφαλούς (secure) και ταυτοποιημένης (authenticated)

πρόσβασης σε υπηρεσίες δικτύων OpenZiti (OpenZiti services) και εν προκειμένω, στην υλοποίηση της παρούσας διπλωματικής εργασίας, σε instances της εφαρμογής Orion Context Broker και συνακόλουθα στις βάσεις δεδομένων με τις οποίες επικοινωνούν τα παραπάνω instances της εφαρμογής. Στη συνέχεια θα παρουσιαστεί συνοπτικά η λειτουργία της εφαρμογής γραμμής εντολών.



Εικόνα 6.11: Markdown του GitHub repository της εφαρμογής γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας

Ο πηγαίος κώδικας της εφαρμογής που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας μπορεί να βρεθεί στο repository που έχει δημιουργηθεί για το σχετικό εγχείρημα (project) στη δημοφιλή ιστοσελίδα διαχείρισης πηγαίου κώδικα GitHub [48]. Στην εικόνα 6.11 παρουσιάζεται η αρχική σελίδα του repository, όπου γίνεται συνοπτική περιγραφή της λειτουργίας της εφαρμογής, ενώ παρέχονται και οδηγίες σχετικά με τη δημιουργία του εκτελέσιμου αρχείου της εφαρμογής όσο και για τη χρήση του.

Η εφαρμογή “ZTcURL” εκμεταλλεύεται τη δυνατότητα που δίνει η βιβλιοθήκη libcurl() του project cURL στον προγραμματιστή να ορίζει ο ίδιος το socket μέσω του οποίου πραγματοποιείται ένα αίτημα (request) HTTP και όχι η ίδια η βιβλιοθήκη αυτόματα με την κλήση συστήματος “socket()”. Ταυτόχρονα, χρησιμοποιώντας το high-level API του SDK της πλατφόρμας OpenZiti για τη γλώσσα προγραμματισμού C, φροντίζει τα sockets που χρησιμοποιεί για την αποστολή των αιτημάτων (requests) HTTP, να έχουν συνδεθεί με κατάλληλο δρομολογητή “άκρου” του εκάστοτε δικτύου Μηδενικής Εμπιστοσύνης OpenZiti.

Με αυτό τον τρόπο διασφαλίζεται ότι το αίτημα (request) HTTP θα μεταφερθεί μέσω του δικτύου OpenZiti προς τον επιθυμητό διακομιστή. Στην εικόνα 6.12 παρουσιάζεται το πιο κρίσιμο κομμάτι από τον πηγαίο κώδικα της εφαρμογής "ZTcURL". Με την εντολή "curl_easy_setopt(curl, CURLOPT_OPENSOCKETFUNCTION, opensocket);" (στη γραμμή 154 της εικόνας 6.12) ορίζεται ότι η βιβλιοθήκη libcurl() θα χρησιμοποιήσει το socket (τύπου curl_socket_t) που θα επιστρέψει η συνάρτηση με όνομα "opensocket" για αιτήματα (requests) HTTP που αφορούν το "easy handle" που δίνεται ως πρώτη παράμετρος (τα "easy handles" του "Easy" API της βιβλιοθήκης libcurl έχουν περιγραφεί στην ενότητα 6.2.4). Παράλληλα, η συνάρτηση "opensocket" είναι κατάλληλα διαμορφωμένη ώστε να επιστρέφει socket συνδεδεμένο με κατάλληλο δρομολογητή "άκρου" του εκάστοτε δικτύου Μηδενικής Εμπιστοσύνης OpenZiti.

Με την εκκίνηση της εκτέλεσης της εφαρμογής πραγματοποιείται ανάλυση του URL που έχει δοθεί ως όρισμα γραμμής εντολών ώστε να μπορεί να χρησιμοποιηθεί τμηματικά από κατάλληλη συνάρτηση, όπως θα παρουσιαστεί στη συνέχεια. Από τις πιο βασικές λειτουργίες της εφαρμογής γραμμής εντολών είναι η αρχικοποίηση και ο τερματισμός αντίστοιχα (μετά τη λήψη απόκρισης από τον διακομιστή) του περιβάλλοντος OpenZiti, αλλά και η ταυτοποίηση του χρήστη μέσω σχετικού αρχείου ταυτότητας OpenZiti, όπως αναλύθηκαν στην ενότητα 6.2.5 της παρούσας Διπλωματικής Εργασίας.

```
154                                 curl_easy_setopt(curl, CURLOPT_OPENSOCKETFUNCTION, opensocket);
155
156                                 Ziti_lib_init();
157
158                                 const char *identity_file = argv[2];
159
160                                 ziti_context ztx = Ziti_load_context(identity_file);
161
162                                 ziti_socket_t sock = Ziti_socket(SOCK_STREAM);
163
164                                 Ziti_connect_addr(sock, host, (unsigned int)strtoul(port, NULL, 10));
165
166                                 curl_easy_setopt(curl, CURLOPT_OPENSOCKETDATA, &sock);
167
168                                 /* call this function to set options for the socket */
169                                 curl_easy_setopt(curl, CURLOPT_SOCKOPTFUNCTION, sockopt_callback);
170
171                                 res = curl_easy_perform(curl);
172
173                                 printf("\n");
174
175                                 Ziti_lib_shutdown();
176
177                                 curl_easy_cleanup(curl);
```

Εικόνα 6.12: Απόσπασμα από τον κώδικα της εφαρμογής γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας

Με την εντολή "curl_easy_setopt(curl, CURLOPT_OPENSOCKETDATA, &sock);" (γραμμή 166 στην εικόνα 6.12) καθορίζεται ότι η συνάρτηση (σε αυτή την περίπτωση η "opensocket") που έχει οριστεί να επιστρέφει το socket που θα χρησιμοποιηθεί για τα αιτήματα (requests) HTTP που αφορούν το "easy handle" που δίνεται ως πρώτη παράμετρος (δηλαδή η συνάρτηση "opensocket" στην περίπτωση του κώδικα της εικόνας 6.12), θα λάβει ως πρώτη της παράμετρο της ("void *clientp", όπως φαίνεται στον ορισμό της συνάρτησης στην εικόνα 6.13) δείκτη σε socket που δίνεται ως τρίτη παράμετρος στην συνάρτηση "curl_easy_setopt" (σε αυτή την περίπτωση "sock" - συνδεδεμένο με κατάλληλο δρομολογητή "άκρου"). Κατόπιν η συνάρτηση opensocket ορίζει ως τιμή επιστροφής της το socket, δείκτη στο οποίο έχει λάβει ως παράμετρο, αφού το έχει

μετατρέψει (cast) σε τύπο “curl_socket_t”, δηλαδή τον τύπο socket που χρησιμοποιεί η βιβλιοθήκη libcurl για την πραγματοποίηση αιτημάτων (requests) HTTP.

Το socket που τελικά θα χρησιμοποιηθεί για την αποστολή του αιτήματος (request) HTTP έχει προηγουμένως συνδεθεί με κατάλληλο δρομολογητή “άκρου” του αντίστοιχου δικτύου Μηδενικής Εμπιστοσύνης OpenZiti με την εντολή “Ziti_connect_addr(sock, host,(unsigned int)strtoul(port, NULL, 10));” (γραμμή 164 του αποσπάσματος κώδικα της εικόνας 6.12) που δέχεται ως ορίσματα επιπλέον το idn hostname ή αλλιώς “intercept address” (η λειτουργία της έχει αναλυθεί στην ενότητα 5.1.1) και τη σχετική θύρα που αντιστοιχούν στην υπηρεσία OpenZiti στην οποία επιθυμούμε να αποκτήσουμε πρόσβαση. Οι παραπάνω παράμετροι έχουν προκύψει έπειτα από ανάλυση (parsing) του URL που έχει προηγουμένως δοθεί ως όρισμα γραμμής εντολών. Τέλος με την εντολή “res = curl_easy_perform(curl);” πραγματοποιείται το αίτημα (request) HTTP.

```
 8 static curl_socket_t opensocket(void *clientp, curlsocktype purpose, struct curl_sockaddr *address){
 9
10                                                                                  curl_socket_t sockfd;
11                                                                                  sockfd = *(curl_socket_t *)clientp;
12
13
14
15
16                                                                                  return sockfd;
17
18 }
```

Εικόνα 6.13: Συνάρτηση που επιστρέφει socket για την πραγματοποίηση μεταφορών δεδομένων με χρήση της βιβλιοθήκης libcurl()

6.3.1 Χρήση της εφαρμογής γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Εργασίας

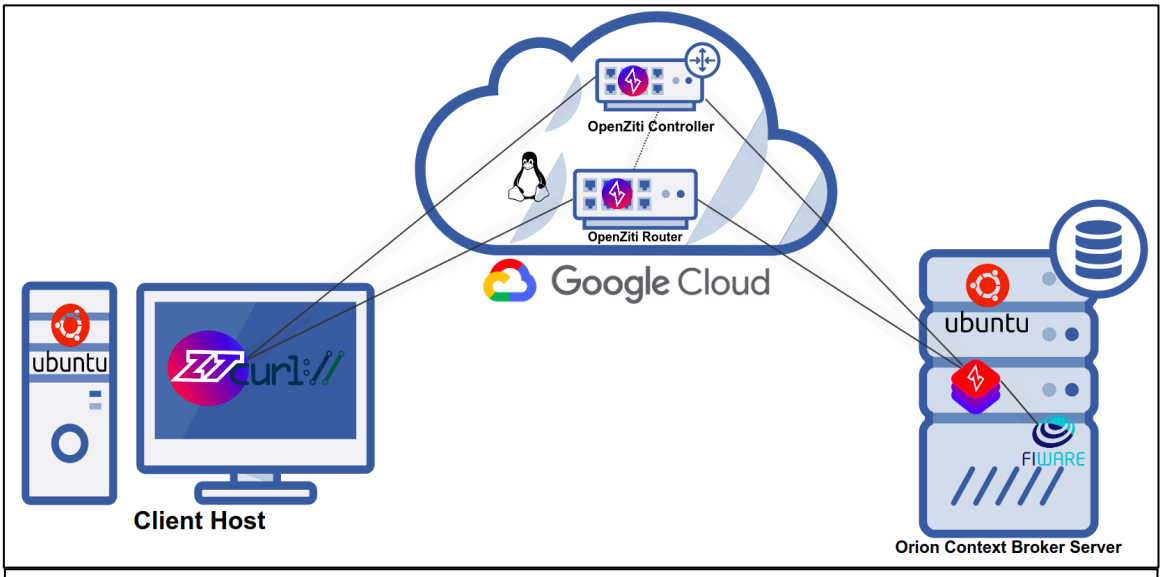
Η εφαρμογή γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας προσφέρει στον χρήστη μια εύκολη και προσιτή εμπειρία χρήσης. Όπως φαίνεται στη σχετική ενότητα του GitHub repository που φαίνεται παραπάνω και στις εικόνες που ακολουθούν δέχεται ως ορίσματα γραμμής εντολών ένα URL και τη διαδρομή (path - σχετική ή απόλυτη) προς ένα αρχείο ταυτότητας OpenZiti (έχει περιγραφεί στην ενότητα 5.1.1 της παρούσας Διπλωματικής Εργασίας), απαραίτητο για την ταυτοποίηση του χρήστη.

```
ubuntu@ZTcURL-Nthplus1-testbench:~/ZTcURL/build$ ./ZTcURL https://orion.openziti.eu:8080/v2/entities/Room2 kalogeras.json | jq .
{
  "id": "Room2",
  "type": "Room",
  "pressure": {
    "type": "Number",
    "value": 720,
    "metadata": {}
  },
  "temperature": {
    "type": "Number",
    "value": 23,
    "metadata": {}
  }
}
```

Εικόνα 6.14: Παράδειγμα χρήσης της εφαρμογής γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας

Σε περίπτωση που δεν διευκρινίζεται σχετική ή απόλυτη διαδρομή προς το αρχείο ταυτότητας OpenZiti, αλλά δίνεται ως όρισμα αποκλειστικά το όνομα αρχείου, η εφαρμογή

το αναζητά στον τρέχοντα ενεργό φάκελο. Το URL πρέπει απαραίτητα να περιλαμβάνει την intercept address (έχει περιγραφεί στην ενότητα 5.1.1 της παρούσας Διπλωματικής Εργασίας) μιας υπηρεσίας δικτύου Μηδενικής Εμπιστοσύνης OpenZiti, την αντίστοιχη θύρα και κατ' επιλογή του χρήστη ανάλογα με το endpoint του στοχευόμενου API στο οποίο επιθυμεί να έχει πρόσβαση, τα αντίστοιχα τμήματα "path" και "query". Αξίζει να σημειωθεί ότι η εφαρμογή γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας μπορεί να χρησιμοποιηθεί προσθετικά με την εφαρμογή parsing json objects "jq" ώστε τα αποτελέσματα να εκτυπώνονται με τρόπο φιλικό προς τον χρήστη όπως φαίνεται στην εικόνα 6.14. Τέλος στην εικόνα 6.15 απεικονίζεται ενδεικτική τοπολογία δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti στην οποία ενσωματώνεται και η εφαρμογή γραμμής εντολών που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας.

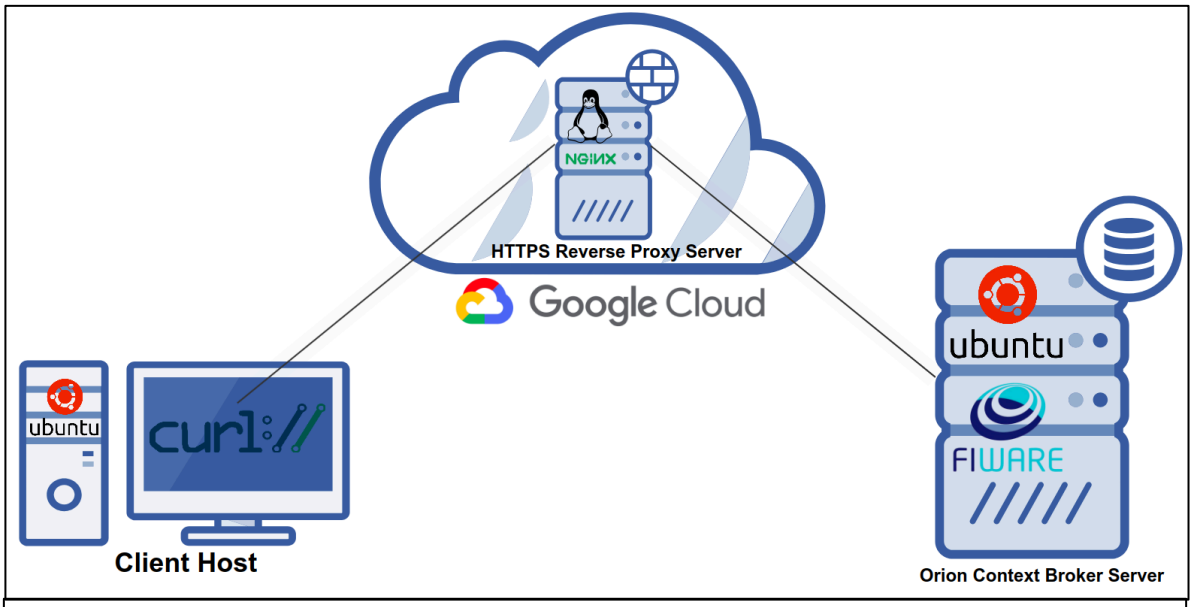


Εικόνα 6.15: Παράδειγμα τοπολογίας δικτύου που περιλαμβάνει την εφαρμογή που αναπτύχθηκε στα πλαίσια της παρούσας Διπλωματικής Εργασίας

7. Μελέτη επίδρασης της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης στις επιδόσεις εφαρμογών

7.1 Δικτυακή υποδομή που χρησιμοποιήθηκε για τη διεξαγωγή μετρήσεων

Με σκοπό τη διερεύνηση πιθανής επίδρασης της Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης στις επιδόσεις εφαρμογών διεξήχθησαν μετρήσεις καθυστέρησης μετ' επιστροφής (Round-Trip) για αιτήματα HTTP με κλιμακούμενα ωφέλιμα φορτία (payloads), που αποστέλλονταν μέσω δικτύων τα οποία είτε υιοθετούν είτε όχι την Αρχιτεκτονική. Ως εφαρμογή πελάτη χρησιμοποιήθηκε το εργαλείο γραμμής εντολών του cURL project ενώ ως διακομιστής η εφαρμογή διαχείρισης περιεχομένου βάσεων δεδομένων FIWARE Orion Context Broker.



Εικόνα 7.1: Τοπολογία δικτύου που χρησιμοποιήθηκε για τη διεξαγωγή μετρήσεων με εφαρμογή standard TLS

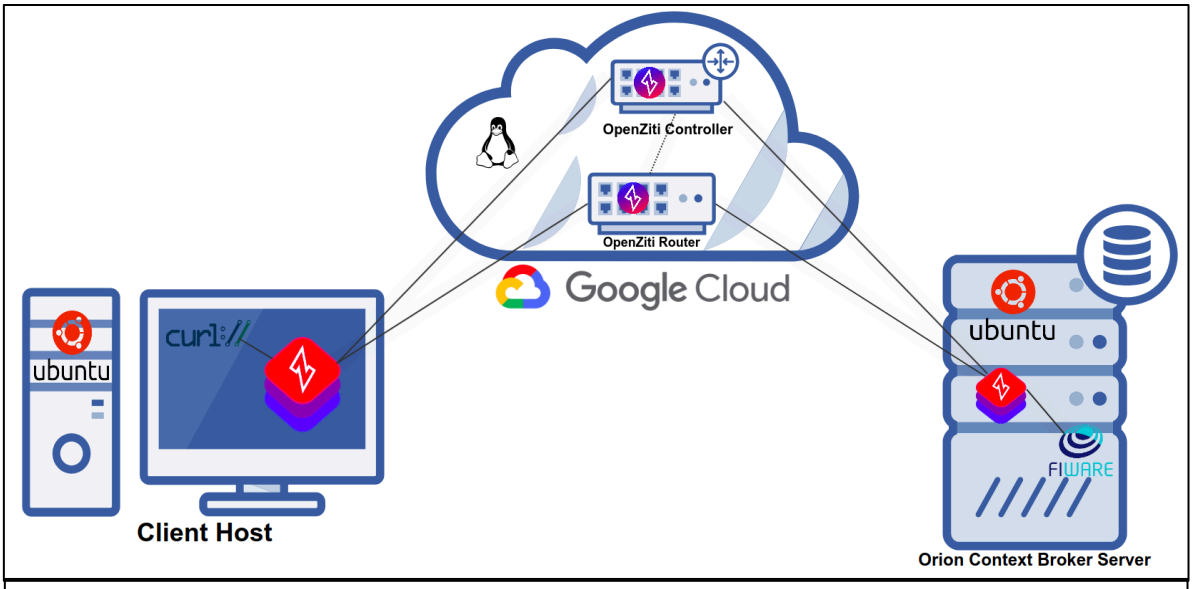
Ως κόμβοι των παραπάνω δικτύων χρησιμοποιήθηκαν εικονικά μηχανήματα που φιλοξενούνταν είτε σε υποδομή νέφους (cloud) είτε τοπικά. Ανάμεσα στην εφαρμογή πελάτη και διακομιστή παρεμβάλλονταν είτε ένας διακομιστής αντίστροφης μεσολάβησης NGINX (η ολοκληρωμένη τοπολογία του δικτύου φαίνεται στην εικόνα 7.1), είτε ένα δίκτυο Μηδενικής Εμπιστοσύνης OpenZiti (η τοπολογία του οποίου παρατίθεται στην εικόνα 7.2), ώστε να συγκριθούν οι μετρήσεις εφαρμόζοντας διαφορετική Αρχιτεκτονική Ασφάλειας στην κάθε περίπτωση.

Τόσο ο διακομιστής αντίστροφης μεσολάβησης NGINX, όσο και τα απαραίτητα για την επικοινωνία των εφαρμογών στη δεύτερη περίπτωση, στοιχεία του δικτύου Μηδενικής

Εμπιστοσύνης OpenZiti, εκτελούνταν σε εικονικά μηχανήματα σε υποδομή νέφους (cloud), ώστε να "απομονωθούν" οι εφαρμογές πελάτη και διακομιστή και κατ' αυτόν τον τρόπο να προσομοιωθούν πιο ρεαλιστικές συνθήκες δικτύου. Τα εικονικά μηχανήματα που λειτουργούσαν τοπικά και φιλοξενούσαν τις εφαρμογές πελάτη και την εφαρμογή Orion Context Broker ως διακομιστή δημιουργήθηκαν με το εργαλείο διαχείρισης εικονικών μηχανών multipass. Αναφορικά με τα μηχανήματα στην υποδομή cloud χρησιμοποιήθηκε η σχετική πλατφόρμα της Google.

7.1.1 Ανάπτυξη δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti

Το δίκτυο Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti που χρησιμοποιήθηκε για τη διεξαγωγή μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip), αναπτύχθηκε σε περιβάλλον Linux που αποτελεί και την πιο ευρέως καθιερωμένη επιλογή για την ανάπτυξη διακομιστών και στοιχείων δικτύου (ελεγκτών, δρομολογητών κλπ.) [49]. Επιλέχθηκε μια σχετικά απλή αλλά επαρκώς ενδεικτική υλοποίηση με ένα ελεγκτή (controller) και έναν δρομολογητή.

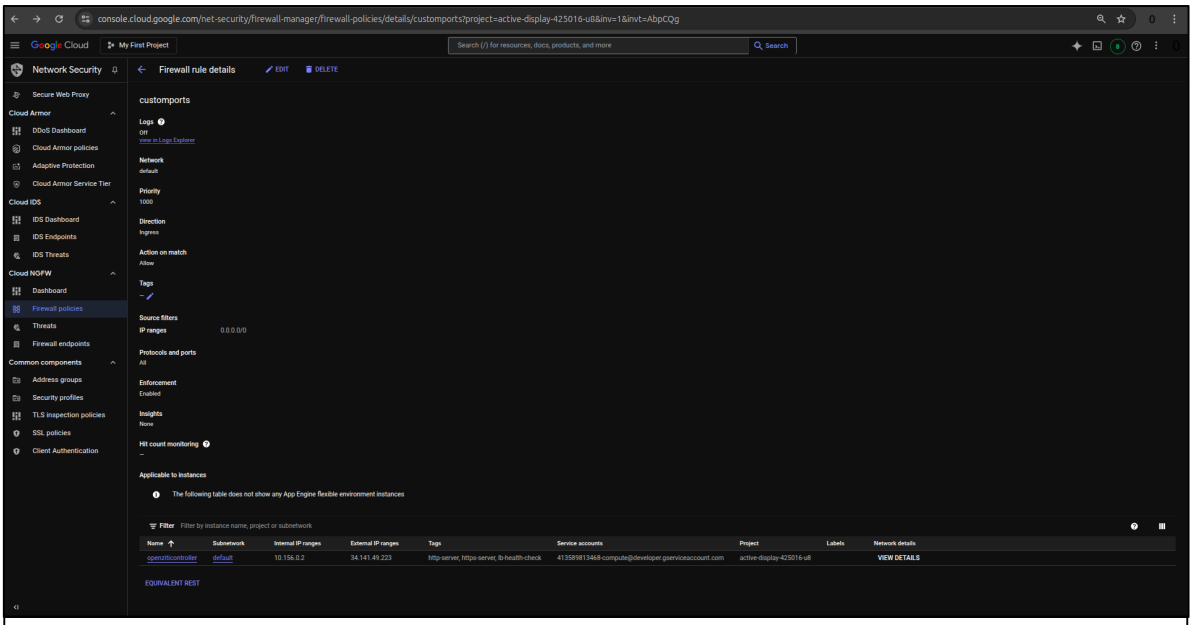


Εικόνα 7.2: Τοπολογία δικτύου που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης

Τόσο ο ελεγκτής όσο και ο δρομολογητής του δικτύου OpenZiti αναπτύχθηκαν ως υπηρεσίες systemd του Linux. Η ανάπτυξη ενός δικτύου OpenZiti αποτελεί μια πολύπλοκη διαδικασία που απαιτεί προσεκτικό σχεδιασμό και εκτέλεση. Η διαδικασία εγκατάστασης ξεκινά με την κατάλληλη διαμόρφωση του τείχους προστασίας (firewall) του εικονικού μηχανήματος στο οποίο θα φιλοξενηθούν ο ελεγκτής και ο δρομολογητής του δικτύου OpenZiti. Η παραπάνω διαδικασία πραγματοποιήθηκε μέσω του γραφικού περιβάλλοντος διαχείρισης της δικτυακής λειτουργίας των εικονικών μηχανών της Google, όπως φαίνεται στην εικόνα 7.3.

Για την εγκατάσταση του δικτύου OpenZiti, χρησιμοποιήθηκε το script "expressInstall" που παρέχεται από την πλατφόρμα. Πριν από την εκτέλεση του script

εγκατάστασης, διασφαλίστηκε η παρουσία των απαραίτητων εργαλείων όπως το "tar", "hostname", "jq" και "curl" στο εικονικό μηχάνημα. Η διαδικασία εγκατάστασης προσαρμόστηκε μέσω κατάλληλων μεταβλητών περιβάλλοντος. Δημιουργήθηκε όνομα DNS για την εγκατάσταση, καθώς παρέχει μεγαλύτερη ευελιξία σε περίπτωση αλλαγής της διεύθυνσης IP του εικονικού μηχανήματος που φιλοξενεί τον ελεγκτή και τον δρομολογητή του δικτύου OpenZiti .



Εικόνα 7.3: Διαμόρφωση κανόνα τείχους προστασίας σε εικονικό μηχάνημα της πλατφόρμας Google Cloud

Οι ρυθμίσεις για τον ελεγκτή και τον δρομολογητή, όπως οι διευθύνσεις και οι θύρες, καθορίστηκαν μέσω μεταβλητών του περιβάλλοντος, με εντολές αντίστοιχες, όπως αυτές που παρατίθενται στην εικόνα 7.4. Το script που χρησιμοποιήθηκε για την εγκατάσταση του δικτύου OpenZiti αρχικά αναζητά τα απαραίτητα εκτελέσιμα αρχεία και αν δεν τα εντοπίσει τοπικά, πραγματοποιεί λήψη τους από το σχετικό αποθετήριο στο GitHub.

```
odysseas@odysseas-IdeaPad-3-15ABA7:~$ export EXTERNAL_DNS="openziti.eu"
odysseas@odysseas-IdeaPad-3-15ABA7:~$ export EXTERNAL_IP="$(curl -s eth0.me)"
export ZITI_CTRL_EDGE_IP_OVERRIDE="${EXTERNAL_IP}"
export ZITI_CTRL_ADVERTISED_ADDRESS="${EXTERNAL_DNS:-${EXTERNAL_IP}}"
export ZITI_CTRL_ADVERTISED_PORT=8440
export ZITI_CTRL_EDGE_ADVERTISED_ADDRESS="${EXTERNAL_DNS:-${EXTERNAL_IP}}"
export ZITI_CTRL_EDGE_ADVERTISED_PORT=8441
export ZITI_ROUTER_ADVERTISED_ADDRESS="${EXTERNAL_DNS:-${EXTERNAL_IP}}"
export ZITI_ROUTER_IP_OVERRIDE="${EXTERNAL_IP}"
export ZITI_ROUTER_PORT=8442
odysseas@odysseas-IdeaPad-3-15ABA7:~$
```

Εικόνα 7.4: Προσαρμογή μεταβλητών περιβάλλοντος για την παραμετροποίηση της εγκατάστασης δικτύου Μηδενικής Εμπιστοσύνης OpenZiti

Κατόπιν πραγματοποιείται έλεγχος σχετικά με τις θύρες του εικονικού μηχανήματος που πρέπει να είναι “ανοικτές”, ώστε ο ελεγκτής και ο δρομολογητής να είναι σε θέση να

δεχθούν “εξωτερικές” συνδέσεις. Στη συνέχεια πραγματοποιείται δημιουργία υποδομής δημοσίου κλειδιού, απαραίτητης για την “υπογραφή” και διαχείριση των ψηφιακών πιστοποιητικών που χρησιμοποιούνται για τις αμοιβαία ταυτοποιημένες συνδέσεις TLS στα πλαίσια του δικτύου OpenZiti . Το script εγκατάστασης εκτυπώνει κατάλληλα μηνύματα αναφορικά με τα προαναφερθέντα βήματα της εγκατάστασης, όπως φαίνεται στην εικόνα 7.5. Μετά την εκτέλεση του script "expressInstall", χρησιμοποιήθηκε το σύστημα διαχείρισης υπηρεσιών "systemd" για την εκκίνηση του ελεγκτή και του δρομολογητή. Η παραπάνω επιλογή διασφαλίζει την αυτόματη επανεκκίνηση των υπηρεσιών σε περίπτωση επανεκκίνησης του εικονικού μηχανήματος.

```
********      Getting OpenZiti Binaries       ********
Getting OpenZiti binaries

No existing binary found, creating the ZITI_BIN_DIR directory (/home/odysseas/.ziti/quickstart/odysseas-IdeaPad-3-15ABA7/ziti-bin/ziti-v1.3.3)
Downloading https://github.com/openziti/ziti/releases/download/v1.3.3/ziti-linux-amd64-1.3.3.tar.gz to /home/odysseas/.ziti/quickstart/odysseas-
ux-amd64-1.3.3.tar.gz
OpenZiti binaries v1.3.3 successfully extracted to /home/odysseas/.ziti/quickstart/odysseas-IdeaPad-3-15ABA7/ziti-bin/ziti-v1.3.3

******** Ensure the Necessary Ports Are Open  ********
Checking Controller's port (8440) Open
Checking Edge Router's port (8442) Open
Checking Edge Controller's port (8441) Open
Checking Router Listener Bind Port's port (10080) Open
Expected ports are all available

******** Generating Public Key Infrastructure ********
Generating PKI
Creating CA: openziti.eu-root-ca
Success

Creating CA: odysseas-IdeaPad-3-15ABA7-edge-controller-root-ca
Success

Creating CA: odysseas-IdeaPad-3-15ABA7-signing-root-ca
Success

Creating intermediate: openziti.eu-root-ca openziti.eu-intermediate 1
Using CA name:  openziti.eu-root-ca
Success

Creating intermediate: odysseas-IdeaPad-3-15ABA7-edge-controller-root-ca odysseas-IdeaPad-3-15ABA7-edge-controller-intermediate 1
Using CA name:  odysseas-IdeaPad-3-15ABA7-edge-controller-root-ca
Success

Creating intermediate: odysseas-IdeaPad-3-15ABA7-signing-root-ca odysseas-IdeaPad-3-15ABA7-signing-intermediate_grandparent_intermediate 2
Using CA name:  odysseas-IdeaPad-3-15ABA7-signing-root-ca
Success

Creating intermediate: odysseas-IdeaPad-3-15ABA7-signing-intermediate_grandparent_intermediate odysseas-IdeaPad-3-15ABA7-signing-intermediate 1
Using CA name:  odysseas-IdeaPad-3-15ABA7-signing-intermediate_grandparent_intermediate
Success
```

Εικόνα 7.5: Λήψη εκτελέσιμων αρχείων, έλεγχος θυρών και δημιουργία υποδομής δημοσίου κλειδιού κατά την εγκατάσταση δικτύου Μηδενικής Εμπιστοσύνης OpenZiti

Η ανάπτυξη δικτυακών στοιχείων ως υπηρεσίες systemd σε συστήματα Linux αποτελεί μια σύγχρονη προσέγγιση στη διαχείριση δικτύων. Το systemd, ως ο κύριος διαχειριστής διεργασιών συστήματος και υπηρεσιών σε πολλές σύγχρονες διανομές Linux, παρέχει ένα ισχυρό πλαίσιο για την εκκίνηση, διαχείριση και παρακολούθηση υπηρεσιών. Η ενσωμάτωση δικτυακών στοιχείων σε αυτό το πλαίσιο προσφέρει πολλαπλά πλεονεκτήματα σε επίπεδο διαχείρισης και ασφάλειας.

Η διαδικασία ανάπτυξης δικτυακών στοιχείων ως υπηρεσίες systemd περιλαμβάνει τη δημιουργία κατάλληλων αρχείων (unit files). Αυτά τα αρχεία καθορίζουν τον τρόπο εκκίνησης, τερματισμού και διαχείρισης κάθε δικτυακού στοιχείου. Ένα σημαντικό πλεονέκτημα αυτής της προσέγγισης είναι η αυτοματοποιημένη διαχείριση εξαρτήσεων (dependencies). Η ασφάλεια αποτελεί άλλο ένα σημαντικό όφελος της χρήσης του systemd για τη διαχείριση δικτυακών στοιχείων.

Το systemd παρέχει εκτεταμένες δυνατότητες απομόνωσης και περιορισμού πρόσβασης, επιτρέποντας τον καθορισμό λεπτομερών πολιτικών ασφαλείας για κάθε υπηρεσία του. Αυτό περιλαμβάνει τη δυνατότητα εκτέλεσης υπηρεσιών σε απομονωμένα περιβάλλοντα, περιορίζοντας την πρόσβαση τους σε συγκεκριμένους πόρους του συστήματος [50][51].

Εικόνα 7.6: Έλεγχος κατάστασης υπηρεσιών ελεγκτή και δρομολογητή δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti

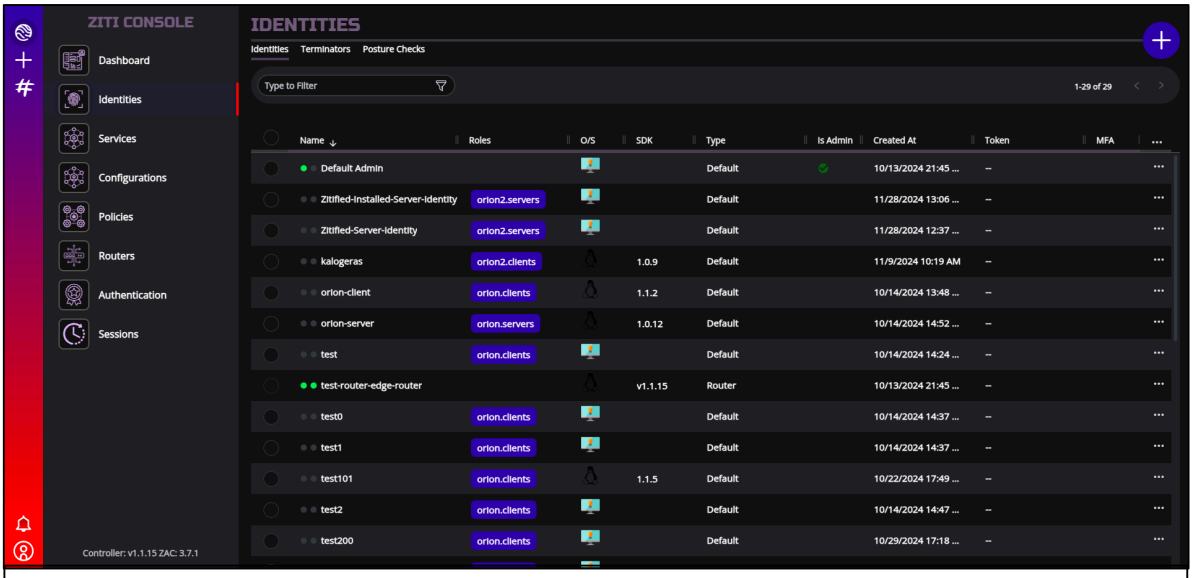
Επιπλέον, η χρήση του systemd για τη διαχείριση δικτυακών στοιχείων επιτρέπει την εύκολη παρακολούθηση και διάγνωση προβλημάτων. Η ευελιξία που προσφέρει το systemd στη διαμόρφωση των υπηρεσιών είναι ιδιαίτερα χρήσιμη για τη διαχείριση πολύπλοκων δικτυακών τοπολογιών. Μέσω των αρχείων διαμόρφωσης του systemd, μπορούν να οριστούν λεπτομερείς ρυθμίσεις για κάθε δικτυακό στοιχείο, συμπεριλαμβανομένων των παραμέτρων εκκίνησης, και των μεταβλητών περιβάλλοντος. Το systemd παρέχει ενοποιημένη καταγραφή συμβάντων και εργαλεία για την παρακολούθηση της κατάστασης των υπηρεσιών, διευκολύνοντας τη γρήγορη αντιμετώπιση προβλημάτων και την ανάλυση της επίδοσης δικτύων.

Πιο συγκεκριμένα μέσω του συστήματος διαχείρισης υπηρεσιών systemd μπορούν να ελεγχθούν ποικίλες παράμετροι αναφορικά με την κατάσταση των υπηρεσιών όπως η θέση των αρχείων διαμόρφωσής τους, η επιλογή αυτόματης εκκίνησης των υπηρεσιών κατά την εκκίνηση του συστήματος, η τρέχουσα κατάσταση των υπηρεσιών, η χρονική στιγμή εκκίνησης τους, το αναγνωριστικό (id) των αντίστοιχων διεργασιών συστήματος, η χρήση πόρων συστήματος, η ομάδα ελέγχου πυρήνα μέσω της οποίας εκτελούνται οι υπηρεσίες και η ακριβής εντολή μέσω της οποίας εκκίνησε η εκτέλεση τους. Παράδειγμα ελέγχου κατάστασης των υπηρεσιών systemd που αντιστοιχούν σε ελεγκτή και δρομολογητή δικτύου OpenZiti, παρατίθεται στην εικόνα 7.6. Μέσω των παραπάνω εντολών διασαφηνίζεται επίσης η θέση των αρχείων διαμόρφωσης που καθορίζουν τη λειτουργία του ελεγκτή και του δρομολογητή, το πόσο λεπτομερής καταγραφή συμβάντων πραγματοποιείται για κάθε διεργασία και με τι διάταξη αυτή εκτυπώνεται.

Συνολικά, η ανάπτυξη δικτυακών στοιχείων ως υπηρεσίες systemd προσφέρει ένα ισχυρό και ευέλικτο πλαίσιο για τη διαχείριση δικτυακών υποδομών σε περιβάλλοντα Linux. Αυτή η προσέγγιση συνδυάζει τα πλεονεκτήματα της αυτοματοποιημένης διαχείρισης, της ενισχυμένης ασφάλειας και της βελτιωμένης παρακολούθησης συμβάντων, καθιστώντας την ιδανική για σύγχρονα, πολύπλοκα δικτυακά περιβάλλοντα. Η υιοθέτηση αυτής της μεθόδου μπορεί να οδηγήσει σε πιο αξιόπιστα, ασφαλή και εύκολα διαχειρίσιμα δίκτυα, ιδιαίτερα σε μεγάλης κλίμακας υποδομές και περιβάλλοντα παραγωγής.

7.1.2 Διαχείριση δικτύου Μηδενικής Εμπιστοσύνης OpenZiti

Η διαχείριση των λογικών οντοτήτων του δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti, πραγματοποιήθηκε με χρήση της διεπαφής γραμμής εντολών (command line interface - CLI) που προσφέρει η πλατφόρμα. Η πλατφόρμα Δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti παρέχει επίσης δυνατότητα διαχείρισης των λογικών οντοτήτων ενός δικτύου OpenZiti και μέσω γραφικού περιβάλλοντος όπως φαίνεται στην εικόνα 7.7.



Εικόνα 7.7: Γραφικό περιβάλλον διαχείρισης λογικών οντοτήτων δικτύων Μηδενικής Εμπιστοσύνης OpenZiti

Για τη δημιουργία των λογικών οντοτήτων που χρησιμοποιήθηκαν για το δίκτυο OpenZiti, εκτελέστηκαν εντολές της διεπαφής γραμμής εντολών όμοιες με αυτές που παρατίθενται στην εικόνα 7.8. Προκειμένου να υπάρχει πρόσβαση στη διαχείριση των λογικών οντοτήτων ενός δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti μέσω της διεπαφής γραμμής εντολών, απαιτείται να έχει προηγηθεί ταυτοποίηση στον ελεγκτή του δικτύου OpenZiti και συνακόλουθα δημιουργία συνεδρίας API και εφήμερου πιστοποιητικού για τις αμοιβαία ταυτοποιημένες συνδέσεις TLS, όπως φαίνεται στην εικόνα 7.9.

Αρχικά, δημιουργήθηκαν δύο διαμορφώσεις για την υπηρεσία. Η πρώτη διαμόρφωση τύπου “host”, ορίζει τις παραμέτρους φιλοξενίας της υπηρεσίας, δηλαδή το πρωτόκολλο TCP, το idn hostname και τη θύρα μέσω των οποίων πραγματοποιείται TCP σύνδεση με την εφαρμογή FIWARE Orion Context Broker. Η δεύτερη διαμόρφωση, τύπου “intercept”, καθορίζει τις παραμέτρους, δηλαδή το πρωτόκολλο TCP, το idn hostname και τη θύρα τα οποία η εφαρμογή tunnel θα αναλύει σε διευθύνσεις IP εικονικών διεπαφών που έχει δημιουργήσει ώστε να προωθούνται τα αντίστοιχα αιτήματα HTTP μέσω του δικτύου OpenZiti.

Μετά τη δημιουργία των διαμορφώσεων, δημιουργήθηκε η ίδια η λογική οντότητα της υπηρεσίας. Η υπηρεσία συσχετίστηκε με τις προηγουμένως δημιουργημένες διαμορφώσεις και της αποδίδεται ένα ή περισσότερα χαρακτηριστικά ρόλων. Στη συνέχεια, δημιουργήθηκαν δύο ταυτότητες: η μία προκειμένου να χρησιμοποιηθεί από την περίπτωση

εφαρμογής tunnel που προορίζεται να δώσει πρόσβαση στο δίκτυο στην εφαρμογή cURL και η άλλη για την περίπτωση της εφαρμογής tunnel που προορίζεται να δώσει πρόσβαση στο δίκτυο στην εφαρμογή Orion Context Broker.

```
echo Create the service configs
ziti edge create config simple.hostv1 host.v1 '{"protocol":"tcp", "address":"localhost","port":'8080'}'
ziti edge create config simple.interceptv1 intercept.v1 '{"protocols":["tcp"],"addresses":["simpleService.ziti"], "portRanges":[{"low":'8080', "high":'8080'}]}'

echo Create the service
ziti edge create service simpleService --configs "simple.hostv1,simple.interceptv1" --role-attributes simple-service

echo Create two identities and enroll the server
ziti edge create identity user simple-client -a simpleserver.clients -o simple-client.jwt
ziti edge create identity device simple-server -a simpleserver.servers -o simple-server.jwt
ziti edge enroll --jwt simple-server.jwt

echo Create service policies
ziti edge create service-policy simple-client-dial Dial --identity-roles '#simpleserver.clients' --service-roles '#simple-service'
ziti edge create service-policy simple-client-bind Bind --identity-roles '#simpleserver.servers' --service-roles '#simple-service'

echo Run policy advisor to check
ziti edge policy-advisor services
```

Εικόνα 7.8: Ενδεικτικές εντολές διεπαφής γραμμής εντολών για τη δημιουργία λογικών οντοτήτων σε δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti

Η ταυτότητα του πελάτη συσχετίστηκε με το χαρακτηριστικό ρόλου "simpleserver.clients", ενώ η ταυτότητα του διακομιστή συσχετίστηκε με το χαρακτηριστικό ρόλου "simpleserver.servers". Για κάθε ταυτότητα δημιουργήθηκε ένα αρχείο JWT (JSON Web Token) που χρησιμοποιήθηκε για την εγγραφή της αντίστοιχης ταυτότητας στο δίκτυο OpenZiti και τη δημιουργία του αντίστοιχου αρχείου ταυτότητας όπως έχει περιγραφεί στην ενότητα 5.1.1 της παρούσας Διπλωματικής Εργασίας.

```
odysseas@odysseas-IdeaPad-3-15ABA7:~$ ziti edge login openziti.eu:8441
Cert #0 in the chain doesn't match
WARNING: server supplied certificate authority doesn't match cached certs at /home/odysseas/.config/ziti/certs/openziti.eu
Replace cached certs [Y/N]: Y
Server certificate chain written to /home/odysseas/.config/ziti/certs/openziti.eu
Using username: admin from identity 'default' in config file: /home/odysseas/.config/ziti/ziti-cli.json
Enter password:
Token: 4a7fe109-763f-4136-af3c-0ca2efd74dc3
Saving identity 'default' to /home/odysseas/.config/ziti/ziti-cli.json
odysseas@odysseas-IdeaPad-3-15ABA7:~$ ziti edge list identities
```

ID	NAME	TYPE	ATTRIBUTES	AUTH-POLICY
3aFymZFIDG	openziticontroller-edge-router	Router		Default
aJZJcsMkN	orion-client	Default	orion.clients	Default
cpxqWQPcq	Default Admin	Default		Default
dC7kcbv4Vr	orion-server	Default	orion.servers	Default

```
results: 1-4 of 4
odysseas@odysseas-IdeaPad-3-15ABA7:~$
```

Εικόνα 7.9: Ταυτοποίηση μέσω της διεπαφής γραμμής εντολών και ενδεικτική χρήση του API διαχείρισης σε δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti

Ακολούθως, δημιουργήθηκαν δύο πολιτικές υπηρεσίας (service policies). Η πρώτη πολιτική, με το όνομα "simple-client-dial", επιτρέπει στις ταυτότητες με το χαρακτηριστικό ρόλου "#simpleserver.clients" να έχουν πρόσβαση ως εφαρμογές πελάτη (dial) στην υπηρεσία με το χαρακτηριστικό ρόλου "#simple-service". Η δεύτερη πολιτική, με το όνομα "simple-client-bind", επιτρέπει στις ταυτότητες με το χαρακτηριστικό ρόλου "#simpleserver.servers" να παρέχουν (bind) την υπηρεσία με το χαρακτηριστικό ρόλου "#simple-service", μέσω του δικτύου Μηδενικής Εμπιστοσύνης OpenZiti.

Τέλος, η διεπαφή γραμμής εντολών της πλατφόρμας OpenZiti παρέχει και εργαλείο εποπτείας των δικαιωμάτων πρόσβασης για τις υπηρεσίες των δικτύων OpenZiti. Αυτή η λειτουργία είναι κρίσιμη για την επαλήθευση της σωστής εφαρμογής των πολιτικών ασφαλείας και την εγγύηση της ομαλής λειτουργίας των υπηρεσιών μέσω δικτύων

Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti. Έτσι είναι δυνατό να ελέγχεται άμεσα ποιες ταυτότητες έχουν ποια δικαιώματα πρόσβασης (dial/bind) για ποιες υπηρεσίες και πόσοι εκ των κοινών δρομολογητών τους (στους οποίους έχουν πρόσβαση οι εκάστοτε ταυτότητες και οι αντίστοιχες υπηρεσίες) είναι διαθέσιμοι.

```
OKAY : simple-client (1) -> simpleService (1) Common Routers: (1/1) Dial: Y Bind: N

OKAY : simple-server (1) -> simpleService (1) Common Routers: (1/1) Dial: N Bind: Y

ERROR: test-service
  - Service is not accessible by any identities. Adjust service policies.
```

Εικόνα 7.10: Ενδεικτικά αποτελέσματα εργαλείου "policy-advisor" της διεπαφής γραμμής εντολών της πλατφόρμας OpenZiti

Αντίστοιχα το εργαλείο εμφανίζει μήνυμα λάθους αν για κάποια υπηρεσία δεν υπάρχει τουλάχιστον μια ταυτότητα με δικαίωμα "πρόσβασης" (dial) και "παροχής" (bind) της. Ενδεικτικά αποτελέσματα του εργαλείου εποπτείας δικαιωμάτων πρόσβασης (policy advisor) παρατίθενται στην εικόνα 7.10. Συνολικά, η παραπάνω αλληλουχία εντολών δημιούργησε το σύνολο των απαραίτητων λογικών οντοτήτων για την ασφαλή, τυποποιημένη πρόσβαση στην εφαρμογή Orion Context Broker μέσω δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti.

7.1.3 Εγκατάσταση και ρύθμιση εφαρμογών tunnel

Στην περίπτωση του δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti, προκειμένου να υπάρχει δυνατότητα σύνδεσης με το δίκτυο τόσο από την εφαρμογή πελάτη όσο και από τον διακομιστή FIWARE Orion Context Broker, χρησιμοποιήθηκαν (στα αντίστοιχα εικονικά μηχανήματα που φιλοξενούνταν και οι εφαρμογές) εφαρμογές tunnel (η λειτουργία τους περιγράφεται αναλυτικά στην ενότητα 5.1.2 της παρούσας Εργασίας). Οι εφαρμογές tunnel εγκαταστάθηκαν στα αντίστοιχα εικονικά μηχανήματα βάσει των οδηγιών που παρέχονται στην τεκμηρίωση της πλατφόρμας Δικτύωσης Μηδενικής Εμπιστοσύνης OpenZiti [52].

```
ubuntu@Tunneler-Client:~$ sudo systemctl status ziti-edge-tunnel.service
● ziti-edge-tunnel.service - Ziti Edge Tunnel
     Loaded: loaded (/usr/lib/systemd/system/ziti-edge-tunnel.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-02-08 11:48:44 EET; 31s ago
    Process: 983 ExecStartPre=/opt/openziti/bin/ziti-edge-tunnel.sh (code=exited, status=0/SUCCESS)
   Main PID: 987 (ziti-edge-tunne)
      Tasks: 6 (limit: 1102)
     Memory: 11.7M (peak: 12.0M)
        CPU: 43ms
     CGroup: /system.slice/ziti-edge-tunnel.service
             └─987 /opt/openziti/bin/ziti-edge-tunnel run --verbose=2 --dns-ip-range=100.64.0.1/10 --identity-dir=/opt/openziti/etc/identities
```

Εικόνα 7.11: Έλεγχος κατάστασης διεργασίας tunnel μέσω του συστήματος διαχείρισης διεργασιών systemd

Οι εφαρμογές tunnel (όπως και ο ελεγκτής και ο δρομολογητής του δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti) αναπτύχθηκαν ως υπηρεσίες systemd. Μέσω του συστήματος διαχείρισης υπηρεσιών systemd, έχουμε τη δυνατότητα να ενημερωθούμε για το εύρος των διευθύνσεων IP το οποίο η εφαρμογή tunnel έχει τη

δυνατότητα να δεσμεύει για υπηρεσίες δικτύων Μηδενικής Εμπιστοσύνης OpenZiti. Τέτοιο παράδειγμα χρήσης του συστήματος systemd παρατίθεται στην εικόνα 7.11.

Στο ίδιο παράδειγμα ελέγχου κατάστασης της υπηρεσίας tunnel μπορούμε να δούμε τη θέση του φακέλου στον οποίο τοποθετούνται τα αρχεία ταυτότητας OpenZiti που χρειάζεται η υπηρεσία systemd για να πραγματοποιεί ταυτοποίηση σε δίκτυα OpenZiti και να αποκτά πρόσβαση στις αντίστοιχες υπηρεσίες. Επίσης, με χρήση των εφαρμογών tunnel πραγματοποιήθηκε η εγγραφή (enrollment) των αντίστοιχων ταυτοτήτων (πελάτη και διακομιστή) στο δίκτυο OpenZiti (όπως φαίνεται στην εικόνα 7.12) και η συνακόλουθη δημιουργία των αρχείων ταυτότητας OpenZiti, που χρησιμοποιούν οι εφαρμογές tunnel για αμοιβαία ταυτοποιημένες συνδέσεις με το δίκτυο Μηδενικής Εμπιστοσύνης.

```
ubuntu@Tunneler-Client:~$ ziti edge create identity test521 -a orion.clients -o test521.jwt
New identity test521 created with id: HLBqDQYL0
Enrollment expires at 2025-02-08T12:57:23.335Z
ubuntu@Tunneler-Client:~$ sudo ziti-edge-tunnel add --jwt "$(< ./test521.jwt)" --identity test521-tunnel-identity
{
  "Success":true,
  "Code":0
}
ubuntu@Tunneler-Client:~$ sudo systemctl restart ziti-edge-tunnel.service
ubuntu@Tunneler-Client:~$
```

Εικόνα 7.12: Δημιουργία και εγγραφή(enrollment) ταυτότητας σε δίκτυο Μηδενικής Εμπιστοσύνης OpenZiti μέσω της διεπαφής γραμμής εντολών και εφαρμογής tunnel

7.1.4 Διακομιστής αντίστροφης μεσολάβησης NGINX

Στην τοπολογία δικτύου που δεν υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης χρησιμοποιήθηκε διακομιστής αντίστροφης μεσολάβησης (reverse proxy server, όπως φαίνεται στην τοπολογία της εικόνας 7.2), ώστε τα δύο δίκτυα που συγκρίθηκαν να έχουν ανάλογο αριθμό κόμβων (network hops) ανάμεσα στην εφαρμογή πελάτη και στο διακομιστή FIWARE Orion Context Broker. Οι διακομιστές αντίστροφης μεσολάβησης αποτελούν κρίσιμο συστατικό σύγχρονων δικτυακών αρχιτεκτονικών, λειτουργώντας ως ενδιάμεσοι κόμβοι μεταξύ των εφαρμογών πελάτη και των υπηρεσιών. Η κύρια λειτουργία τους είναι η αποδοτική διαχείριση αιτημάτων, η εξισορρόπηση φόρτου (load balancing), η προστασία από επιθέσεις και η βελτίωση της επίδοσης μέσω προσωρινής αποθήκευσης (caching). Το NGINX διακρίνεται για την υψηλή του επίδοση και την ευελιξία στη διαμόρφωση ασφαλών και κλιμακούμενων λύσεων.

Το NGINX σχεδιάστηκε με βάση ένα μοντέλο ασύγχρονης επεξεργασίας αιτημάτων (event-driven architecture), το οποίο του επιτρέπει να χειρίζεται χιλιάδες ταυτόχρονες συνδέσεις με ελάχιστη κατανάλωση πόρων. Ως διακομιστής αντίστροφης μεσολάβησης, το NGINX δρομολογεί τα εισερχόμενα αιτήματα HTTP/HTTPS προς έναν ή περισσότερους διακομιστές (π.χ., εφαρμογές σε Node.js, Django ή containers όπως στην παρούσα υλοποίηση), διασφαλίζοντας την απομόνωση των ευαίσθητων υπηρεσιών από το δημόσιο Διαδίκτυο.

Η διαμόρφωση γίνεται μέσω αρχείων ρυθμίσεων (nginx.conf), τα οποία ορίζουν στο εσωτερικό τους “server blocks” (παράδειγμα server block παρατίθεται στην εικόνα 7.13), κανόνες δρομολόγησης και πολιτικές ασφαλείας. Η ενεργοποίηση του πρωτοκόλλου HTTPS στον NGINX απαιτεί δημιουργία και εγκατάσταση ψηφιακών πιστοποιητικών

SSL/TLS, τα οποία εξασφαλίζουν την ταυτοποίηση του διακομιστή και την κρυπτογράφηση των συνδέσεων. Για πειραματικούς σκοπούς ή ανάπτυξη, μπορούν να χρησιμοποιηθούν self-signed πιστοποιητικά, τα οποία είναι δυνατόν να δημιουργηθούν με το εργαλείο OpenSSL. Ωστόσο, για παραγωγικά περιβάλλοντα, συνιστάται η χρήση πιστοποιητικών από αξιόπιστες αρχές έκδοσης (Certificate Authorities - CAs), όπως η Let's Encrypt, η οποία προσφέρει δωρεάν πιστοποιητικά μέσω του πρωτοκόλλου ACME.

```
server {

        listen 1027 ssl;

        server_name orion.tls.openziti.eu;
        ssl_certificate "/etc/letsencrypt/live/tls.openziti.eu/fullchain.pem";
        ssl_certificate_key "/etc/letsencrypt/live/tls.openziti.eu/privkey.pem";
        location / {

                proxy_pass https://orion.openziti.eu:535;

        }
}
```

Εικόνα 7.13: Απόσπασμα από αρχείο ρυθμίσεων διακομιστή αντίστροφης μεσολάβησης NGINX

Σε αρχεία ρυθμίσεων (π.χ., /etc/nginx/sites-available/example.com), μπορεί να προστεθεί νέο “server block” το οποίο να ορίζει κάποια θύρα στην οποία δύναται να δέχεται αιτήματα ο διακομιστής και ορίζει τις παραμέτρους TLS. Βασικές παράμετροι περιλαμβάνουν τη διαδρομή προς το αρχείο που περιέχει το ψηφιακό πιστοποιητικό (ssl_certificate),το ιδιωτικό κλειδί (ssl_certificate_key), και δυνητικά τις προτιμώμενες μεθόδους κρυπτογράφησης (π.χ., ssl_ciphers).

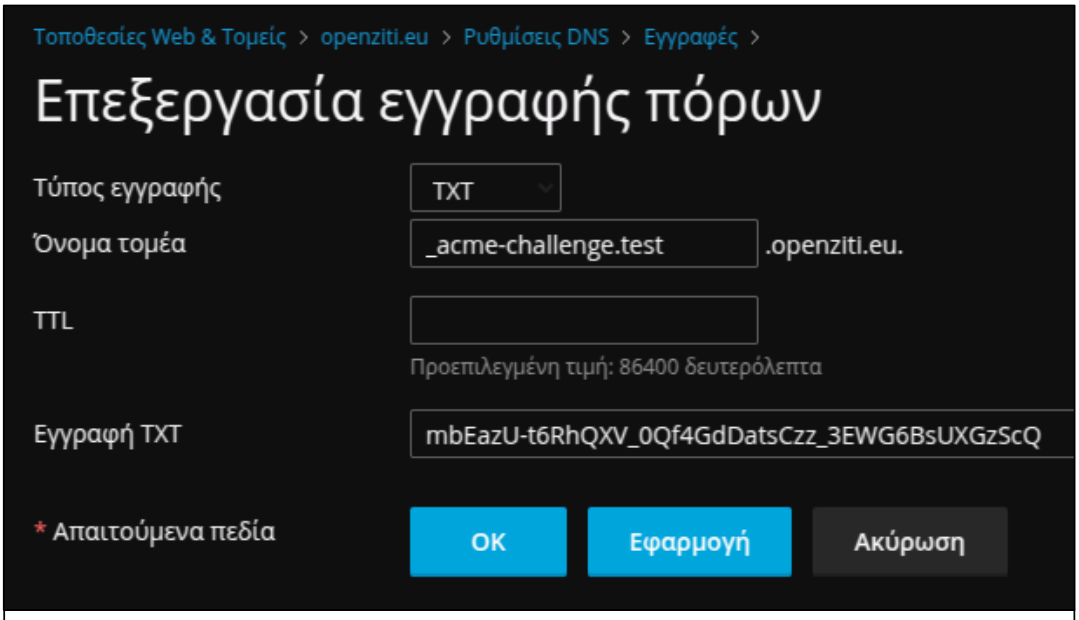
```
odysseas@odysseas-IdeaPad-3-15ABA7:~$ wildcard_url=test.openziti.eu
odysseas@odysseas-IdeaPad-3-15ABA7:~$ sudo docker run -it --rm --name certbot   -v "/etc/letsencrypt:/etc/letsencrypt"   -v "/var/lib/letsencrypt:/var/lib/letsencrypt"   certbot/certbot cert
only -d "*.${wildcard_url}"                  --manual                  --preferred-challenges dns                  --email "${your_email}"                  --agree-tos
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for *.test.openziti.eu

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please deploy a DNS TXT record under the name:

_acme-challenge.test.openziti.eu.

with the following value:

mbEazU-t6RhQXV_0Qf4GdDatsCzz_3EWG6BsUXGzScQ

Before continuing, verify the TXT record has been deployed. Depending on the DNS
provider, this may take some time, from a few seconds to multiple minutes. You can
check if it has finished deploying with aid of online tools, such as the Google
Admin Toolbox: https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.test.openziti.eu.
Look for one or more bolded line(s) below the line ';ANSWER'. It should show the
value(s) you've just added.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Press Enter to Continue

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/test.openziti.eu/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/test.openziti.eu/privkey.pem
This certificate expires on 2025-05-09.
These files will be updated when the certificate renews.

NEXT STEPS:
- This certificate will not be renewed automatically. Autorenewal of --manual certificates requires the use of an authentication hook script (--manual-auth-hook) but one was not provided. To
 renew this certificate, repeat this same certbot command before the certificate's expiry date.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
odysseas@odysseas-IdeaPad-3-15ABA7:~$
```

Εικόνα 7.14: Δημιουργία ψηφιακού πιστοποιητικού με χρήση του εργαλείου “certbot” και της μεθόδου “DNS challenge”

Η διαμόρφωση ενός ασφαλούς διακομιστή αντίστροφης μεσολάβησης συνεπάγεται ορισμένες προκλήσεις, κυρίως σχετικές με τη διαχείριση πιστοποιητικών και την ενημέρωση ρυθμίσεων. Η αυτοματοποίηση της ανανέωσης πιστοποιητικών (π.χ., με το

εργαλείο Certbot για πιστοποιητικά που δημιουργούνται μέσω της Let's Encrypt) αποτελεί απαραίτητη πρακτική για την αποφυγή διακοπών λειτουργίας.

Ο διακομιστής αντίστροφης μεσολάβησης που χρησιμοποιήθηκε στο δίκτυο που δεν υιοθετεί Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης είχε στη διάθεση του ψηφιακό πιστοποιητικό που δημιουργήθηκε μέσω του εργαλείου Certbot, όπως φαίνεται στην εικόνα 7.14. Η διαδικασία δημιουργίας ψηφιακού πιστοποιητικού με τη μέθοδο “DNS challenge” απαιτεί τη δημιουργία εγγραφής DNS με συγκεκριμένο περιεχόμενο όπως ζητείται από το εργαλείο Certbot, με σκοπό την απόδειξη ελέγχου του αντίστοιχου domain. Η δημιουργία τέτοιων εγγραφών DNS μπορεί να πραγματοποιηθεί μέσω των διεπαφών διαχείρισης που προσφέρει ο εκάστοτε πάροχος domain names, όπως π.χ. εικόνα 7.15 μέσω γραφικού περιβάλλοντος.



Εικόνα 7.15: Δημιουργία εγγραφής DNS για την ολοκλήρωση της διαδικασίας δημιουργίας ψηφιακού πιστοποιητικού μέσω της μεθόδου “DNS challenge”

Επιπλέον, σχετικά με την υποστήριξη TLS από διακομιστές αντίστροφης μεσολάβησης, αξίζει να σημειωθεί ότι η τακτική ενημέρωση των αλγορίθμων κρυπτογράφησης και η απενεργοποίηση λιγότερο ασφαλών πρωτοκόλλων (π.χ., TLS 1.0/1.1) μπορούν να ενισχύσουν την ανθεκτικότητα έναντι γνωστών ευπαθειών (π.χ., POODLE, BEAST). Τέλος, με τη χρήση εργαλείων όπως το SSL Labs Server Test μπορεί να ελεγχθεί το κατά πόσον ότι οι ρυθμίσεις πληρούν τα σύγχρονα πρότυπα ασφαλείας.

Η χρήση του NGINX ως ασφαλούς διακομιστή αντίστροφης μεσολάβησης αποτελεί θεμελιώδη στρατηγική για την προστασία και βελτιστοποίηση σύγχρονων δικτυακών εφαρμογών. Ωστόσο, η επιτυχής υλοποίηση απαιτεί συνεχή ενημέρωση, αυστηρή εφαρμογή πολιτικών ασφαλείας και ενσωμάτωση με συστήματα αυτοματοποιημένης διαχείρισης. Μελλοντικές τάσεις ενδέχεται να εστιάσουν στην ενσωμάτωση πρωτοκόλλων όπως το HTTP/3 και την υποστήριξη αλγορίθμων κρυπτογράφησης ανθεκτικών έναντι επιθέσεων από κβαντικούς υπολογιστές, ώστε να διασφαλιστεί η μακροπρόθεσμη ακεραιότητα των συστημάτων.

7.1.5 Ανάπτυξη της εφαρμογής FIWARE Orion Context Broker ως διακομιστή

Η ανάπτυξη της εφαρμογής FIWARE Orion Context Broker ως διακομιστή πραγματοποιήθηκε με ελαφρώς διαφορετικό τρόπο στις δύο περιπτώσεις δικτύων για τις οποίες διεξήχθησαν μετρήσεις καθυστέρησης μετ' επιστροφής (Round-Trip). Και στις δύο περιπτώσεις η εφαρμογή Orion Context Broker αναπτύχθηκε ως docker-compose container, αλλά με μικρές διαφορές στα αρχεία "docker-compose.yml" που καθορίζουν τις ακριβείς παραμέτρους ανάπτυξης του διακομιστή. Οι παραπάνω διαφορές οφείλονται στον τρόπο με τον οποίο εφαρμόζεται το πρωτόκολλο TLS σε κάθε μια από τις δύο περιπτώσεις (mutual/standard TLS).

```yaml
version: "3"

services:
  orion:
    image: fiware/orion
    ports:
      - "1026:1026"
    depends_on:
      - mongo
    volumes:
        - './privkey.pem:/privkey.pem'
        - './fullchain.pem:/fullchain.pem'
    command: -dbURI mongodb://mongo -https -cert /fullchain.pem -key /privkey.pem

  mongo:
    image: mongo:6.0
    command: --nojournal
```

Εικόνα 7.16: Αρχείο "docker-compose.yml" που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής FIWARE Orion Context Broker με υποστήριξη TLS

Στην περίπτωση που εφαρμόζεται του τυπικού (standard) TLS, η ανάπτυξη της εφαρμογής Orion Context Broker γίνεται με τρόπο τέτοιο ώστε να έχει τη δυνατότητα να δέχεται αιτήματα HTTPS και για τον σκοπο αυτό στην εφαρμογή πρέπει να παρέχονται ψηφιακό πιστοποιητικό και το αντίστοιχο ιδιωτικό κλειδί (όπως φαίνεται στην εικόνα 7.16). Στην περίπτωση δικτύου που υιοθετεί Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης, την αμφίδρομη εφαρμογή TLS, για ό,τι αφορά τον διακομιστή Orion Context Broker, αναλαμβάνει η εφαρμογή tunnel που έχει εγκατασταθεί στο αντίστοιχο εικονικό μηχάνημα, η οποία ακολούθως δημιουργεί απλές TCP συνδέσεις με τον ίδιο τον διακομιστή. Συνεπώς στην περίπτωση αυτή η εφαρμογή Orion Context Broker αναπτύχθηκε ως διακομιστής χωρίς να της παρέχεται ψηφιακό πιστοποιητικό (κατά τρόπο όμοιο με αυτό στην εικόνα 5.5) και διαμορφωμένη να δέχεται από την τοπικά ανεπτυγμένη εφαρμογή tunnel απλά αιτήματα HTTP χωρίς εφαρμογή TLS.

7.2 Σύγκριση μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip)

Στους πίνακες 7.1-7.5 περιλαμβάνονται στατιστικά στοιχεία (ελάχιστη, μέγιστη, μέση τιμή και τυπική απόκλιση) για ενδεικτικές μετρήσεις (τουλάχιστον 50 σε κάθε

περίπτωση) καθυστέρησης μετ' επιστροφής (Round-Trip) αιτημάτων HTTP με διαφορετικά μεγέθη μεταφερόμενων δεδομένων (payloads). Παράλληλα, στα διαγράμματα 7.1-7.3 απεικονίζονται και γραφικές παραστάσεις σύγκρισης των μέσων τιμών καθυστέρησης μετ' επιστροφής (Round-Trip) των αιτημάτων HTTP, για τις περιπτώσεις τοπολογιών δικτύου που είτε υιοθετούν είτε όχι Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης, και για διαφορετικά μεγέθη μεταφερόμενων δεδομένων (payload). Τα μεγέθη των μεταφερόμενων δεδομένων (payload) των αιτημάτων HTTP ταξινομήθηκαν ανάλογα με τον αριθμό των attributes που περιλάμβαναν οι οντότητες (entities) της εφαρμογής διαχείρισης βάσεων δεδομένων Orion Context Broker που ανακτώνταν σε κάθε σύνολο μετρήσεων.

TLS implementation		mutual	standard
Round-Trip-Time (seconds)	minimum	0.219869	0.231558
	maximum	0.309788	0.341096
	average	0.293491	0.270289
	std dev	0.021661	0.026291
Average Download Speed(B/s)	minimum	1207	1096
	maximum	1701	1615
	average	1281	1395
	std dev	109	129

Πίνακας 7.1: Σύγκριση στατιστικών δεδομένων 50 μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip) για απλό healthcheck της εφαρμογής FIWARE Orion Context Broker

Αξίζει να σημειωθεί ότι η σύγκριση των παραπάνω μετρήσεων ουσιαστικά αποτελεί και σύγκριση ανάμεσα στην εκδοχή του πρωτοκόλλου TLS που χρησιμοποιήθηκε σε κάθε περίπτωση, δηλαδή, της τυπικής εκδοχής του πρωτοκόλλου TLS, που αποτελεί αυτή τη στιγμή την πιο ευρέως χρησιμοποιούμενη υλοποίηση, και της εκδοχής αμφίδρομης εφαρμογής του πρωτοκόλλου (mTLS), που χρησιμοποιούν τα δίκτυα Μηδενικής Εμπιστοσύνης OpenZiti.

TLS implementation		mutual	standard
Round-Trip-Time (seconds)	minimum	0.216762	0.232731
	maximum	0.294865	0.340009
	average	0.286881	0.271600
	std dev	0.014862	0.024610
Average Download Speed(B/s)	minimum	474	411
	maximum	645	601
	average	489	519
	std dev	30	45

Πίνακας 7.2: Σύγκριση στατιστικών δεδομένων 50 μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip) για ανάκτηση οντοτήτων(entities) με 2 attributes

TLS implementation		mutual	standard
Round-Trip-Time (seconds)	minimum	0.217339	0.237094
	maximum	0.503044	0.319197
	average	0.294525	0.271104
	std dev	0.032600	0.023386
Average Download Speed(B/s)	minimum	1248	1967
	maximum	2889	2648
	average	2150	2332
	std dev	176	196

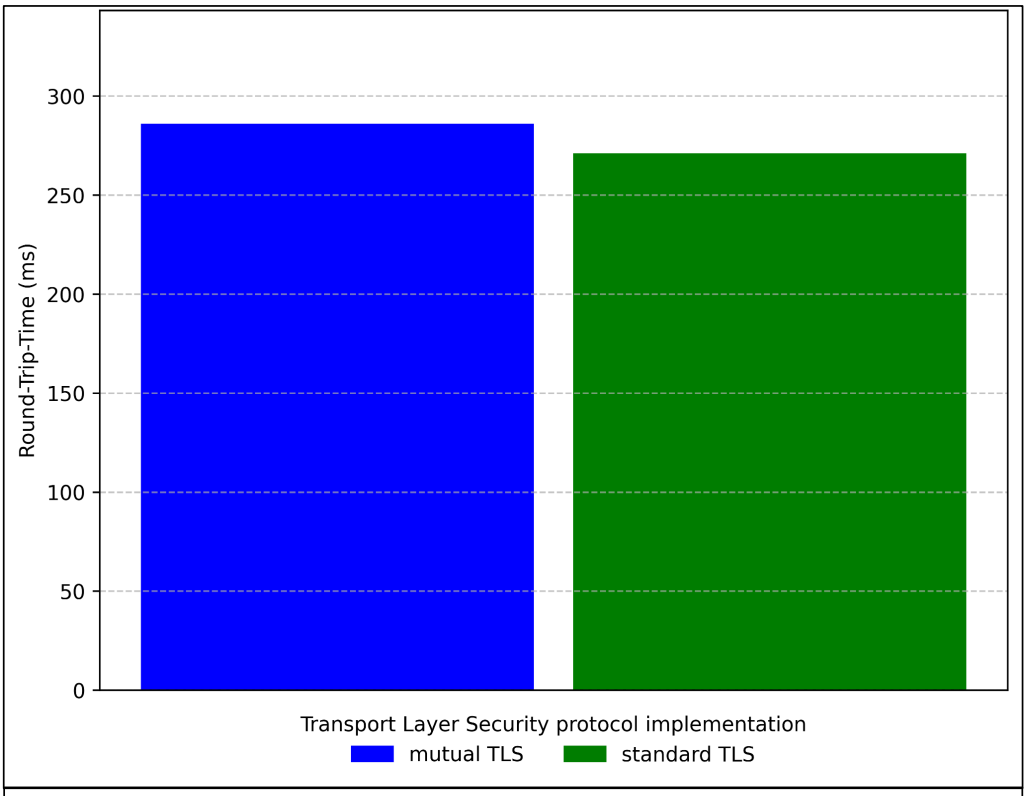
Πίνακας 7.3: Σύγκριση στατιστικών δεδομένων 50 μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip) για ανάκτηση οντοτήτων(entities) με 10 attributes

TLS implementation		mutual	standard
Round-Trip-Time (seconds)	minimum	0.220780	0.235005
	maximum	0.506577	0.348114
	average	0.294501	0.269052
	std dev	0.033445	0.028874
Average Download Speed(B/s)	minimum	12347	17968
	maximum	28331	26616
	average	21433	23499
	std dev	1822	2357

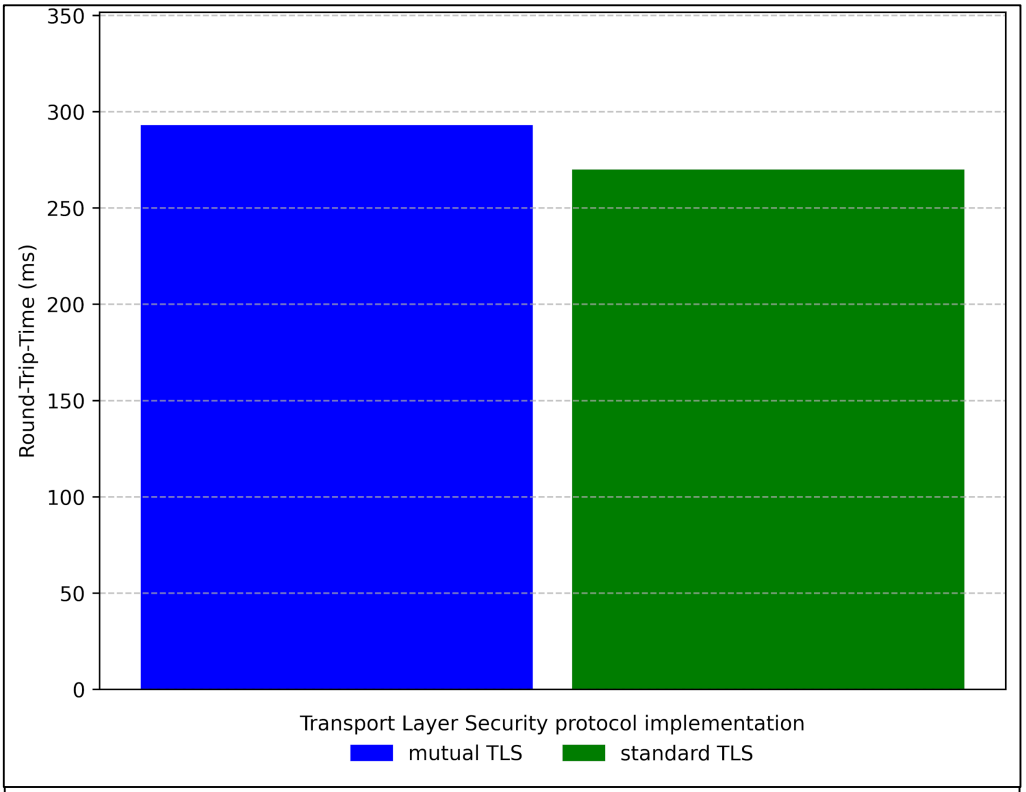
Πίνακας 7.4: Σύγκριση στατιστικών δεδομένων 50 μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip) για ανάκτηση οντοτήτων(entities) με 100 attributes

TLS implementation		mutual	standard
Round-Trip-Time (seconds)	minimum	1.225162	1.110194
	maximum	1.382839	1.373189
	average	1.288393	1.251216
	std dev	0.034381	0.068977
Average Download Speed(B/s)	minimum	468150	471440
	maximum	528401	583120
	average	502816	518986
	std dev	13080	28850

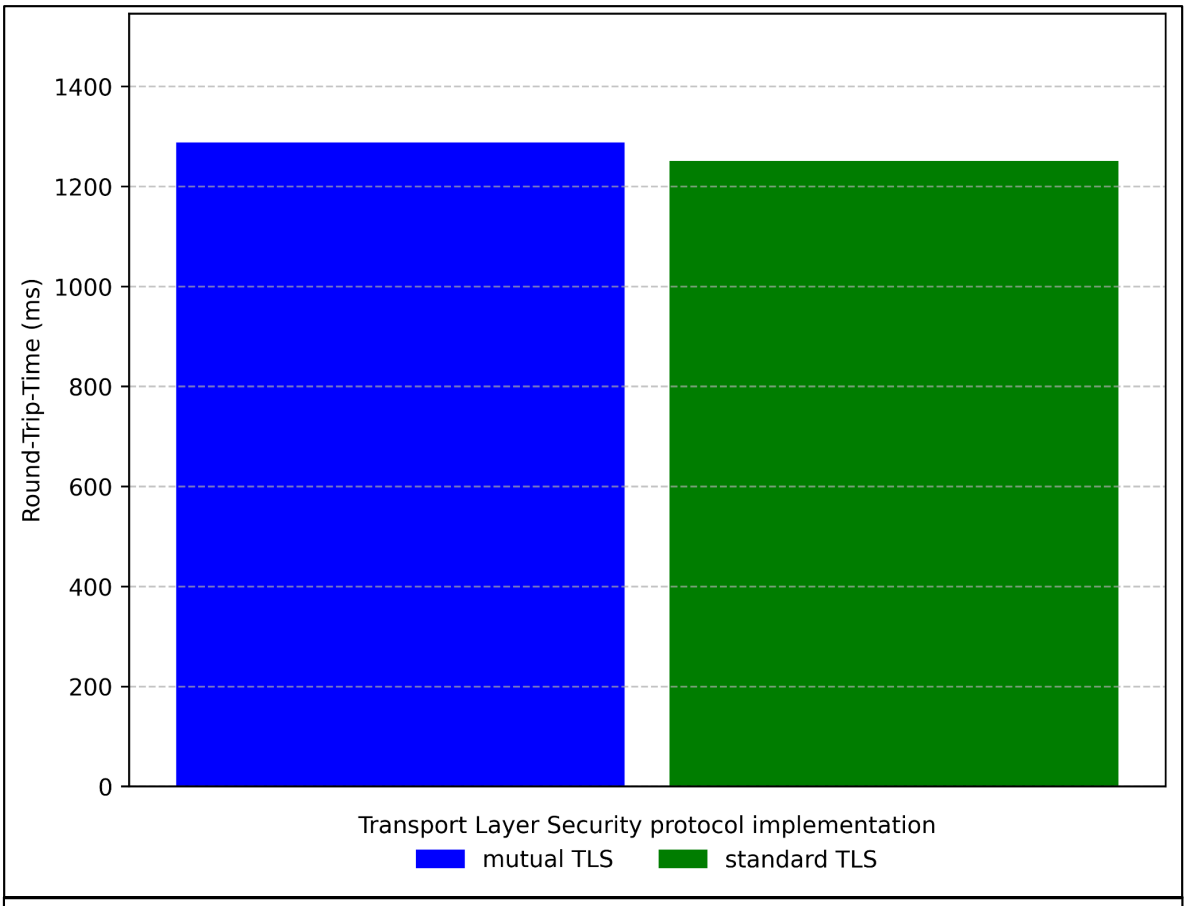
Πίνακας 7.5: Σύγκριση στατιστικών δεδομένων 50 μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip) για ανάκτηση οντοτήτων(entities) με 10^4 attributes



Διάγραμμα 7.1: Γραφική παράσταση σύγκρισης μέσης καθυστέρησης μετ' επιστροφής (Round-Trip) για απλό healthcheck της εφαρμογής FIWARE Orion Context Broker



Διάγραμμα 7.2: Γραφική παράσταση σύγκρισης μέσης καθυστέρησης μετ' επιστροφής (Round-Trip) για ανάκτηση οντοτήτων(entities) με 2 attributes



Διάγραμμα 7.3: Γραφική παράσταση σύγκρισης μέσης καθυστέρησης μετ' επιστροφής (Round-Trip) για ανάκτηση οντοτήτων(entities) με 10^4 attributes

7.3 Συμπεράσματα από τη μελέτη των μετρήσεων καθυστέρησης μετ' επιστροφής (Round-Trip)

Εξετάζοντας τα αποτελέσματα των μετρήσεων που απεικονίζονται στους πίνακες και τις γραφικές παραστάσεις της προηγούμενης ενότητας διαπιστώνουμε αρκετά μικρή αύξηση των τιμών της μέσης καθυστέρησης μετ' επιστροφής (Round-Trip) κατά μέσο όρο 24 χιλιοστά του δευτερολέπτου για τα διάφορα μεγέθη μεταφερόμενων δεδομένων (payload), που αντιστοιχεί σε ποσοστιαία αύξηση κατά μέσο όρο 6%. Η αύξηση αυτή οφείλεται στην αμφίδρομη χρήση του πρωτοκόλλου TLS και τη συνακόλουθη ταυτοποίηση του πελάτη (client) που λαμβάνει χώρα στις περιπτώσεις που εφαρμόζεται Αρχιτεκτονική Ασφάλειας Μηδενικής Εμπιστοσύνης.

Παράλληλα οι συνδέσεις ανάμεσα στους δρομολογητές και τις εφαρμογές tunnel του δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti, με βάση τις μετρήσεις για τη μέση ταχύτητα λήψης δεδομένων, φαίνονται να είναι ελαφρώς πιο αργές σε σύγκριση με τις αντίστοιχες συνδέσεις των εφαρμογών πελάτη και διακομιστή με τον διακομιστή αντίστροφης μεσολάβησης, γεγονός που επίσης επηρεάζει τη συνολική καθυστέρηση μετ' επιστροφής (Round-Trip). Το ελαφρώς μειωμένο αυτό εύρος ζώνης των συνδέσεων ανάμεσα στις δύο περιπτώσεις τοπολογιών που χρησιμοποιήθηκαν, αποτελεί ιδιαίτερο χαρακτηριστικό της συγκεκριμένης υλοποίησης των δικτύων Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης OpenZiti που θα μπορούσε να εξαλειφθεί με κατάλληλες

προσαρμογές-βελτιστοποιήσεις της λειτουργίας των στοιχείων του δικτύου ή να μην συναντάται σε άλλες υλοποιήσεις Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης.

Επίσης τμήμα της καθυστέρησης οφείλεται και στις εσωτερικές αλληλεπιδράσεις ανάμεσα στα στοιχεία λογισμικού των δικτύων OpenZiti (όπως αυτές περιγράφονται στην ενότητα 5.4 της παρούσας Διπλωματικής Εργασίας) που απαιτούνται ώστε να επιτευχθεί τελικά πρόσβαση σε υπηρεσίες μέσω των δικτύων Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης OpenZiti. Παρ' όλα αυτά η καθυστέρηση μετ' επιστροφής (Round-Trip) φαίνεται ακόμα και με την υιοθέτηση Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης να είναι εντός των αποδεκτών ορίων της βιομηχανίας των δικτυακών υπηρεσιών.

Στα αποτελέσματα της ενότητας 7.2 παρατηρείται ότι κατά την αύξηση του ωφέλιμου μεταφερόμενου φορτίου (payload) ανά απόκριση (response) σε αιτήματα HTTP, αυξάνεται και η μέση ταχύτητα λήψης δεδομένων. Η παραπάνω τάση που παρατηρείται, οφείλεται στο γεγονός ότι κατά την ανάκτηση οντοτήτων (entities) της εφαρμογής διαχείρισης βάσεων δεδομένων Orion Context Broker με μεγαλύτερο μέγεθος, το ωφέλιμο φορτίο των αντίστοιχων τμημάτων (segments) TCP, πακέτων IP και πλαισίων ethernet τα οποία ενθυλακώνουν στο ωφέλιμο φορτίο (payload) τους τις αποκρίσεις (responses) στα αιτήματα HTTP, έχουν αντίστοιχα μεγαλύτερη αναλογία ωφέλιμου φορτίου προς το σύνολο των μεταφερόμενων δεδομένων (το οποίο περιλαμβάνει επιπλέον επικεφαλίδες του κάθε στρώματος, checksums-CRC κλπ., των οποίων το μέγεθος διατηρείται σε μεγάλο βαθμό σταθερό, ανεξαρτήτως των διακυμάνσεων του ωφέλιμου φορτίου).

Μελετώντας τον τρόπο λειτουργίας των δικτύων Μηδενικής Εμπιστοσύνης ευρύτερα, αλλά και τους μηχανισμούς λειτουργίας των δικτύων OpenZiti ειδικότερα, αναπόφευκτα καταλήγουμε στην παρατήρηση ότι σε κάποιο σημείο κατά τη διάρκεια χρήσης/πρόσβασης σε ένα δίκτυο που υιοθετεί Αρχιτεκτονική Μηδενικής Εμπιστοσύνης απαιτείται κάποιου είδους ταυτοποίηση του χρήστη. Η ταυτοποίηση αυτή αναμφίβολα θα επιφέρει κάποιο κόστος αναφορικά με τον συνολικό χρόνο απόκρισης των υπηρεσιών. Με βάση τα παραπάνω μπορούμε να συμπεράνουμε ότι η Αρχιτεκτονική Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης πράγματι επηρεάζει σε μικρό βαθμό τις επιδόσεις των εφαρμογών.

Οι μέθοδοι που επιλέγονται για την εν λόγω ταυτοποίηση, όπως και η συχνότητα με την οποία διεξάγονται αποτελούν αποφασιστικούς παράγοντες αναφορικά με το αποτύπωμα της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης στις επιδόσεις των εφαρμογών. Για παράδειγμα η επιμήκυνση της διάρκειας εγκυρότητας των συνεδριών (sessions) API ή των συνεδριών υπηρεσιών θα μπορούσαν να αποτελέσουν παράγοντες που θα συρρίκνωναν την επίδραση της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης στις επιδόσεις των εφαρμογών. Φυσικά τέτοιες προσεγγίσεις είναι καταφανές ότι θα αποτελούσαν συμβιβασμούς σε επίπεδο Ασφάλειας.

Λαμβάνοντας υπ' όψιν τους γενικότερους συμβιβασμούς ανάμεσα στην ασφάλεια και τις επιδόσεις έχει παρατηρηθεί ότι η υιοθέτηση χρήσης TOTP έχει προσθέσει έναν ισχυρό κρίκο ασφάλειας στην πρόσβαση σε δικτυακές υπηρεσίες. Ωστόσο η χρήση τους δεν ενσωματώνεται σε κάθε αίτημα (request) προς μια υπηρεσία αλλά συνήθως αποκλειστικά στη διαδικασία ταυτοποίησης για τη δημιουργία συνεδριών (sessions), αφού σε αντίθετη περίπτωση η χρήση τους θα ήταν μη πρακτική. Σε διαφορετικές υλοποιήσεις η ενσωμάτωση των διαπιστευτηρίων (credentials) ταυτοποίησης στα αιτήματα υπηρεσιών (service requests) θα μπορούσε να βελτιώσει τους χρόνους απόκρισης, αποτελώντας όμως συμβιβασμό σε επίπεδο ασφάλειας εφόσον θα προϋπέθετε δημοσίως εκτεθειμένα ακραία

σημεία (endpoints), γνωστά και σε μη ταυτοποιημένους χρήστες και εφαρμογές. Αντίστοιχα, για δικτυακές υπηρεσίες μεγάλης κλίμακας, η διαχείριση ψηφιακών πιστοποιητικών X.509 για τους χρήστες μπορεί να αποτελέσει για τους διαχειριστές αποτρεπτικό παράγοντα σχετικά με την αμφίδρομη εφαρμογή του πρωτοκόλλου TLS.

Με λίγα λόγια, η επίδραση της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης στις επιδόσεις των εκάστοτε δικτυακών εφαρμογών πρέπει να μελετάται υπό το πρίσμα των συγκεκριμένων απαιτήσεων κάθε σεναρίου χρήσης. Κατ' αυτόν τον τρόπο οι μηχανικοί και διαχειριστές δικτύων είναι σε θέση να λαμβάνουν τις κατάλληλες αποφάσεις και να ανευρίσκεται η χρυσή τομή του συμβιβασμού ανάμεσα στην ασφάλεια και τις επιδόσεις, για κάθε σύστημα, ανάλογα με τα ιδιαίτερα χαρακτηριστικά και τις απαιτήσεις του.

Διαφορετικές επιλογές αναφορικά με την Αρχιτεκτονική Ασφάλειας είναι λογικό να υιοθετούνται από τους μηχανικούς δικτύων σε περιπτώσεις που καλούνται να διαχειριστούν την πρόσβαση σε πολύ ευαίσθητα δεδομένα σε σύγκριση με περιπτώσεις όπου οι επιδόσεις είναι μακράν σημαντικότερες της ασφάλειας έως και ζωτικής σημασίας για ένα σύστημα. Παραδείγματος χάριν, διαφορετικές επιλογές σχετικά με την ασφάλεια και τους μηχανισμούς επίτευξης της απαιτούνται σε στρατιωτικές υποδομές, π.χ. σε κανάλια επικοινωνίας ελέγχου και διοίκησης ανάμεσα σε μάχιμες στρατιωτικές μονάδες, και διαφορετικές σε συστήματα έγκαιρης προειδοποίησης για φυσικές καταστροφές.

Η πρώτη περίπτωση υπόκειται σε ένα πλαίσιο αυστηρά απόρρητων πληροφοριών, διαρροή των οποίων μπορεί να έχει καταστροφικές συνέπειες, ενώ η δεύτερη περίπτωση αφορά πληροφορία η οποία αναπόφευκτα καθίσταται ελεύθερα προσβάσιμη στον οποιονδήποτε (η πραγματοποίηση μια φυσικής καταστροφής) και κατ' επέκταση το βάρος των αποφάσεων σχεδιασμού τείνει καταλυτικά προς το πεδίο των επιδόσεων.

Σε περιβάλλοντα Διαδικτύου των Πραγμάτων (IoT), όπου οι πόροι είναι συχνά περιορισμένοι, η εφαρμογή της αρχιτεκτονικής μηδενικής εμπιστοσύνης απαιτεί ιδιαίτερη προσοχή. Έρευνες έχουν δείξει ότι η χρήση πιο αποδοτικών κρυπτογραφικών αλγορίθμων και η βελτιστοποίηση των πρωτοκόλλων επικοινωνίας μπορούν να μειώσουν σημαντικά την επιβάρυνση στις επιδόσεις των εφαρμογών [53]. Παράλληλα άλλες μελέτες έχουν υπερτονίσει τη σημασία της προσεκτικής σχεδίασης και υλοποίησης των μηχανισμών ασφαλείας στο πλαίσιο της Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης. Έτσι έχει καταστεί φανερό πως η υιοθέτηση προσεγγίσεων βασισμένων στην ανάλυση κινδύνου, επιτρέπει εξισορρόπηση μεταξύ ασφάλειας και επιδόσεων [54].

Συμπερασματικά, αν και η επίδραση της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης στις επιδόσεις των εφαρμογών φαίνεται να μην είναι σημαντική, αποτελεί ένα πολύπλοκο ζήτημα που απαιτεί προσεκτική εξέταση και σχεδιασμό κατά περίπτωση. Ενώ η εισαγωγή πρόσθετων ελέγχων ασφαλείας μπορεί να επηρεάσει ελαφρώς τις επιδόσεις, η πράξη δείχνει ότι με κατάλληλες τεχνικές και βελτιστοποιήσεις, είναι δυνατόν να επιτευχθεί ένα υψηλό επίπεδο ασφάλειας χωρίς σημαντικές επιπτώσεις στη λειτουργικότητα των εφαρμογών. Η συνεχής έρευνα και ανάπτυξη στον τομέα αυτό αναμένεται να οδηγήσει σε ακόμη πιο αποτελεσματικές λύσεις στο μέλλον, επιτρέποντας την ευρύτερη υιοθέτηση της Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης σε ποικίλα περιβάλλοντα και εφαρμογές.

8. Συμπεράσματα και πιθανές ερευνητικές προεκτάσεις

8.1 Συμπεράσματα

Η παρούσα Διπλωματική Εργασία εξέτασε την εφαρμογή της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης (ZTNA) μέσω της πλατφόρμας ανοικτού κώδικα OpenZiti, με στόχο την ανάδειξη των δυνατοτήτων, των προκλήσεων που παρουσιάζονται στην υιοθέτηση της και των επιδόσεων αυτής της προσέγγισης Ασφάλειας Δικτύων. Η εφαρμογή Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης μέσω της πλατφόρμας ανοικτού κώδικα OpenZiti, εξετάστηκε τόσο από τη σκοπιά της πρακτικής υλοποίησης της όσο και από τη σκοπιά της επίδρασης της Αρχιτεκτονικής στις επιδόσεις των εφαρμογών. Τα κύρια συμπεράσματα από την παρούσα Διπλωματική Εργασία βασίζονται σε:

1. Υλοποίηση Εφαρμογής γραμμής εντολών: Η εφαρμογή που αναπτύχθηκε αξιοποιεί τη βιβλιοθήκη libcurl() και το OpenZiti SDK-C για αποστολή αιτημάτων HTTP μέσω δικτύου Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης, επιδεικνύοντας τη δυνατότητα ενσωμάτωσης μηχανισμών όπως η συνεχής ταυτοποίηση, η κατάτμηση πόρων και η πρόσβαση ελαχίστων προνομίων. Κατ’ αυτόν τον τρόπο κατέστη φανερή η δυνατότητα εφαρμογής της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης συνολικά, αλλά και ασφαλών πρωτοκόλλων (π.χ. mTLS), σε υπάρχοντα δικτυακά συστήματα.
2. Συγκριτική Μελέτη Επιδόσεων: Πραγματοποιήθηκαν μετρήσεις καθυστέρησης μετ' επιστροφής (Round-Trip) αιτήματων HTTP προς τον FIWARE Orion Context Broker, καταδεικνύοντας ότι η υιοθέτηση της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης εισάγει ελαφρώς μετρήσιμη αύξηση στην καθυστέρηση (κυρίως λόγω κρυπτογράφησης και ταυτοποίησης), ωστόσο μέσα σε διαχειρίσιμα πλαίσια για κρίσιμες εφαρμογές. Τα παραπάνω ευρήματα δείχνουν ότι η ασφάλεια συχνά επιβάλλει συμβιβασμούς σε επίπεδο επιδόσεων, ιδιαίτερα σε περιβάλλοντα όπου εφαρμόζεται ισχυρή κρυπτογράφηση. Πιο συγκεκριμένα, η χρήση κρυπτογραφικών πρωτοκόλλων (π.χ., ECDH, AES-256) αυξάνει το υπολογιστικό κόστος, που αποτελεί παράμετρο με αυξημένη σημασία ιδιαίτερα σε συσκευές με περιορισμένους πόρους, όπως αυτές του Διαδικτύου των Πραγμάτων (IoT).

Η ευελιξία της πλατφόρμας OpenZiti στην υποστήριξη πολλαπλών πρωτοκόλλων (π.χ., UDP, TLS 1.3) και η δυνατότητα δυναμικής ρύθμισης πολιτικών προσφέρουν ένα ισχυρό πλαίσιο για επίτευξη ισορροπίας ανάμεσα στην ασφάλεια και τις επιδόσεις των εφαρμογών.

8.2 Μελλοντικές Ερευνητικές Κατευθύνσεις

Η παρούσα Διπλωματική Εργασία έδειξε ότι η κρυπτογράφηση (ιδίως η αμφίδρομη εφαρμογή TLS) προκαλεί διαχειρίσιμη αύξηση της καθυστέρησης μετ' επιστροφής

(Round-Trip). Μελλοντικές έρευνες θα μπορούσαν να εστιάσουν στην υιοθέτηση κβαντικά ανθεκτικών αλγορίθμων (π.χ., Lattice-based κρυπτογράφηση) ή υβριδικών προτύπων που συνδυάζουν παραδοσιακά και μετα-κβαντικά πρωτόκολλα, όπως έχει προταθεί από τους Bernstein και Lange (2017) [15]. Επιπλέον, η εξέταση ελαφρύτερων πρωτοκόλλων (π.χ. NOISE Protocol Framework) θα μπορούσε να μειώσει την υπολογιστική επιβάρυνση σε συσκευές IoT, όπως παρατηρεί και σχετική μελέτη του NIST (2023) σε πρόσφατες οδηγίες για τη βέλτιστη εφαρμογή Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης [55].

Η συνεχής ταυτοποίηση και η δυναμική προσαρμογή πολιτικών ασφαλείας θα μπορούσαν να ενισχυθούν από αλγορίθμους μηχανικής μάθησης. Για παράδειγμα, συστήματα βασισμένα σε ενισχυτική μάθηση θα μπορούσαν να βελτιστοποιήσουν τις διαδικασίες εξουσιοδότησης πρόσβασης σε πραγματικό χρόνο, να εξισορροπήσουν την ασφάλεια ως προς τις επιδόσεις και να μειώσουν τις εσφαλμένες ειδοποιήσεις για δικτυακές εισβολές. Για παράδειγμα, θα μπορούσαν να υλοποιηθούν νευρωνικά δίκτυα που να προσαρμόζουν το μέγεθος κλειδιού και τους αλγορίθμους κρυπτογράφησης βάσει του φόρτου δικτυακής κίνησης.

Παράλληλα, Generative Adversarial Networks (GANs) θα μπορούσαν να αποτελέσουν κρίσιμο εργαλείο για την προσομοίωση επιθέσεων και να εκπαιδεύσουν συστήματα Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης. Παρόμοιες προσεγγίσεις θα μπορούσαν να ενσωματωθούν στην πλατφόρμα OpenZiti για αυτόματη επιλογή πρωτοκόλλων ανάλογα με τις απαιτήσεις σχετικά με τους χρόνους καθυστέρησης. Επιπλέον, η χρήση Explainable AI (XAI) θα μπορούσε να διευκολύνει τους σχεδιαστές και διαχειριστές δικτύων σε σχέση με τη διαδικασία λήψης αποφάσεων αναφορικά με την ασφάλεια, ιδιαίτερα σε πολυσχιδείς, μεγάλης κλίμακας δικτυακές υποδομές.

Η εφαρμογή της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης σε βιομηχανικά περιβάλλοντα (Industry 4.0) και δίκτυα αισθητήρων (WSN) διαμορφώνει ένα απαιτητικό πεδίο έρευνας λόγω των περιορισμών που υφίστανται στους υπολογιστικούς και δικτυακούς πόρους. Η ασφάλεια των επικοινωνιών σε βιομηχανικά συστήματα Διαδικτύου των Πραγμάτων (IoT) φαίνεται ότι θα θέσει στο προσκήνιο την ανάγκη για χρήση βιβλιοθηκών κρυπτογράφησης με καλύτερες επιδόσεις και διαχειρίσιμους μηχανισμούς ταυτοποίησης (π.χ. TOTP με δυναμικά κλειδιά).

Μια πιθανή εξέλιξη θα μπορούσε να είναι η δημιουργία υλοποιήσεων παρόμοιων με την πλατφόρμα OpenZiti που να υποστηρίζουν συσκευές Διαδικτύου των Πραγμάτων (IoT) μέσω βιβλιοθηκών όπως το TinyDTLS, ελαχιστοποιώντας την κατανάλωση ενέργειας χωρίς να θυσιάζεται η ασφάλεια των επικοινωνιών. Επιπλέον, η ενσωμάτωση πρωτόκολλων όπως το CoAP (Constrained Application Protocol) στη λειτουργία ολοκληρωμένων λύσεων Δικτύωσης Μηδενικής Εμπιστοσύνης θα μπορούσε να διερευνηθεί στην κατεύθυνση βελτιστοποίησης των επικοινωνιών σε συσκευές χαμηλής κατανάλωσης ενέργειας και με περιορισμένους υπολογιστικούς πόρους.

8.2.1 Κλιμακωσιμότητα και Ενσωμάτωση Κβαντικών Τεχνολογιών

Η παρούσα Διπλωματική Εργασία αξιολόγησε τις επιδόσεις εφαρμογών σε τοπολογίες δικτύων σχετικά μικρής κλίμακας, λόγω των περιορισμένων διαθέσιμων πόρων. Ωστόσο, η κλιμάκωση σε δίκτυα με χιλιάδες κόμβους (π.χ. cloud infrastructures) σίγουρα

αποτελεί εύφορο πεδίο για περαιτέρω έρευνα. Αποκεντρωμένα συστήματα ταυτοποίησης βασισμένα σε blockchain θα μπορούσαν να μειώσουν την εξάρτηση από κεντρικές Αρχές Πιστοποίησης (CAs), προσφέροντας διαφάνεια και βελτιώνοντας τη συνολική ανθεκτικότητα των συστημάτων ελέγχου πρόσβασης, έναντι επιθέσεων.

Παράλληλα, η χρήση Service Meshes (π.χ., Istio) σε συνδυασμό με την εφαρμογή των Αρχών της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης, θα μπορούσε ενδεχομένως να αποτελέσει βήμα προς τη βελτιστοποίηση της διαχείρισης μικροϋπηρεσιών. Μια πιθανή υλοποίηση θα μπορούσε να ενσωματώσει πρωτόκολλα όπως το Decentralized Identifiers (DIDs), σε υλοποιήσεις Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης, επιτρέποντας την αυτόνομη διαχείριση ταυτοτήτων χωρίς την ανάγκη για μια κεντρική αρχή διαχείρισης ελέγχου πρόσβασης.

Οι συνεχώς αναπτυσσόμενες υλοποιήσεις κβαντικών υπολογιστών φαίνεται ότι θα θέσουν εν αμφιβόλω την προστασία που σήμερα παρέχουν τα κλασσικά κρυπτογραφικά συστήματα και θα καταστήσουν επιτακτική την ανάγκη αναβάθμισης των δικτυακών στρατηγικών ασφάλειας. Ερευνητικές πρωτοβουλίες όπως το NIST Post-Quantum Cryptography Standardization Project (2024) έχουν εντοπίσει αλγορίθμους (π.χ. Kyber, Dilithium) που φαίνεται ότι μπορούν να αποτελέσουν ακρογωνιαίο λίθο της παραπάνω μετάβασης [56]. Επιπλέον, η χρήση Quantum Key Distribution (QKD) για την ασφάλεια των συνδέσεων μεταξύ δρομολογητών θα μπορούσε να αποτελέσει καινοτόμο πεδίο έρευνας.

Παράλληλα, αξίζει να σημειωθεί ότι η εφαρμογή της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης δεν αφορά μόνο τεχνικές λεπτομέρειες, αλλά και ευρύτερες κοινωνικοοικονομικές διαστάσεις. Όπως υπογραμμίζει το ENISA (2023), η προστασία κρίσιμων υποδομών (π.χ., ενεργειακά δίκτυα, υγειονομικές υπηρεσίες) απαιτεί ολιστική υιοθέτηση των Αρχών του μοντέλου Μηδενικής Εμπιστοσύνης από το σύνολο των δρώντων στα πλαίσια των παραπάνω συστημάτων [57]. Η παρούσα εργασία φιλοδοξεί να συμβάλλει σε αυτό το πλαίσιο, προσφέροντας πρακτικά εργαλεία και μετρήσεις που να μπορούν δυνητικά να αξιοποιηθούν από οργανισμούς κατά τη μετάβαση από τα παραδοσιακά μοντέλα ασφάλειας.

8.3 Πρακτικές Επιπτώσεις και Παρατηρήσεις για Υλοποίηση

Με βάση τη μελέτη στα πλαίσια της παρούσας Διπλωματικής Εργασίας φαίνεται ότι η ενσωμάτωση της Αρχιτεκτονικής Ασφάλειας Δικτύων Μηδενικής Εμπιστοσύνης σε υπάρχοντα συστήματα απαιτεί:

1. Σταδιακή Υιοθέτηση των αρχών της Μηδενικής Εμπιστοσύνης: Ενσωμάτωση σε κρίσιμα συστήματα πρώτα (π.χ., διαχείριση ταυτοτήτων) πριν την επέκταση σε όλες τις λειτουργίες των δικτυακών υποδομών.
2. Εκπαίδευση Προσωπικού: Έμφαση στη συνεχή κατάρτιση στη χρήση νέων τεχνολογιών (π.χ., Biometric Authentication) και τη βαθύτερη κατανόηση των πολιτικών ασφαλείας.
3. Συμμόρφωση με σύγχρονα πρότυπα Ασφάλειας: Συμμόρφωση με κατευθυντήριες γραμμές όπως αυτές που περιγράφονται σε σημαντικές δημοσιεύσεις (π.χ. NIST SP

800-207) για την Αρχιτεκτονική Μηδενικής Εμπιστοσύνης, προς αποφυγή φαινομένων ασυμβατότητας, ώστε να διευκολυνθεί η ευρύτερη ενσωμάτωση του μοντέλου [1].

Συνοψίζοντας, η παρούσα εργασία επιβεβαίωσε τη βιωσιμότητα της υιοθέτησης Αρχιτεκτονικής Ασφάλειας Μηδενικής Εμπιστοσύνης ως πρότυπο ασφάλειας για τα σύγχρονα δίκτυα, ενώ ταυτόχρονα επεδίωξε να σκιαγραφήσει κατευθύνσεις για περαιτέρω έρευνα. Με συνδυασμό του θεωρητικού υπόβαθρου, πρακτικής υλοποίησης και συστηματικής αξιολόγησης επιχειρήθηκε να διαμορφωθεί ένα όσο το δυνατόν πιο ολοκληρωμένο πλαίσιο ως βάση για μελλοντικές ερευνητικές προσπάθειες. Η συνεχής ενσωμάτωση της Αρχιτεκτονικής Μηδενικής Εμπιστοσύνης στο φάσμα των δικτυακών υποδομών, σε συνδυασμό με την προώθηση νέων τεχνολογιών (τεχνητή νοημοσύνη, blockchain, μετα-κβαντική κρυπτογραφία), θα καθορίσει σε μεγάλο βαθμό το τοπίο της ασφάλειας των επικοινωνιών στην ψηφιακή εποχή.

Βιβλιογραφικές Αναφορές

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero trust architecture,” Zero Trust Architecture, vol. 800–207, no. 800–207, Aug. 2020, doi: https://doi.org/10.6028/nist.sp.800-207.

[2] FED, “Build Security Into Your Network’s DNA: The Zero Trust...,” Forrester. https://www.forrester.com/report/Build-Security-Into-Your-Networks-DNA-The-Zero-Trust-Network-Architecture/RES57047

[3] “BeyondCorp: A New Approach to Enterprise Security,” research.google. https://research.google/pubs/beyondcorp-a-new-approach-to-enterprise-security/

[4] B. D. Lund, T.-H. Lee, Z. Wang, T. Wang, and N. R. Mannuru, “Zero Trust Cybersecurity: Procedures and Considerations in Context,” Encyclopedia, vol. 4, no. 4, pp. 1520–1533, Oct. 2024, doi: https://doi.org/10.3390/encyclopedia4040099.

[5] N. G. S and Shankaramma, “Framework Analysis and Zero Trust Security Issues in Contemporary Network Systems,” 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions(CSITSS), pp. 1–6, Nov. 2024, doi: https://doi.org/10.1109/csitss64042.2024.10816783.

[6] J. Singh, “Zenith Armor : Advancing Security with Zero Trust Measures,” INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT, vol. 08, no. 04, pp. 1–5, Apr. 2024, doi: https://doi.org/10.55041/ijsrem31326.

[7] S. Ahmadi, “Zero Trust Architecture in Cloud Networks: Application, Challenges and Future Opportunities,” Journal of Engineering Research and Reports, vol. 26, no. 2, pp. 215–228, Feb. 2024, doi: https://doi.org/10.9734/jerr/2024/v26i21083.

[8] P. Dhiman et al., “A Review and Comparative Analysis of Relevant Approaches of Zero Trust Network Model,” Sensors, vol. 24, no. 4, p. 1328, Jan. 2024, doi: https://doi.org/10.3390/s24041328.

[9] V. Mavroudis, “Zero-Trust Network Access(ZTNA).” Available: https://arxiv.org/pdf/2410.20611

[10] C. Liu et al., “Dissecting zero trust: research landscape and its implementation in IoT,” Cybersecurity, vol. 7, no. 1, May 2024, doi: https://doi.org/10.1186/s42400-024-00212-0.

[11] E. Rescorla, “The Transport Layer Security(TLS) Protocol Version 1.3,” datatracker.ietf.org, Aug. 2018. https://datatracker.ietf.org/doc/html/rfc8446

[12] D. Boneh and V. Shoup, “A Graduate Course in Applied Cryptography,” 2017. Available: https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_4.pdf

[13] S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell, and D. Cooper, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List(CRL) Profile,”

IETF, May 01, 2008. https://datatracker.ietf.org/doc/html/rfc5280

[14] K. Bhargavan, B. Blanchet, and N. Kobeissi, “Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate,” IEEE Xplore, May 01, 2017. https://ieeexplore.ieee.org/document/7958594

[15] D. J. Bernstein and T. Lange, “Post-quantum cryptography,” Nature, vol. 549, no. 7671, pp. 188–194, Sep. 2017, doi: https://doi.org/10.1038/nature23461.

[16] J. Jose, A. Muhammad, and W.-C. Song, “Securing Digital Identity in the Zero Trust Architecture: A Blockchain Approach to Privacy-Focused Multi-Factor Authentication,” IEEE open journal of the Communications Society, pp. 1–1, Jan. 2024, doi: https://doi.org/10.1109/ojcoms.2024.3391728.

[17] B. Zhang, S. Yang, X. Zheng, and X. Wang, “STCA: Stacked Token-based Continuous Authentication Protocol for Zero Trust IoT,” 2022 IEEE Wireless Communications and Networking Conference(WCNC), pp. 1–6, Apr. 2024, doi: https://doi.org/10.1109/wcnc57260.2024.10571244.

[18] C. Bast and K.-H. Yeh, “Emerging Authentication Technologies for Zero Trust on the Internet of Things,” Symmetry, vol. 16, no. 8, pp. 993–993, Aug. 2024, doi: https://doi.org/10.3390/sym16080993.

[19] C. P. Autry, W. Henderson, Mykhailo Magal, and A. W. Roscoe, “dOISP: Rapidly Deployable Quantum-Resistant, Fully Decentralized Layer 3 Authentication and Encryption for True Zero-Trust Networking in Disparate Complex Networks,” 2021 International Conference on Electrical, Computer and Energy Technologies(ICECET), pp. 1–8, Jul. 2024, doi: https://doi.org/10.1109/icecet61485.2024.10698231.

[20] M. James, T. Newe, D. O’Shea, and G. D. O’Mahony, “Authentication and Authorization in Zero Trust IoT: A Survey,” pp. 1–7, Jun. 2024, doi: https://doi.org/10.1109/issc61953.2024.10603175.

[21] M. Schukat and P. Cortijo, “Public key infrastructures and digital certificates for the Internet of things,” 2015 26th Irish Signals and Systems Conference(ISSC), Jun. 2015, doi: https://doi.org/10.1109/issc.2015.7163785.

[22] C. Zhang, D. Yang, C. Chen, and P. Ren, “Checking Compliance of X.509 Digital Certificates over IPv6 with Specifications,” vol. 5280, pp. 445–450, Dec. 2020, doi: https://doi.org/10.1145/3444370.3444611.

[23] A. Yakubov, W. M. Shbair, A. Wallbom, D. Sanda, and R. State, “A blockchain-based PKI management framework,” NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, Apr. 2018, doi: https://doi.org/10.1109/noms.2018.8406325.

[24] “Insights of JSON Web Token,” International Journal of Recent Technology and Engineering, vol. 8, no. 6, pp. 1707–1710, Mar. 2020, doi: https://doi.org/10.35940/ijrte.f7689.038620.

[25] S. Dalimunthe, J. Reza, and A. Marzuki, “Model for Storing Tokens in Local Storage(Cookies) Using JSON Web Token(JWT) with HMAC(Hash-based Message Authentication Code) in E-Learning Systems,” Journal of Applied Engineering and Technological Science(JAETS), vol. 3, no. 2, pp. 149–155, Jun. 2022, doi: https://doi.org/10.37385/jaets.v3i2.662.

[26] None Achmad Ardiansyah and None Mepa Kurniasih, “Implementasi Algoritma AES-256 Untuk Pengamanan Layanan API Pada Restful Dengan Autentikasi Json Web Tokens,” Prosiding SNITek(Seminar Nasional Inovasi Teknologi), vol. 2, no. -, pp. 315–327, Jul. 2022, doi: https://doi.org/10.59134/prosidng.v2i-.138.

[27] A. Alkhulaifi and E.-S. M. El-Alfy, “Exploring Lattice-based Post-Quantum Signature for JWT Authentication: Review and Case Study,” IEEE Xplore, May 01, 2020. https://ieeexplore.ieee.org/abstract/document/9129505(accessed Mar. 14, 2024).

[28] Adiva Fiqri Nugraha, H. Kabetta, I Komang Setia Buana, and R Budiarto Hadiprakoso, “Performance and Security Comparison of Json Web Tokens(JWT) and Platform Agnostic Security Tokens(PASETO) on RESTful APIs,” Aug. 2023, doi: https://doi.org/10.1109/icocics58778.2023.10277377.

[29] M. Gupta, A. Gupta, Britto Raj S, and A. Sharma, “JWTAMH: JSON Web Tokens Based Authentication Mechanism for HADOOP.,” ICST Transactions on Scalable Information Systems, vol. 11, Jul. 2024, doi: https://doi.org/10.4108/eetsis.5429.

[30] Iskander Zulkarneev and K. A. Basalay, “JSON Web Tokens Lifecycle-Based Threat Classification,” pp. 1920–1924, Jun. 2024, doi: https://doi.org/10.1109/edm61683.2024.10615042.

[31] None Asyura Binti Sofian et al., “Enhancing Authentication Security: Analyzing Time-Based One-Time Password Systems,” vol. 1, no. 3, pp. 56–70, Jul. 2024, doi: https://doi.org/10.62951/ijcts.v1i3.25.

[32] X. Cao et al., “Dynamic Group Time-Based One-Time Passwords,” IEEE Transactions on Information Forensics and Security, vol. 19, pp. 4897–4913, Jan. 2024, doi: https://doi.org/10.1109/tifs.2024.3386350.

[33] P. Li, L. Pan, F. Chen, T. Hoang, and R. Wang, “TOTPAuth: A Time-based One Time Password Authentication Proof-of-Concept against Metaverse User Identity Theft,” Jun. 2023, doi: https://doi.org/10.1109/metacom57706.2023.00117.

[34] M. K. Sharma and M. J. Nene, “Dual factor third-party biometric-based authentication scheme using quantum one time passwords †,” Security and Privacy, Aug. 2020, doi: https://doi.org/10.1002/spy2.129.

[35] Ioana Boureanu et al., “LURK: Server-Controlled TLS Delegation,” Dec. 2020, doi: https://doi.org/10.1109/trustcom50675.2020.00036.

[36] W. Simpson and K. Foltz, “Assured Identity for Enterprise Level Security.” Available: https://www.iaeng.org/publication/WCE2017/WCE2017_pp440-445.pdf

[37] P. Thapar and U. Batra, “Implementation of Elliptical Curve Cryptography Based Diffie-Hellman Key Exchange Mechanism in Contiki Operating System for Internet of Things,” International Journal of Electrical and Electronics Research, vol. 10, no. 2, pp. 335–340, Jun. 2022, doi: https://doi.org/10.37391/ijeer.100245.

[38] Hiba Hilal Hadi and Ammar Ali Neamah, “An image encryption method based on modified elliptic curve Diffie-Hellman key exchange protocol and Hill Cipher,” Open Engineering, vol. 14, no. 1, Jan. 2024, doi: https://doi.org/10.1515/eng-2022-0552.

[39] H. K. M. Aljader and R. K. K. Ajeena, “The optimized Diffie-Hellman key exchange using the graphical method,” Journal of Discrete Mathematical Sciences and Cryptography, vol. 26, no. 6, pp. 1691–1697, 2023, doi: https://doi.org/10.47974/jdmsc-1615.

[40] J. Rangarajan, K Suriyakrishnaan, V. Meenakshi, and M. Tholkapiyan, “Crypto Analysis with Modified Diffie–Hellman Key Exchange based Sensor Node Security Improvement in Wireless Sensor Networks,” pp. 1222–1227, Feb. 2023, doi: https://doi.org/10.1109/icais56108.2023.10073749.

[41] K. H. Moussa, A. H. El-Sakka, S. Shaaban, and H. N. Kheirallah, “Group Security Authentication and Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange for LTE Military Grade Communication,” IEEE Access, vol. 10, pp. 80352–80364, 2022, doi: https://doi.org/10.1109/access.2022.3195304.

[42] J. Rivera, Khan Talha, W. Akbar, A. Muhammad, and W.-C. Song, “Secure enrollment token delivery for Zero Trust networks using blockchain,” 2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS), Sep. 2022, doi: https://doi.org/10.23919/apnoms56106.2022.9919940.

[43] “curl,” curl.se. https://curl.se/

[44] telefonicaid, “GitHub - telefonicaid/fiware-orion: Context Broker and CEF building block for context data management, providing NGSI interfaces.,” GitHub, Sep. 12, 2024. https://github.com/telefonicaid/fiware-orion(accessed Dec. 07, 2024).

[45] “What is OpenZiti? | OpenZiti,” openziti.io. https://openziti.io/docs/learn/introduction/

[46] “The Open Source platform for our smart digital future,” FIWARE. https://www.fiware.org/

[47] “libcurl - the multiprotocol file transfer library,” curl.se. https://curl.se/libcurl/

[48] Liodakis Odysseas, “GitHub - LiodakisOdysseas/ZTcURL: A Zero-Trust Networking Architecture-embedded command line client application that utilizes curl’s libcurl() library and the OpenZiti-C SDK to allow secure, authenticated access to OpenZiti services.,” GitHub, 2024. https://github.com/LiodakisOdysseas/ZTcURL(accessed Dec. 09, 2024).

[49] “Host OpenZiti Anywhere | OpenZiti,” Openziti.io, 2021. https://openziti.io/docs/learn/quickstarts/network/hosted (accessed Feb. 09, 2025).

[50] “Using systemd features to secure services,” Redhat.com, 2020. https://www.redhat.com/en/blog/systemd-secure-services (accessed Feb. 10, 2025).

[51] “Mastering systemd: Securing and sandboxing applications and services,” *Redhat.com*, 2019. https://www.redhat.com/en/blog/mastering-systemd (accessed Feb. 10, 2025).

[52] “Tunneling on Debian GNU/Linux | OpenZiti,” Openziti.io, 2025. https://openziti.io/docs/reference/tunnelers/linux/debian-package (accessed Feb. 10, 2025).

[53] B. Huber and F. Kandah, “Zero Trust+: A Trusted-based Zero Trust architecture for IoT at Scale,” Jan. 2024, doi: https://doi.org/10.1109/icce59016.2024.10444321.

[54] Z. Feng, Z. Peng, Q. Wang, and W. Qi, “A Dual-layer Zero Trust Architecture for 5G Industry MEC Applications Access Control,” 2022 IEEE 5th International Conference on Electronic Information and Communication Technology(ICEICT), Aug. 2022, doi: https://doi.org/10.1109/iceict55736.2022.9908891.