ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αντιμετώπιση Δικτυακών Επιθέσεων με Χρήση eXpress Data Path (XDP)

Αντιμετώπιση Επιθέσεων DNS Water Torture στο Επίπεδο Δεδομένων με Χρήση Μεθόδων Μηχανικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σταύρος Ν. Κορέντης

Επιβλέπων : Συμεών Παπαβασιλείου

Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αντιμετώπιση Δικτυακών Επιθέσεων με Χρήση eXpress Data Path (XDP)

Αντιμετώπιση Επιθέσεων DNS Water Torture στο Επίπεδο Δεδομένων με Χρήση Μεθόδων Μηχανικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σταύρος Ν. Κορέντης

Επιβλέπων : Συμεών Παπαβασιλείου

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Ιουλίου 2021

….....	….....	….....
Συμεών Παπαβασιλείου	Ευστάθιος Συκάς	Δημήτριος Σούντρης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

……...

Σταύρος Κορέντης

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

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον Καθηγητή Ε.Μ.Π. κ. Συμεών Παπαβασιλείου για την επίβλεψη αυτής της διπλωματικής εργασίας καθώς και τον Ομότιμο Καθηγητή Ε.Μ.Π. κ. Βασίλειο Μάγκλαρη για την εμπιστοσύνη που μου έδειξε και την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα. Ακόμη, θα ήθελα να ευχαριστήσω ιδιαίτερα τον Υποψήφιο Διδάκτορα Νίκο Κωστόπουλο για τις χρήσιμες συμβουλές του, το χρόνο που αφιέρωσε σε αυτήν την εργασία και την καθοδήγησή του σε όλα τα στάδια εκπόνησής της. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την υποστήριξη και συμπαράσταση τους, στη διάρκεια των σπουδών μου.

Περίληψη

Το Σύστημα Ονοματοδοσίας Τομέων (DNS) είναι απαραίτητο για την εύρυθμη λειτουργία του διαδικτύου, καθώς μέσω αυτού πραγματοποιείται η αντιστοίχιση των ονομάτων υπολογιστών σε διευθύνσεις IP και αντίστροφα. Αυτό το κάνει στόχο κακόβουλων χρηστών, οι οποίοι μέσω Κατανεμημένων Επιθέσεων Άρνησης Παροχής Υπηρεσιών (Distributed Denial of Service Attacks, DDoS Attacks) αποσκοπούν στην διατάραξη της λειτουργίας του.

Μία τέτοια επίθεση είναι η λεγόμενη DNS Water Torture. Στόχος της είναι ο υπεύθυνος εξυπηρετητής (Authoritative Server) μίας ζώνης DNS και σκοπός της επίθεσης είναι να πλημμυρίζει το θύμα με πολύ μεγάλο όγκο άκυρων ερωτημάτων, τα οποία προέρχονται από αναδρομικούς εξυπηρετητές (Recursive Resolvers), έτσι ώστε να εξαντληθεί η υπολογιστική ισχύς του εξυπηρετητή και να καταστεί ανίκανος να απαντά σε έγκυρα ερωτήματα καλόβουλων χρηστών. Τα ερωτήματα αυτά, περιέχουν τυχαία, μη επαναλαμβανόμενα ονόματα έτσι ώστε να παρακάμπτουν την προσωρινή μνήμη των αναδρομικών εξυπηρετητών και να φτάνουν πάντα στο θύμα.

Ο μηχανισμός που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας αποσκοπεί στην αντιμετώπιση αυτής της επίθεσης, συνδυάζοντας τις δυνατότητες προγραμματισμού στο επίπεδο δεδομένων με τεχνικές επιβλεπόμενης μηχανικής μάθησης.

Πιο αναλυτικά, τα ερωτήματα που δέχεται ο αναδρομικός εξυπηρετητής, ελέγχονται ως προς την εγκυρότητα του ονόματος DNS που περιέχουν, πριν προωθηθούν στον αντίστοιχο αρμόδιο εξυπηρετητή. Ο έλεγχος αυτός γίνεται με την χρήση του ταξινομητή Naive Bayes, ο οποίος σύμφωνα με την εκπαίδευση που έχει γίνει, κάνει μία πρόβλεψη σχετικά με το εάν το DNS όνομα του ερωτήματος είναι έγκυρο ή όχι. Ο Naive Bayes ενδείκνυται για εφαρμογές επεξεργασίας φυσικής γλώσσας και αυτό τον καθιστά καλή επιλογή για την αντιμετώπιση της επίθεσης DNS Water Torture. Ο ταξινομητής έχει υλοποιηθεί με τη χρήση extended Berkeley Packet Filter (eBPF) και το πρόγραμμα έχει προσαρτηθεί στο eXpress Data Path (XDP) hook. Το XDP hook βρίσκεται στο χαμηλότερο επίπεδο της στοίβας δικτύου του Linux και η επεξεργασία των πακέτων στο σημείο αυτό είναι πολύ πιο αποτελεσματική και πιο γρήγορη, αφού δεν έχει πραγματοποιηθεί ακόμη κατανομή μνήμης, για το πακέτο, από τον πυρήνα. Ανάλογα με το αποτέλεσμα της πρόβλεψης, το πακέτο συνεχίζει την πορεία του ή απορρίπτεται. Ως αποτέλεσμα, τα ερωτήματα των καλόβουλων χρηστών εξυπηρετούνται κανονικά ενώ το μεγαλύτερο ποσοστό της κακόβουλης κίνησης δεν φτάνει τον Authoritative εξυπηρετητή.

Σκοπός αυτής της εργασίας είναι η μελέτη της απόδοσης αυτού του μηχανισμού και της βελτίωσης χρησιμοποιώντας το XDP.

Λέξεις Κλειδιά : Σύστημα Ονοματοδοσίας Τομέων, Κατανεμημένες Επιθέσεις Άρνησης Παροχής Υπηρεσιών, επίθεση DNS Water Torture, Μηχανική Μάθηση, Ταξινομητής Naive Bayes, extended Berkeley Packet Filter, eXpress Data Path

Abstract

The Domain Name System (DNS) is essential for the operation of the Internet, as it assigns computer names to IP addresses and vice versa. This makes it a target of malicious users, who through Distributed Denial of Service attacks (DDoS attacks) aim to disrupt its operation.

Such an attack is the so-called DNS Water Torture. Its target is the Authoritative Server of a DNS zone and the aim of the attack is to flood the victim with a large volume of invalid queries, which come from recursive Servers (Recursive Resolvers), in order to exhaust the Server's computing power and make it unable to answer benign users' valid queries. These invalid queries contain random, non-repetitive names in order to bypass the Recursors' cache and thus always reach the victim.

The mechanism developed in the context of this dissertation aims to address this attack by combining data plane programming capabilities with supervised machine learning techniques.

More specifically, queries received by the Recursive Server are checked for the validity of the DNS name they contain, before being forwarded to the Authoritative Server. This check is done using the Naive Bayes classifier, which according to its training, makes a prediction as to whether the DNS name of the query is valid or not. Naive Bayes is suitable for natural language processing applications and this makes it a good choice for mitigating DNS Water Torture attack. The classifier is implemented using the extended Berkeley Packet Filter (eBPF) and the program is attached to the eXpress Data Path (XDP) hook. The XDP hook is at the lowest level of the Linux network stack and packet processing at this point is much more efficient and faster, as the kernel has not yet allocated memory for the package. Depending on the outcome of the prediction, the package passes or is dropped. As a result, benign users' queries are served normally while most of the malicious traffic does not reach the Authoritative Server.

The purpose of this work is to study the performance of this mechanism and the improvement using XDP.

Keywords : Domain Name System (DNS), Distributed Denial of Service attacks (DDoS), DNS Water Torture attack, Machine Learning, Naive Bayes Classifier, extended Berkeley Packet Filter (eBPF), eXpress Data Path (XDP)

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

1. Εισαγωγή

1.1. Περιγραφή του προβλήματος

Η ραγδαία εξέλιξη της τεχνολογίας, έχει κάνει το διαδίκτυο αναπόσπαστο κομμάτι της καθημερινότητας των ανθρώπων. Μέσω αυτού, υπολογιστικά συστήματα έχουν την δυνατότητα να ανταλλάσσουν ταχύτατα πληροφορίες μεταξύ τους και να παρέχουν μία πληθώρα υπηρεσιών στους χρήστες του.

Η επικοινωνία μεταξύ των υπολογιστών του διαδικτύου προϋποθέτει εκείνοι να μπορούν να αναγνωριστούν μεταξύ τους και για την επίτευξη αυτού, χρησιμοποιούνται οι διευθύνσεις IP.

Στην πράξη όμως, οι άνθρωποι δεν μπορούν να απομνημονεύουν τέτοιες διευθύνσεις, αφού αυτές αποτελούνται από αριθμούς. Γι' αυτό χρησιμοποιούνται τα ονόματα υπολογιστών.

Η αντιστοίχιση των ονομάτων αυτών με τις διευθύνσεις IP γίνεται μέσω του Συστήματος Ονοματοδοσίας Τομέων (DNS). Κάθε φορά που ένας χρήστης επιχειρεί να αποκτήσει πρόσβαση σε μια υπηρεσία στο διαδίκτυο χρησιμοποιώντας ένα όνομα υπολογιστή, το DNS παρέχει την μετάφραση αυτού του ονόματος στην σωστή διεύθυνση IP.

Συνεπώς, είναι φανερό ότι το DNS είναι απαραίτητο για την εύρυθμη λειτουργία του διαδικτύου. Αυτό συγχρόνως το κάνει και στόχο κακόβουλων χρηστών, που θέλουν να διαταράξουν αυτήν την εύρυθμη λειτουργία για τους δικούς τους σκοπούς.

Ένας τρόπος για να το πετύχουν αυτό είναι οι κατανεμημένες επιθέσεις άρνησης παροχής υπηρεσιών (Distributed Denial of Service attacks ή DDoS attacks). Τέτοιες επιθέσεις αποσκοπούν στο να καταστήσουν μια υπηρεσία μη προσβάσιμη στους πελάτες της, εξαντλώντας την με μεγάλο όγκο κίνησης από πολλαπλές πηγές.

Μία τέτοια επίθεση είναι η λεγόμενη DNS Water Torture. Η επίθεση αυτή πλημμυρίζει τον αρμόδιο (Authoritative) εξυπηρετητή μιας ζώνης DNS με πολύ μεγάλο όγκο άκυρων ερωτημάτων, δηλαδή μνημάτων για ονόματα που δε βρίσκονται αποθηκευμένα στα αρχεία του εξυπηρετητή αυτού. Τα ερωτήματα αυτά στέλνονται από τον επιτιθέμενο σε αναδρομικούς εξυπηρετητές και προωθούνται από αυτούς στον Authoritative Server. Επίσης, περιέχουν ονόματα που δημιουργούνται τυχαία και είναι μη επαναλαμβανόμενα έτσι ώστε να παρακάμπτουν την προσωρινή μνήμη των αναδρομικών εξυπηρετητών και να φτάνουν πάντα στο θύμα. Σκοπός είναι να εξαντληθεί η υπολογιστική ισχύς του εξυπηρετητή και να καταστεί ανίκανος να απαντά σε έγκυρα ερωτήματα καλόβουλων χρηστών.

1.2. Σκοπός της εργασίας

Οι επιτιθέμενοι μπορούν να χρησιμοποιήσουν έναν υψηλό αριθμό μολυσμένων συσκευών (bots) για να στείλουν μεγάλο όγκο κίνησης στο θύμα. Επομένως, η αντιμετώπισή της απαιτεί μηχανισμούς, οι οποίοι επιτυγχάνουν υψηλές επιδόσεις κατά το διαχωρισμό κακόβουλης και καλόβουλης κίνησης.

Ο μηχανισμός που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας ταξινομεί τα ερωτήματα που γίνονται στον αναδρομικό εξυπηρετητή, σε έγκυρα ή άκυρα , με σκοπό την απόρριψη των άκυρων και την προώθηση των έγκυρων, συνδυάζοντας τις δυνατότητες προγραμματισμού στο επίπεδο δεδομένων (Data Plane) με τεχνικές επιβλεπόμενης μηχανικής μάθησης.

Συγκεκριμένα, γίνεται χρήση του Naive Bayes Classifier για την ταξινόμηση των ονομάτων των ερωτημάτων, σε έγκυρα ή άκυρα. Ο ταξινομητής έχει υλοποιηθεί με extended Berkeley Packet Filter (eBPF), το οποίο είναι σύνολο εντολών και περιβάλλον εκτέλεσης εντός του πυρήνα του Linux. Το eBPF επιτρέπει την προσάρτηση κώδικα στο eXpress Data Path (XDP) hook, το οποίο είναι το χαμηλότερο επίπεδο της στοίβας δικτύου του Linux και πετυχαίνει πολύ υψηλές επιδόσεις κατά την επεξεργασία εισερχόμενων πακέτων, καθώς στο σημείο που γίνεται η επεξεργασία δεν έχει γίνει ακόμα κατανομή μνήμης για το πακέτο από τον πυρήνα. Έτσι, για κάθε εισερχόμενο πακέτο, γίνεται έλεγχος του ονόματος του ερωτήματος που περιέχει και ανάλογα με την πρόβλεψη του Naive Bayes, το πακέτο συνεχίζει την πορεία του ή απορρίπτεται. Ως αποτέλεσμα, το μεγαλύτερο ποσοστό της καλόβουλης κίνησης διέρχεται κανονικά από τον αναδρομικό εξυπηρετητή ενώ τα άκυρα ερωτήματα απορρίπτονται και δεν φτάνουν στον Authoritative Server.

Στόχος αυτής της εργασίας είναι η ανάλυση της απόδοσης αυτού του μηχανισμού και όχι η ακρίβεια του Naive Bayes στις ταξινομήσεις. Η τελευταία, έχει μελετηθεί [35] και τα αποτελέσματα δείχνουν ότι μπορεί να επιτευχθεί υψηλό ποσοστό ακρίβειας προβλέψεων.

1.3. Δομή της εργασίας

Η εργασία έχει την παρακάτω δομή:

- **Κεφάλαιο 2** : παρουσιάζονται οι θεωρητικές γνώσεις που είναι απαραίτητες για την κατανόηση του μηχανισμού που υλοποιήθηκε
- **Κεφάλαιο 3** : περιγράφεται ο μηχανισμός που αναπτύχθηκε, ο τρόπος λειτουργίας του καθώς και τα προβλήματα-περιορισμοί που αντιμετωπίστηκαν κατά την υλοποίησή του
- **Κεφάλαιο 4** : αναλύονται τα πειράματα που εκτελέστηκαν και γίνεται αξιολόγηση των αποτελεσμάτων τους
- **Κεφάλαιο 5** : περιλαμβάνονται τα συμπεράσματα της εργασίας και προτείνονται μελλοντικές επεκτάσεις

2. Θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο, αναλύεται το θεωρητικό υπόβαθρο πάνω στο οποίο βασίζεται η παρούσα διπλωματική εργασία.

2.1. DNS

Το Σύστημα Ονοματοδοσίας Διαδικτύου (Domain Name System, DNS) είναι ένα ιεραρχικό σύστημα ονοματοδοσίας για δίκτυα υπολογιστών. Το σύστημα αυτό ουσιαστικά λειτουργεί σαν κατάλογος και αντιστοιχίζει τις διευθύνσεις IP με υπολογιστικούς πόρους. Η αντιστοίχιση ονομάτων με αριθμητικές διευθύνσεις είναι φυσική αναγκαιότητα καθώς οι άνθρωποι θυμούνται ευκολότερα ονόματα παρά αριθμούς και συνδέουν πιο εύκολα έννοιες και σημασίες με τα πρώτα παρά με τους δεύτερους.

Το DNS είναι αναπόσπαστο κομμάτι του διαδικτύου και οποιοδήποτε πρόβλημα στο Σύστημα Ονοματοδοσίας, έχει άμεσο αντίκτυπο στους χρήστες του διαδικτύου, καθώς παρεμποδίζεται η πρόσβασή τους σε ιστοσελίδες και διάφορες υπηρεσίες.

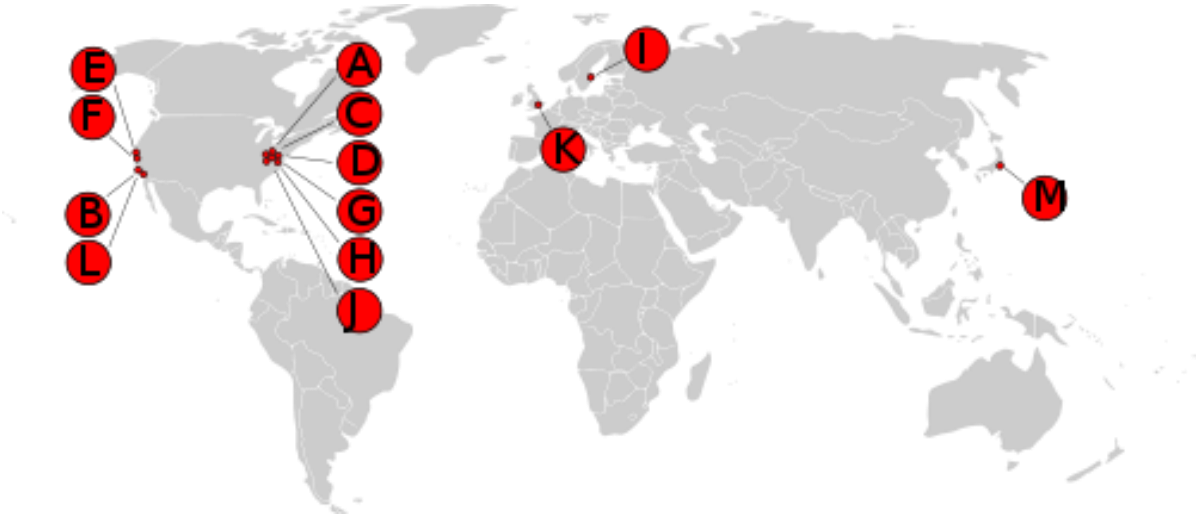
Επομένως, είναι απαραίτητο να διασφαλίζεται κάθε στιγμή η εύρυθμη λειτουργία του. Για την καλύτερη περιγραφή και κατανόηση των κινδύνων που απειλούν το DNS, χρειάζεται αρχικά να αναλυθεί ο τρόπος λειτουργίας και τα βασικά χαρακτηριστικά του.

2.1.1. Δομή

Το DNS χρησιμοποιεί μια ιεραρχία για τη διαχείριση του κατανεμημένου συστήματος βάσης δεδομένων. Η ιεραρχία αυτή, που ονομάζεται επίσης χώρος ονομάτων τομέων, έχει δεντρική δομή. Αποτελείται από ένα σύνολο εξυπηρετητών DNS (DNS Servers), οι οποίοι είναι κατάλληλα κατανεμημένοι σε αυτήν. Το DNS, λοιπόν, είναι ένα εύρωστο σύστημα, καθώς δεν εξαρτάται από ένα μοναδικό σημείο αποτυχίας. Οι πληροφορίες που περιέχει ένας εξυπηρετητής βρίσκονται κάθε στιγμή αποθηκευμένες και στους άλλους Servers της ιεραρχίας.

Το δέντρο DNS έχει έναν μόνο τομέα στην κορυφή της δομής του που ονομάζεται Root Domain (τομέας ρίζα). Μια τελεία (.) είναι ο ορισμός για τον τομέα αυτόν. Οι εξυπηρετητές που σχετίζονται με αυτόν τον τομέα ονομάζονται Root DNS Servers και είναι παγκόσμια κατανεμημένοι σε 13 φυσικές τοποθεσίες, με τις περισσότερες από αυτές να βρίσκονται στην Βόρεια Αμερική. Στην πραγματικότητα, οι

εξυπηρετητές αυτοί δεν είναι μόνο 13, αλλά κάθε εξυπηρετητής αποτελείται από ένα σύμπλεγμα εξυπηρετητών με κοινή διεύθυνση IP. Ο καταλληλότερος εξυπηρετητής του συμπλέγματος επιλέγεται με γεωγραφικά κριτήρια, δηλαδή ποιος εξυπηρετητής βρίσκεται κοντύτερα στον υπολογιστή που διατυπώνει το ερώτημα DNS.



Σχήμα 2.1 - Οι 13 Root DNS Servers.

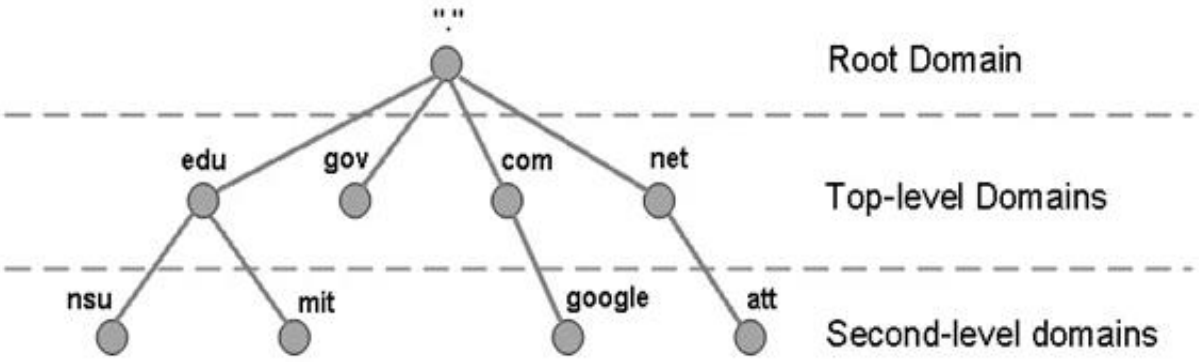
Κάτω από τον Root βρίσκονται οι τομείς ανώτατου επιπέδου (Top Level Domains, TLD) που χωρίζουν την ιεραρχία DNS σε τμήματα και κάτω από αυτούς βρίσκονται οι τομείς δεύτερου επιπέδου (Second Level Domains, SLD ή 2LD).

Παρακάτω αναφέρονται οι πιο γνωστοί τομείς DNS ανώτερου επιπέδου και οι τύποι οργανισμών που τους χρησιμοποιούν [2]. Κάτω από τους τομείς ανώτατου επιπέδου, ο χώρος ονομάτων τομέων χωρίζεται περαιτέρω σε υποτομείς (Subdomains) που αντιπροσωπεύουν μεμονωμένους οργανισμούς.

TLDs	Ανατίθεται σε
com	Εταιρείες
edu	Εκπαιδευτικά ιδρύματα
gov	Κυβερνητικούς οργανισμούς
mil	Στρατιωτικούς οργανισμούς
org	Οργανισμούς εκτός των ανωτέρω

Πίνακας 2.1 - Top Level Domains (TLDs).

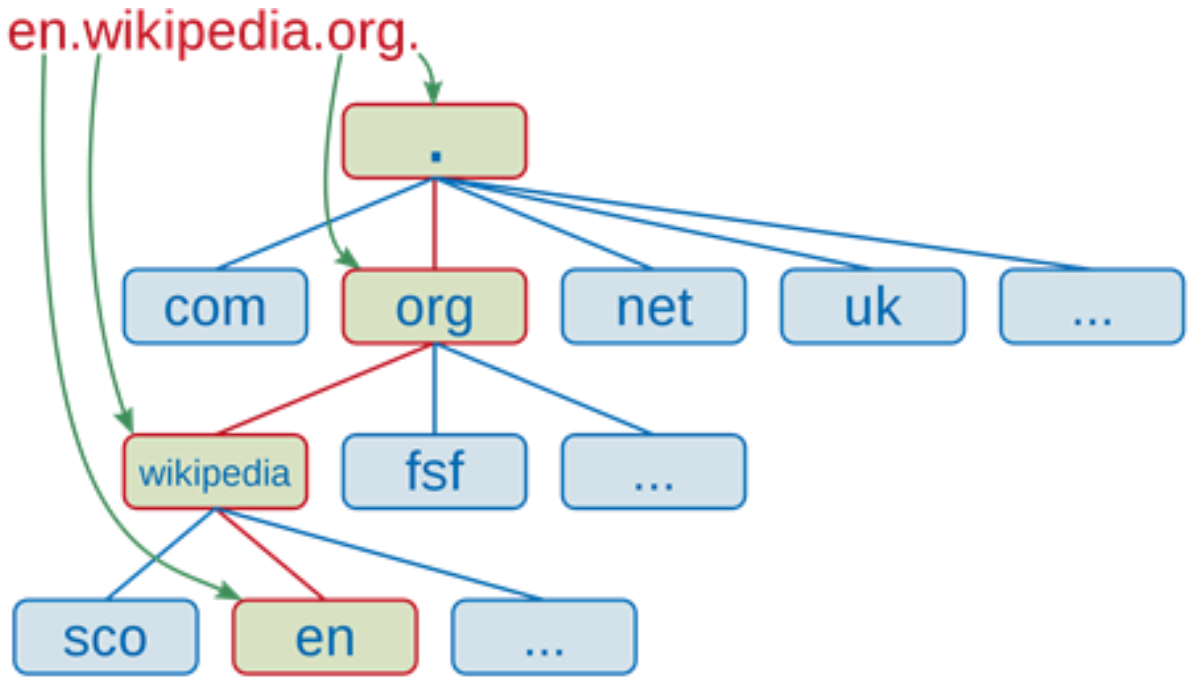
Πρόσθετοι τομείς ανώτερου επιπέδου οργανώνουν γεωγραφικά τον χώρο ονομάτων τομέων. Για παράδειγμα, ο τομέας ανώτατου επιπέδου για την Ελλάδα είναι gr. Η ιεραρχία DNS απεικονίζεται παρακάτω :



Σχήμα 2.2 - Η ιεραρχία DNS.

Κάθε κόμβος ή φύλλο, σε αυτή τη δεντρική δομή, έχει μία ετικέτα (Label), και μηδέν ή περισσότερες εγγραφές (Resource Records) οι οποίες περιέχουν πληροφορίες σχετικά με κάποιο όνομα τομέα DNS (Domain name). Το διαδίκτυο είναι νοητά χωρισμένο σε αυτούς τους τομείς (Domains), οι οποίοι παριστάνονται ως ένα υπό-δέντρο του δέντρου DNS. Ένα Domain είναι δυνατόν να περιέχει οντότητες ή να χωρίζεται σε περισσότερους υποτομείς για να διευκολύνεται η διαχείριση των κεντρικών υπολογιστών ενός οργανισμού. Κάθε τομέας σε ένα υπό-δέντρο θεωρείται μέρος όλων των τομέων που βρίσκονται πάνω από αυτόν.

Η ακριβής θέση ενός πόρου στην ιεραρχία DNS προσδιορίζεται από το πλήρως πιστοποιημένο όνομα τομέα(Fully Qualified Domain Name, FQDN). Το FQDN προκύπτει από την ένωση όλων των labels, ξεκινώντας από εκείνο του πόρου μέχρι τη ρίζα του δέντρου DNS με τελείες ανάμεσά τους.



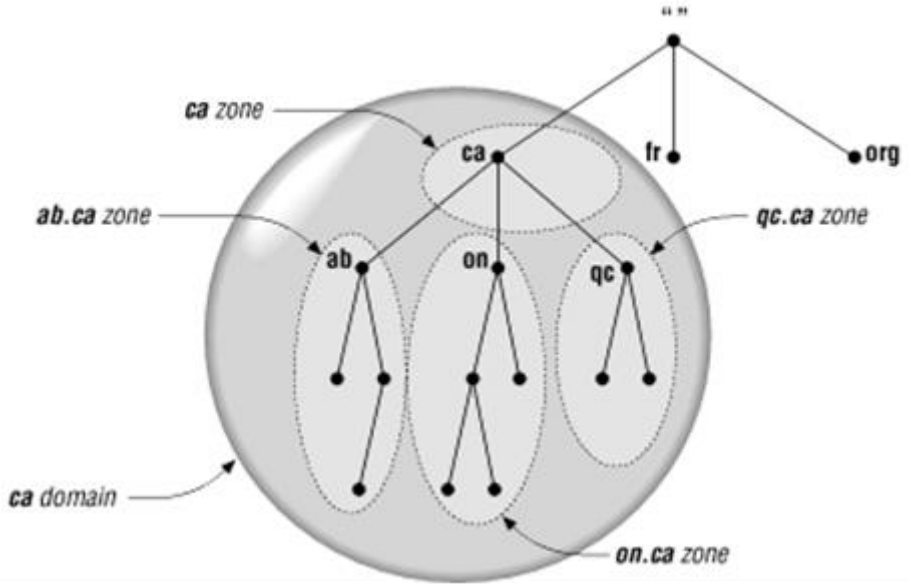
Σχήμα 2.3 - Σχηματισμός του FQDN του en.wikipedia.org.

Η διαχείριση των εγγραφών DNS (DNS records), δηλαδή των αντιστοιχίσεων μεταξύ ονομάτων υπολογιστικών πόρων και διευθύνσεων IP (ή άλλων πληροφοριών σε ορισμένες περιπτώσεις), στους DNS Servers, δε γίνεται με βάση τα Domains, αλλά τις ζώνες (zones). Ζώνη ονομάζεται μια ομάδα τομέων και υποτομέων για τους οποίους ένας DNS Server είναι υπεύθυνος για τη διατήρηση της βάσης δεδομένων DNS και των πληροφοριών διευθύνσεων για αυτούς τους τομείς. Ένας τέτοιος Server ονομάζεται Authoritative DNS Server.

Μία ζώνη περιλαμβάνει ένα σύνολο από Authoritative εξυπηρετητές DNS που διατηρούν την ίδια πληροφορία και είναι υπεύθυνοι να απαντάνε σε ερωτήματα που αφορούν αυτήν την ζώνη. Ένας Authoritative Server μπορεί να είναι είτε πρωταρχικός εξυπηρετητής (Primary Server, Master) είτε δευτερεύων εξυπηρετητής (Secondary Server, Slave). Ο Primary εξυπηρετητής μιας ζώνης είναι αυτός που έχει αποθηκευμένες όλες τις έγκυρες εγγραφές αυτής της ζώνης και προσδιορίζεται στην εγγραφή Start-Of-Authority (SOA). Ένας δευτερεύων εξυπηρετητής διατηρεί αντίγραφο του αρχείου ζώνης του Primary Server. Τα περιεχόμενα του αρχείου ζώνης του πρωταρχικού εξυπηρετητή μεταφέρονται στους δευτερεύοντες εξυπηρετητές με ενέργειες που ονομάζονται μεταφορές ζώνης (zone transfers). Σκοπός της διαίρεσης ενός Domain σε πολλές ζώνες είναι η αποτελεσματικότερη διαχείρισή του με την απόδοση της διαχείρισης των ζωνών σε διαφορετικά τμήματα μιας οργάνωσης (zone delegation), ο διαμοιρασμός της κίνησης σε περισσότερους DNS Servers και η ταχύτερη επίλυση ονομάτων.

Για παράδειγμα, στο παρακάτω σχήμα φαίνονται 4 ζώνες διαχείρισης εγγραφών DNS, καθεμία από τις οποίες προσδιορίζεται με ένα διακεκομμένο κύκλο. Καθεμία

από τις ζώνες αυτές έχει αποκλειστική κυριότητα και ευθύνη για τη διαχείριση των εγγραφών που περιλαμβάνει.



Σχήμα 2.4 - Διαίρεση της ιεραρχίας DNS σε ζώνες.

2.1.2. Λειτουργία

Το DNS, όπως προαναφέρθηκε, είναι υπεύθυνο για την μετάφραση ενός ονόματος υπολογιστή σε διεύθυνση IP. Ο μηχανισμός επίλυσης ονόματος λειτουργεί με χρήση των αναδρομικών DNS εξυπηρετητών που ονομάζονται recursors (ή Resolvers). Αυτοί οι εξυπηρετητές εντοπίζουν και καθορίζουν τους name Servers που είναι υπεύθυνοι για το όνομα προς επίλυση με μια σειρά ερωτημάτων, ξεκινώντας από την ετικέτα που βρίσκεται πιο δεξιά στο όνομα (Top-Level).

Για την σωστή λειτουργία τους, οι Resolvers διατηρούν ένα αρχείο με τις διευθύνσεις των Root DNS Servers, το οποίο μπορεί να ενημερώνεται από τους διαχειριστές τους.

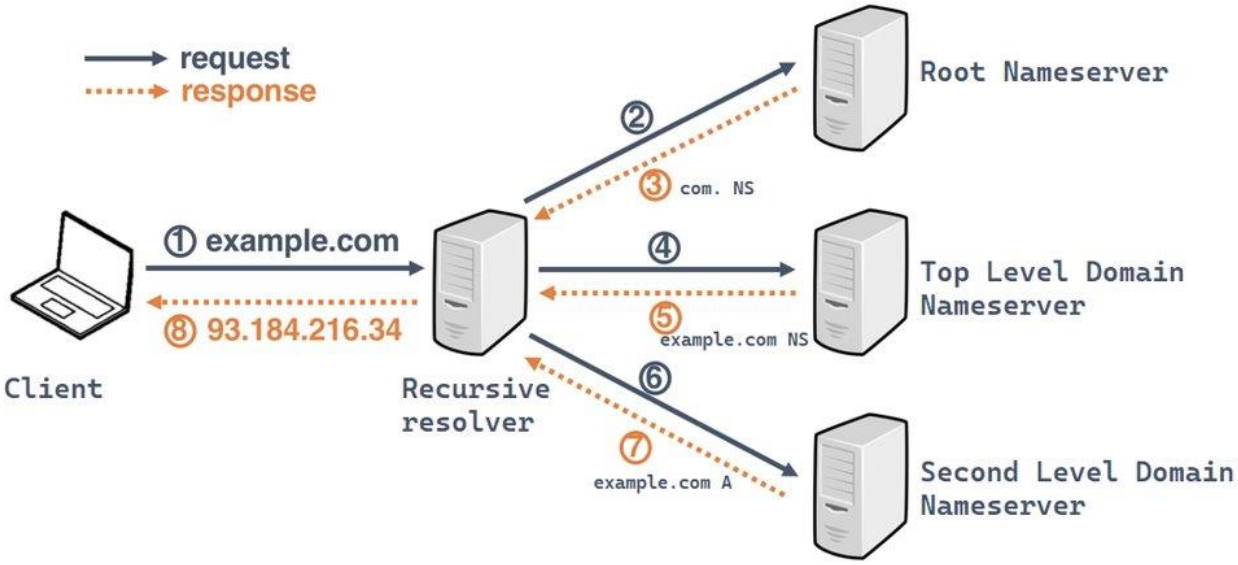
Όταν ένας αναδρομικός εξυπηρετητής (Resolver), λάβει ένα αίτημα για επίλυση ονόματος από κάποιον υπολογιστή, έστω www.example.com, στέλνει ένα καινούργιο αίτημα στον Root DNS Server που βρίσκεται πλησιέστερα, ρωτώντας τον ποια είναι η διεύθυνση IP του υπολογιστή με FQDN www.example.com. Ο Root DNS Server είναι υπεύθυνος να απαντά έως ένα label βάθος στην ιεραρχία DNS, δηλαδή να γνωρίζει πληροφορίες μόνο για το Top-Level Domain της ερώτησης. Έτσι, θα αγνοήσει το πρώτο τμήμα του FQDN και θα αναζητήσει στα αρχεία του ποιος είναι ο TLD DNS Server που είναι υπεύθυνος για τη ζώνη com και τη διεύθυνση IP του και θα την στείλει στον Resolver.

Ο Resolver θα χρησιμοποιήσει αυτήν την πληροφορία και θα στείλει ένα νέο αίτημα στον TLD DNS Server, που είναι υπεύθυνος για τη ζώνη com, ρωτώντας τον ποια είναι η IP του υπολογιστή με FQDN www.example.com. Ο TLD Server, όμως, είναι υπεύθυνος να απαντάει μέχρι δύο labels βάθος στην ιεραρχία DNS. Συνεπώς, θα αναζητήσει στα αρχεία του ποιος είναι ο Authoritative DNS Server που είναι υπεύθυνος για τη ζώνη example.com και τη διεύθυνση IP του και θα την στείλει στον Resolver.

Στην συνέχεια ο Resolver θα στείλει αίτημα στον Authoritative DNS Server της ζώνης example.com, ρωτώντας και πάλι ποια είναι η διεύθυνση IP του υπολογιστή με FQDN www.example.com.

Ο Authoritative DNS Server επειδή ξέρει ότι είναι υπεύθυνος για τη ζώνη example.com, θα αναζητήσει στα αρχεία του την αντιστοίχιση του FQDN και της διεύθυνσης IP και θα επιστρέψει στον recursor τη διεύθυνση IP του υπολογιστή για τον οποίο ρώτησε.

Τέλος, ο Resolver επιστρέφει τη διεύθυνση IP στον υπολογιστή από τον οποίο ξεκίνησε η διαδικασία.



Σχήμα 2.5 - Διαδικασία επίλυσης του ονόματος example.com.

Η παραπάνω διαδικασία ονομάζεται επαναληπτική (iterative) επίλυση ερωτήματος. Η αναζήτηση ενός ονόματος μπορεί να γίνει και με αναδρομικό (recursive) τρόπο. Σε αυτήν την περίπτωση, ο κάθε Server ζητάει από τον αμέσως κατώτερο στην ιεραρχία Server να αναλάβει εκείνος την αναζήτηση της εγγραφής εκ μέρους του, όπως ακριβώς λειτουργεί και ο recursor για τον υπολογιστή που διατυπώνει το ερώτημα DNS. Ωστόσο, ο τρόπος που εφαρμόζεται, συνήθως, στην πράξη είναι ο

επαναληπτικός τρόπος. Σε κάθε περίπτωση, όμως, ο αρχικός υπολογιστής διατυπώνει ένα αναδρομικό ερώτημα στον recursor.

Αυτός ο μηχανισμός ωστόσο, θέτει μεγάλο φόρτο στους Root DNS Servers, καθώς για κάθε επίλυση ονόματος στο διαδίκτυο θα έπρεπε να στέλνεται ένα αίτημα σε έναν από αυτούς. Γι' αυτόν τον λόγο χρησιμοποιείται η τεχνική της προσωρινής αποθήκευσης από τους Resolvers για να αποφορτιστούν οι Root Servers, και ως εκ τούτου οι Root DNS Servers καταλήγουν να χρησιμοποιούνται σε σχετικά λίγες επιλύσεις ονομάτων.

Η προσωρινή αποθήκευση απαντήσεων από ερωτήματα που έχει δεχτεί, ένας Resolver, βοηθάει στην ταχύτερη λειτουργία του DNS καθώς αν ένα ερώτημα που έχει απαντηθεί, επαναληφθεί ,ο Resolver θα αναζητήσει την απάντηση στην προσωρινή του μνήμη (cache) και θα την στείλει χωρίς να εκτελέσει τα παραπάνω βήματα. Επομένως, στην παραπάνω διαδικασία προστίθεται και το βήμα κατά το οποίο ο Resolver αναζητά στην cache του την απάντηση στο ερώτημα που του έχει τεθεί και εάν την βρει απαντάει απευθείας στον υπολογιστή που έστειλε το αίτημα. Εάν όχι, εκτελεί την διαδικασία κανονικά και στο τέλος αποθηκεύει την απάντηση στην προσωρινή του μνήμη.

Ωστόσο, μία τέτοια εγγραφή δεν μπορεί να παραμείνει στην cache για πάντα καθώς μπορεί να μην ισχύει μετά από κάποιο χρονικό διάστημα (αλλαγή διεύθυνσης IP ενός υπολογιστή). Για αυτόν τον λόγο η κάθε απάντηση έχει μια παράμετρο, που ονομάζεται TTL (Time To Live) και υποδεικνύει τον χρόνο, σε δευτερόλεπτα, για τον οποίο μία εγγραφή θεωρείται έγκυρη. Όταν λήξει το TTL, η αποθηκευμένη εγγραφή αφαιρείται από την προσωρινή μνήμη. Το TTL κάθε εγγραφής ορίζεται στο αρχείο ζώνης του Authoritative DNS Server.

Η επιλογή του Resolver που θα χρησιμοποιηθεί για την επίλυση του ονόματος είναι ελεύθερη. Ένα δίκτυο λοιπόν, μπορεί να χρησιμοποιήσει είτε το recursor ενός παρόχου διαδικτύου είτε να χρησιμοποιήσει recursors που τους διαχειρίζονται άλλοι οργανισμοί και είναι ρυθμισμένοι να εξυπηρετούν τα αναδρομικά ερωτήματα DNS οποιουδήποτε υπολογιστή. Στην τελευταία περίπτωση, αυτοί οι εξυπηρετητές ονομάζονται Open Resolvers.

Οι open Resolvers μπορούν να χρησιμοποιηθούν εναλλακτικά όταν για παράδειγμα ο Resolver που παρέχεται από τον DNS πάροχο έχει κάποια βλάβη ή έχει χαμηλές επιδόσεις και δημιουργεί καθυστέρηση στην διαδικασία επίλυσης ονόματος. Ωστόσο, προκύπτουν αρκετοί κίνδυνοι επειδή οι open Resolvers απαντούν σε ερωτήματα από οποιονδήποτε χρήστη. Αυτοί οι κίνδυνοι, οδηγούν σε δικτυακές επιθέσεις που θα αναλυθούν σε επόμενα κεφάλαια.

2.1.3. Τύποι Εγγαφών

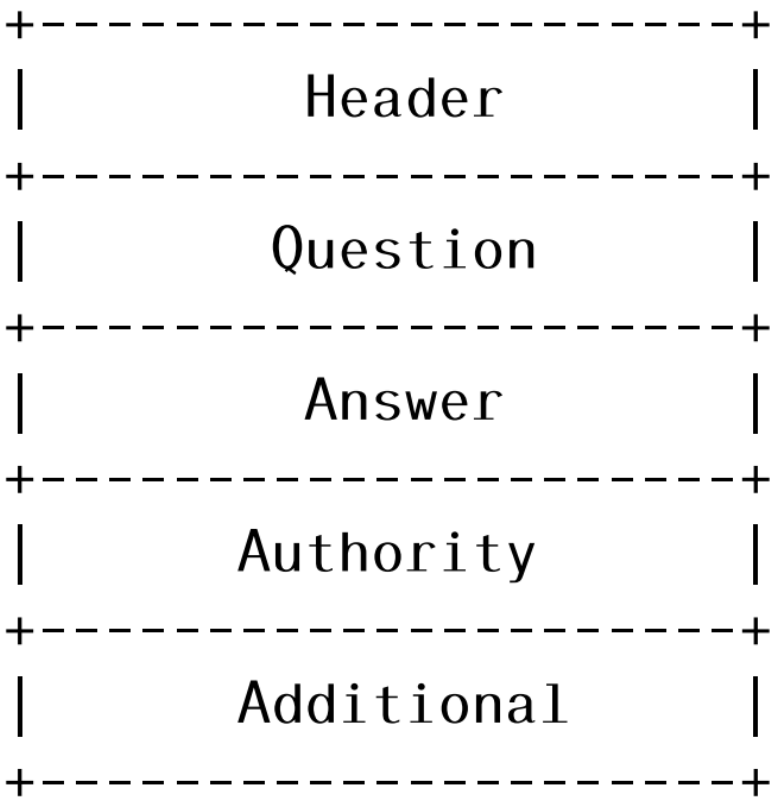
Το DNS χρησιμοποιεί ένα σύνολο από τύπους εγγραφών για να απεικονίσει ένα όνομα υπολογιστή (FQDN) σε διάφορους δικτυακούς πόρους. Αυτές οι εγγραφές ονομάζονται εγγραφές πόρων (Resource Records ή RR) και έχουν κάποια περιεχόμενα, τα οποία θα αναλυθούν στην συνέχεια. Οι εγγραφές αυτές αποθηκεύονται στα αρχεία ζώνης των Authoritative εξυπηρετητών DNS. Όταν επιστρέφεται μία απάντηση σε έναν υπολογιστή από έναν Resolver, ουσιαστικά στέλνει τα Resource Records που σχετίζονται με την αρχική ερώτηση του υπολογιστή.

Τα περιεχόμενα των Resource Records είναι τα εξής [6]:

- NAME: Το FQDN του υπολογιστικού πόρου που αφορά η συγκεκριμένη εγγραφή
- TYPE: Ο τύπος της εγγραφής. Υπάρχουν πολλοί διαφορετικοί τύποι εγγραφών DNS, ο καθένας από τους οποίους εξυπηρετεί διαφορετικό σκοπό. Ενδεικτικά, κάποιοι από αυτούς παρουσιάζονται παρακάτω:
    - A: χρησιμοποιείται για την αντιστοίχιση ενός FQDN σε μια διεύθυνση IP των 32 bits.
    - AAAA: χρησιμοποιείται για την αντιστοίχιση ενός FQDN σε μια διεύθυνση IPv6 των 128 bits.
    - ANY: επιστρέφει όλες τις εγγραφές που γνωρίζει ο name Server που ερωτάται.
    - NS: οι εγγραφές αυτές προσδιορίζουν τους εξυπηρετητές DNS που είναι Authoritative για μία ζώνη.
    - MX: προσδιορίζει τους Mail Servers μιας ζώνης DNS.
    - PTR: χρησιμοποιούνται για την αντιστοίχιση μιας διεύθυνσης IP σε ένα FQDN, δηλαδή την αντίστροφη διαδικασία της επίλυσης ενός ονόματος.
    - SOA (Start Of Authority): περιλαμβάνει πληροφορίες σχετικές με το ποιος είναι ο Primary εξυπηρετητής της ζώνης, ποιος είναι ο διαχειριστής της ζώνης, πώς μπορεί κάποιος να επικοινωνήσει μαζί του, ποιο είναι το προκαθορισμένο TTL των εγγραφών και άλλες παραμέτρους που σχετίζονται με τις εγγραφές της ζώνης.
- CLASS: Η κλάση της εγγραφής. Συνήθως, έχει την τιμή IN για DNS records που σχετίζονται με ονόματα υπολογιστών στο διαδίκτυο, Servers και διευθύνσεις IP. Υπάρχουν και άλλες τιμές που δεν χρησιμοποιούνται (π.χ. CH).
- TTL: Ο χρόνος για τον οποίο η εγγραφή θεωρείται έγκυρη
- RDLENGTH: Το μήκος του πεδίο RDATA
- RDATA: Περιέχει την τιμή του Resource Record και επιπρόσθετες πληροφορίες της απάντησης όπως την σειρά προτεραιότητας των Mail Server ή τα ονόματα υπολογιστών σε εγγραφές τύπου MX.

2.1.4. Πακέτο DNS

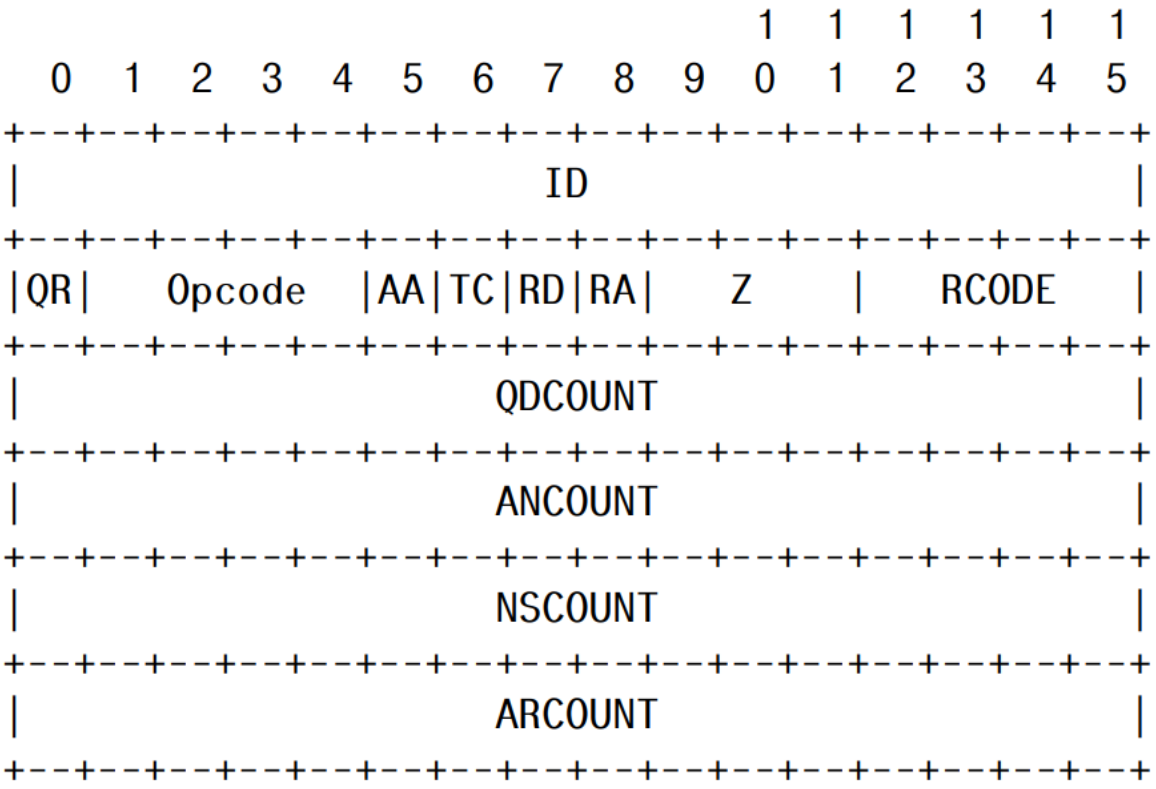
Το πρωτόκολλο DNS λειτουργεί χρησιμοποιώντας δύο τύπους μηνυμάτων, τα ερωτήματα (DNS queries) και τις απαντήσεις (DNS replies). Και οι δύο τύποι χρησιμοποιούν την ίδια δομή, η οποία φαίνεται στο παρακάτω σχήμα [7] :



Σχήμα 2.6 - Δομή πακέτων DNS.

Η επικεφαλίδα περιγράφει τον τύπο του πακέτου και ποια πεδία περιέχονται στο πακέτο. Στη συνέχεια ακολουθούν τα τμήματα ερωτήσεων, απαντήσεων, αρχής και των επιπρόσθετων στοιχείων.

Τα πακέτα είτε είναι ερωτήματα είτε απαντήσεις έχουν την ίδια δομή στις επικεφαλίδες τους η οποία είναι η εξής :



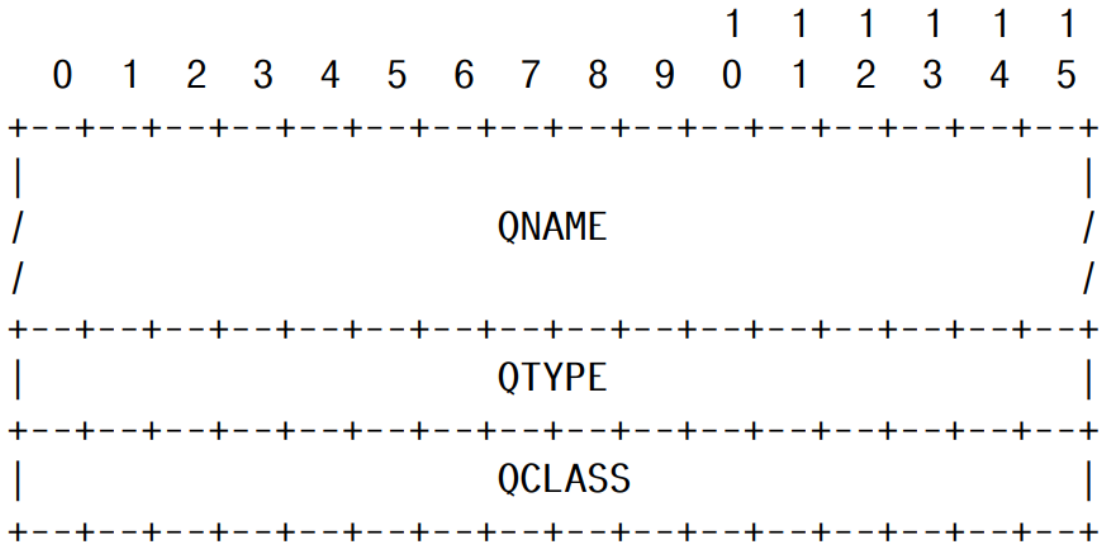
Σχήμα 2.7 - Επικεφαλίδα πακέτων DNS.

Κάθε ένα από τα πεδία περιγράφεται παρακάτω [7]:

- ID : Ένας αριθμός 16bit που χρησιμοποιείται για να γίνει το ταίριασμα ερωτήματος απάντησης.
- QR : Πεδίο 1bit που προσδιορίζει εάν το πακέτο είναι ερώτημα ή απάντηση. Στην πρώτη περίπτωση παίρνει την τιμή 0 ενώ στην δεύτερη 1.
- OPCODE : Πεδίο 4 bit που προσδιορίζει το είδος του ερωτήματος. Για τυπικά ερωτήματα χρησιμοποιείται η τιμή 0.
- AA : Απάντηση Αρχής (Authoritative Answer) - Αυτό το 1bit πεδίο έχει νόημα μόνο στα πακέτα απαντήσεων και δηλώνει ότι η απάντηση προέρχεται από έναν name Server ο οποίος είναι Authoritative για το Domain προς επίλυση.
- TC : Truncation - Δηλώνει ότι το μήνυμα έχει κοπεί (truncated).
- RD : Recursion Desired - Με την τιμή 1 σε αυτό το πεδίο, καθορίζεται αναδρομική επίλυση του ονόματος.
- RA : Recusrion Available - Αυτό το πεδίο παίρνει την τιμή 1 σε μία απάντηση, και δείχνει εάν η αναδρομική επίλυση ονόματος υποστηρίζεται από τον name Server.
- Z : Δεσμευμένο bit για μελλοντική χρήση. Παίρνει την τιμή 0.
- RCODE : Response code - Το συγκεκριμένο πεδίο έχει μέγεθος 4 bit και παίρνει τις εξής τιμές :
    - 0 - Δεν υπάρχει σφάλμα.
    - 1 - Σφάλμα διατύπωσης ερωτήματος

    - 2 - Σφάλμα εξυπηρετητή
    - 3 - Σφάλμα ονόματος(το Domain προς επίλυση δεν υπάρχει)
    - 4 - Ο name Server δεν υποστηρίζει το είδος ερωτήματος
    - 5 - Ο name Server δεν εκτελεί την λειτουργία που του ζητήθηκε.
- QDCOUNT : Ο αριθμός των ερωτημάτων στο τμήμα ερωτήσεων του πακέτου.
- ANCOUNT : Ο αριθμός εγγραφών (Resource Records) στο τμήμα απαντήσεων του πακέτου.
- NSCOUNT : Ο αριθμός εγγραφών name Server στο τμήμα αρχής του πακέτου.
- ARCOUNT : Ο αριθμός εγγραφών στο τμήμα επιπρόσθετων στοιχείων του πακέτου.

Το τμήμα ερωτήσεων ενός πακέτου ερωτήματος DNS έχει την παρακάτω δομή:

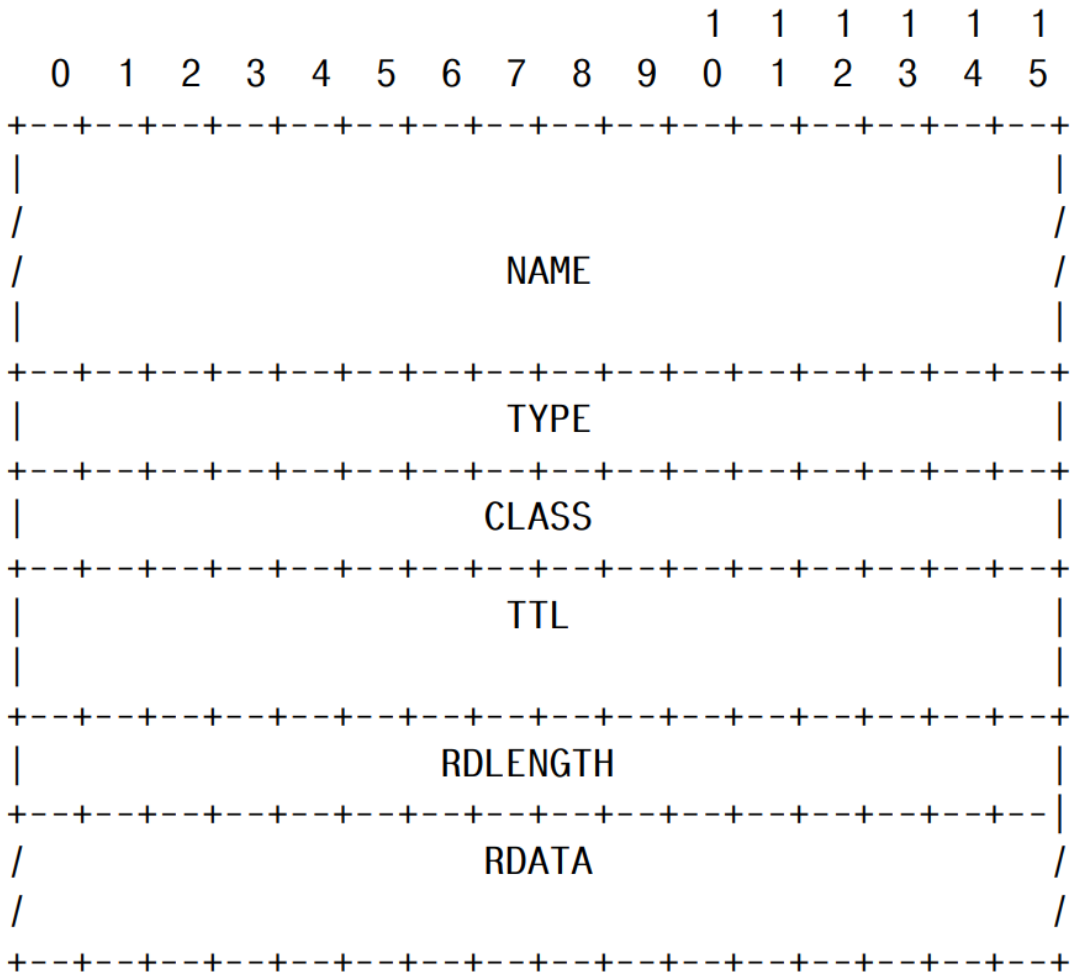


Σχήμα 2.8 - Τμήμα ερωτήσεων ενός πακέτου ερωτήματος DNS.

Κάθε ένα από τα πεδία περιγράφεται παρακάτω [7] :

- QNAME : Μία σειρά από ετικέτες, όπου κάθε μία αποτελείται από έναν αριθμό και στην συνέχεια μία λέξη με μήκος χαρακτήρων (bytes) όσο αυτός ο αριθμός. Η ένωση αυτών των λέξεων με τελείες δίνει το Domain name προς επίλυση. Ο αριθμός μηδέν ως μήκος ετικέτας σηματοδοτεί τον τερματισμό του Domain name.
- QTYPE : Ο τύπος του ερωτήματος (π.χ. A, AAAA, MX). Περιγράφεται με έναν κωδικό μεγέθους 2 byte.
- QCLASS : Η κλάση του ερωτήματος (π.χ. IN). Περιγράφεται με έναν κωδικό μεγέθους 2 byte.

Τέλος, το τμήμα απαντήσεων ενός πακέτου απάντησης DNS έχει την εξής δομή



Σχήμα 2.9 - Τμήμα απαντήσεων ενός πακέτου απάντησης DNS.

και ουσιαστικά πρόκειται για ένα (ή περισσότερα) Resource Record αφού αυτό είναι και η απάντηση σε ένα ερώτημα DNS. Τα πεδία περιγράφηκαν παραπάνω.

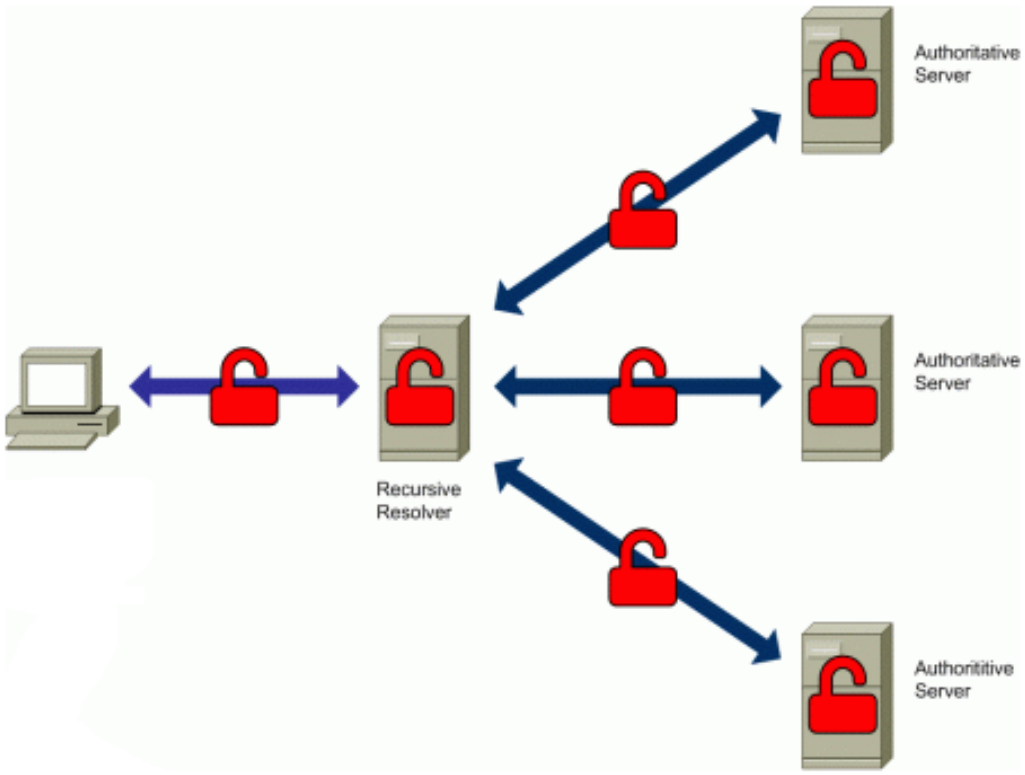
Τα τμήματα ερωτήσεων/απαντήσεων σε πακέτα ερώτησης/απάντησης DNS αντίστοιχα, έχουν μεταβλητό μέγεθος, ανάλογα με το πλήθος ερωτήσεων/απαντήσεων.

2.1.5. Θέματα ασφαλείας στο DNS

Οποιαδήποτε εφαρμογή λοιπόν χρησιμοποιεί το διαδίκτυο, χρειάζεται το DNS για να επικοινωνήσει με επιτυχία. Οι πληροφορίες που μεταφέρει το DNS σε κάθε εφαρμογή που τις χρειάζεται είναι δημόσιες, αφού όλοι μπορούν να έχουν

πρόσβαση σε αυτές, αλλά οι ενέργειες που εκτελεί ο κάθε χρήστης στο διαδίκτυο είναι ιδιωτικές και θα πρέπει να μην είναι προσβάσιμες από τρίτους. Το DNS ωστόσο δεν σχεδιάστηκε με κύριο γνώμονα την ιδιωτικότητα [8, 9] και την προστασία τέτοιων πληροφοριών και ως εκ τούτου προκύπτουν ορισμένα θέματα που έχουν σοβαρές επιπτώσεις στην ασφάλεια των χρηστών του διαδικτύου.

Στο παρακάτω σχήμα φαίνονται οι δύο κυριότεροι τύποι επικοινωνίας κατά την διάρκεια επίλυσης ονόματος.



Σχήμα 2.10 - Οι επικοινωνίες μεταξύ των υπολογιστών που συμμετέχουν στην επίλυση ενός ονόματος. Τα εικονίδια με τις ξεκλείδωτες κλειδαριές δείχνουν σε ποια σημεία μπορεί να συμβεί διαρροή πληροφορίας.

Αυτοί είναι :

1) η επικοινωνία μεταξύ του υπολογιστή που κάνει το αίτημα και του Resolver
2) η επικοινωνία μεταξύ του Resolver και των Authoritative Name Servers

Και στις δύο περιπτώσεις, η ανταλλαγή μηνυμάτων γίνεται χωρίς κρυπτογράφηση, δηλαδή δεν υπάρχει κάποιος μηχανισμός στο πρωτόκολλο του DNS που να αποτρέπει κάποιο κακόβουλο χρήστη από το να δει και να διαβάσει τα περιεχόμενα των ερωτημάτων και των απαντήσεων που στέλνονται. Επιπλέον, τα συστήματα που

επεξεργάζονται αυτά τα ερωτήματα, έχουν πρόσβαση σε αυτά αλλά και σε άλλες επιπρόσθετες πληροφορίες (π.χ. ποιος έκανε τα ερωτήματα) [8].

Παρατηρώντας την διαδικασία επίλυσης ονόματος προκύπτουν τα εξής σημεία στα οποία μπορεί υπάρξει διαρροή πληροφοριών :

- στην επικοινωνία μεταξύ του υπολογιστή και του Resolver
- στον Resolver
- στην επικοινωνία μεταξύ του Resolver και των Authoritative Servers
- στους Authoritative Servers

Οποιοσδήποτε έχει πρόσβαση στην επικοινωνία μεταξύ του υπολογιστή που υποβάλλει ένα ερώτημα για επίλυση ονόματος και του Resolver, μπορεί να δει τα μηνύματα που ανταλλάσσουν αλλά και να τα παραποιήσει. Το ίδιο μπορεί να συμβεί και στην επικοινωνία μεταξύ του Resolver και των Authoritative Servers.

Επίσης, είναι φανερό ότι τόσο ο Resolver όσο και οι Authoritative Servers μπορούν να καταγράφουν όλες τις ερωτήσεις και τις απαντήσεις που επεξεργάζονται. Για παράδειγμα ένας πάροχος διαδικτύου που προσφέρει την υπηρεσία του Resolver ή μία εταιρεία που λειτουργεί σαν open Resolver μπορεί να αποθηκεύει όλα τα DNS ερωτήματα που δέχεται και να χρησιμοποιεί αυτόν τον όγκο πληροφοριών για δικούς της σκοπούς (πώληση σε τρίτους για ανάλυση και εξαγωγή στατιστικών στοιχείων).

Η τοποθέτηση των Resolver παίζει ακόμη έναν ρόλο στην ιδιωτικότητα των χρηστών, καθώς όσο πιο κοντά είναι ο Resolver στον υπολογιστή που κάνει την ερώτηση τόσο πιο εύκολο είναι για τους Authoritative Servers να συνδέσουν ερωτήματα με χρήστες. Αν για παράδειγμα ο ίδιο ο υπολογιστής που στέλνει το ερώτημα λειτουργεί και σαν Resolver για τον εαυτό του , τότε οι Authoritative Servers μπορούν να δουν την διεύθυνση από την οποία ξεκίνησε το ερώτημα. Με την χρήση εξωτερικών Resolver η των open Resolvers δίνεται λύση σε αυτό το πρόβλημα , καθώς οι Authoritative Servers μπορούν να δουν την διεύθυνση μόνο των Resolvers που τους στέλνουν αιτήματα [8].

Ένα άλλο πρόβλημα που υπάρχει στο DNS είναι ότι τόσο οι απαντήσεις που δέχεται ένας υπολογιστής από έναν Resolver όσο και οι απαντήσεις που δέχεται ένας Resolver από τους Authoritative Servers δεν φέρουν κάποια εγγύηση σχετικά με την προέλευσή τους. Αυτό σημαίνει ότι ένας υπολογιστής ή ένας Resolver μπορούν να λάβουν μια κατασκευασμένη απάντηση και να την επεξεργαστούν σαν να ήταν η κανονική. Αυτό το είδος επίθεσης ονομάζεται Man In The Middle Attack [54] και θα αναλυθεί σε επόμενα κεφάλαια. Έχει σοβαρές συνέπειες για τους χρήστες καθώς μπορεί να οδηγήσει σε πλήρη έλεγχο ενός υπολογιστικού συστήματος.

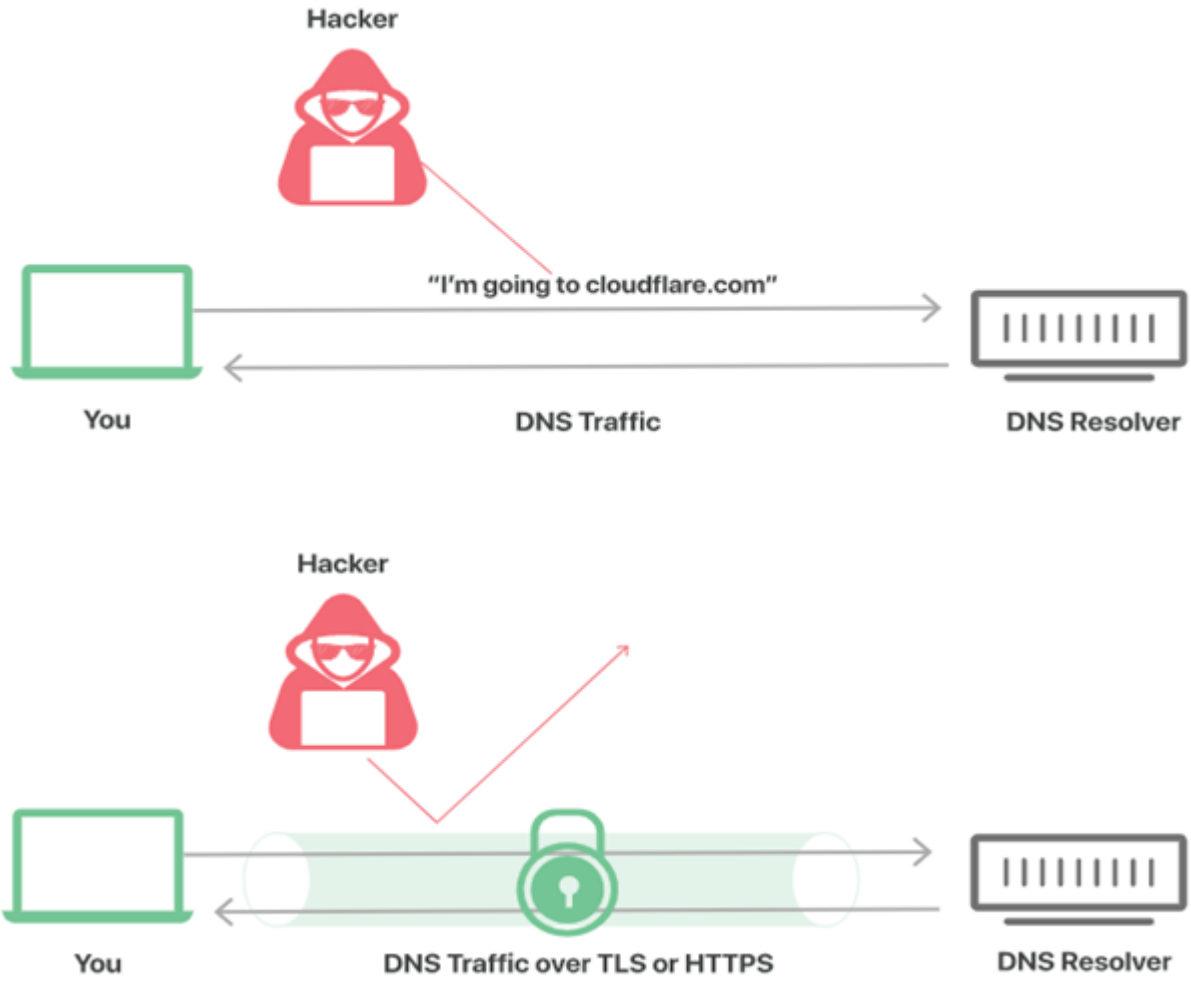
Λύσεις στα παραπάνω προβλήματα έρχονται να φέρουν ορισμένοι μηχανισμοί στο DNS.

Ένας από αυτούς είναι το DNS Security Extensions (DNSSEC) [10, 11]. Το DNSSEC είναι ένα πρωτόκολλο ασφάλειας το οποίο δημιουργήθηκε για να αντιμετωπίσει το πρόβλημα της έλλειψης πιστοποίησης προέλευσης των μηνυμάτων. Η λειτουργία του βασίζεται στην ψηφιακή υπογραφή των μηνυμάτων για να εξασφαλιστεί η εγκυρότητά τους.

Το DNSSEC υλοποιεί μια διαδικασία ιεραρχικής πολιτικής ψηφιακής υπογραφής των μηνυμάτων [10]. Προκειμένου μια επίλυση ονόματος να θεωρηθεί έγκυρη πρέπει τα μηνύματα να υπογράφονται σε κάθε επίπεδο. Για παράδειγμα, στην περίπτωση επίλυσης του ονόματος google.com ο Root DNS Server θα υπογράψει ένα κλειδί για τον .com nameserver και ο .com nameserver θα υπογράψει ένα κλειδί για τον Authoritative nameserver του google.com.

Ένας άλλος μηχανισμός που λύνει το πρόβλημα της διαρροής πληροφορίας στην επικοινωνία μεταξύ υπολογιστή και Resolver είναι το DoT (DNS over TLS) [12]. Το DoT χρησιμοποιεί το πρωτόκολλο TLS για να κρυπτογραφήσει τα μηνύματα που στέλνονται. Επομένως, ακόμη και αν κάποιος κακόβουλος χρήστης μπορούσε να δει τα πακέτα που ανταλλάσσει ένας υπολογιστής με έναν Resolver , δεν θα γινόταν να τα διαβάσει λόγω της κρυπτογράφησης τους. Το DoT χρησιμοποιεί το UDP port 853 σε αντίθεση με το απλό πρωτόκολλο του DNS που χρησιμοποιεί το port 53.

Παρόμοια λειτουργεί και το DoH (DNS over HTTPS), με την διαφορά ότι τα ερωτήματα στέλνονται μέσω του πρωτοκόλλου HTTP και του TCP port 443, με αποτέλεσμα να μην ξεχωρίζουν μέσα στην κίνηση HTTPS.



Σχήμα 2.11 - Ένας κακόβουλος χρήστης προσπαθεί να υποκλέψει πληροφορίες από την κίνηση DNS. Όταν χρησιμοποιείται το DoT ή το DoH η κίνηση είναι κρυπτογραφημένη.

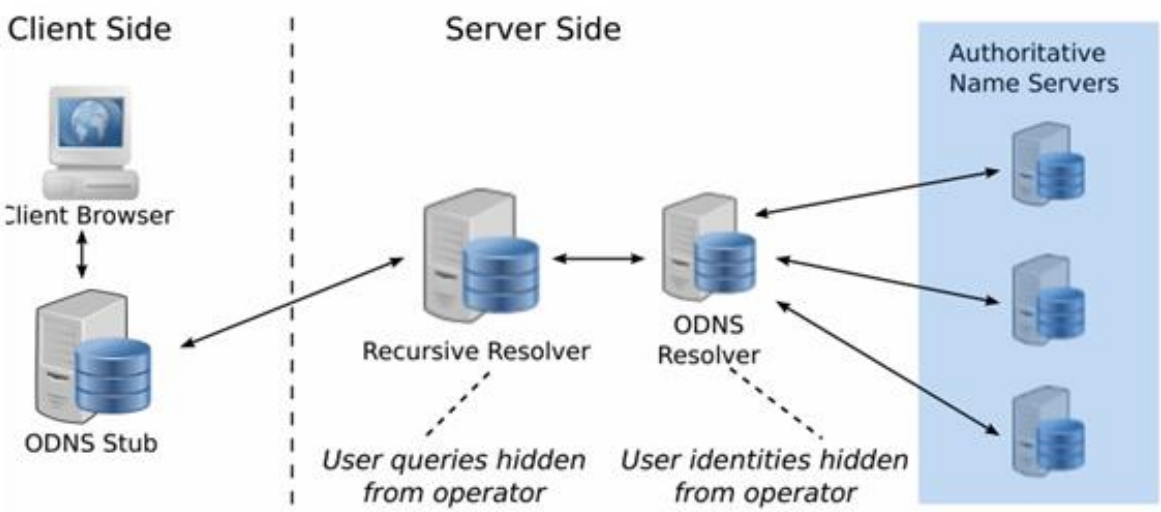
Ωστόσο, ακόμη και με την χρήση του DoT ή του DoH οι Resolver και οι Authoritative Servers εξακολουθούν να μπορούν να καταγράφουν τα ερωτήματα που τους γίνονται και να τα συνδέουν με χρήστες.

Για να αντιμετωπιστεί αυτό το πρόβλημα, κατασκευάστηκε το Oblivious DNS (ODNS) [13, 53], το οποίο είναι ένα νέο σχέδιο του οικοσυστήματος DNS που επιτρέπει στους τρέχοντες DNS εξυπηρετητές να παραμένουν αμετάβλητοι και συμβάλλει στην καλύτερη προστασία των δεδομένων.

Στο σύστημα ODNS, τροποποιείται ο υπολογιστής, που κάνει ένα αίτημα για επίλυση ονόματος (πελάτης), με ένα τοπικό πρόγραμμα επίλυσης αλλά υπάρχει και ένας νέος εξουσιοδοτημένος DNS Server για .odns [53]. Για να αποφευχθεί η διαρροή πληροφοριών από έναν κακόβουλο χρήστη, το ερώτημα DNS πρέπει να είναι κρυπτογραφημένο. Ο πελάτης αρχικά δημιουργεί ένα αίτημα, έστω για το

www.example.com, στην συνέχεια ένα κλειδί συνεδρίας k (session key), με το οποίο κρυπτογραφεί το Domain προς επίλυση και τέλος προσαρτά το TLD .odns. Έτσι προκύπτει το {www.example.com}(κρυπτογραφημένο με k).odns. Ο πελάτης το προωθεί σε έναν Resolver, συμπεριλαμβάνοντας το session key, κρυπτογραφημένο με το δημόσιο κλειδί του Authoritative Server του .odns Domain, στο τμήμα επιπρόσθετων στοιχείων του ερωτήματος DNS. Ο Resolver το προωθεί στον Authoritative dns Server για το .odns Domain ο οποίος αποκρυπτογραφεί το session key με το ιδιωτικό κλειδί του και με το session key το Domain που ζητήθηκε. Τέλος, ο Authoritative Server προωθεί το αίτημα DNS στον κατάλληλο name Server, ενεργώντας ως αναδρομικός αναλυτής. Για την απάντηση ακολουθείται η αντίστροφη πορεία με τις αντίστοιχες κρυπτογραφήσεις.

Οι name Servers βλέπουν εισερχόμενα αιτήματα DNS αλλά δεν γνωρίζουν από ποιους πελάτες προέρχονται και αντίστοιχα παρόλο που οι Resolvers βλέπουν από ποιους προέρχονται τα ερωτήματα, δεν γνωρίζουν το Domain για το οποίο γίνεται το αίτημα. Έτσι, κανείς δεν μπορεί να συνδέσει έναν πελάτη με τα αντίστοιχα ερωτήματα DNS.



Σχήμα 2.12 - Τρόπος λειτουργίας Oblivious DNS.

2.2. Δικτυακές επιθέσεις

Μία από τις μεγαλύτερες απειλές στο σύγχρονο διαδίκτυο είναι οι δικτυακές επιθέσεις, οι οποίες θα αναλυθούν σε αυτό το κεφάλαιο.

2.2.1. DoS and DDoS

Ως επίθεση άρνησης παροχής υπηρεσιών (Denial of Service, DoS) ορίζεται μία επίθεση που αποσκοπεί στο να αποκόψει νόμιμους χρήστες του διαδικτύου από ένα δικτυακό πόρο, προσωρινά ή και για μεγάλο χρονικό διάστημα, κάνοντας τον πόρο αυτό μη διαθέσιμο παρεμποδίζοντας την ομαλή του λειτουργία. Αυτές οι επιθέσεις λειτουργούν πλημμυρίζοντας το θύμα με πολύ μεγάλο όγκο κίνησης, στοχεύοντας στην κατασπατάληση των πόρων του (επεξεργαστική ισχύ, φυσική μνήμη, κλπ). Το θύμα καλείται να επεξεργαστεί όλα τα μηνύματα που δέχεται από τον επιτιθέμενο και αδυνατεί να εξυπηρετήσει τους πελάτες του.

Οι επιθέσεις DoS μπορούν να χωριστούν σε δύο κατηγορίες [14] :

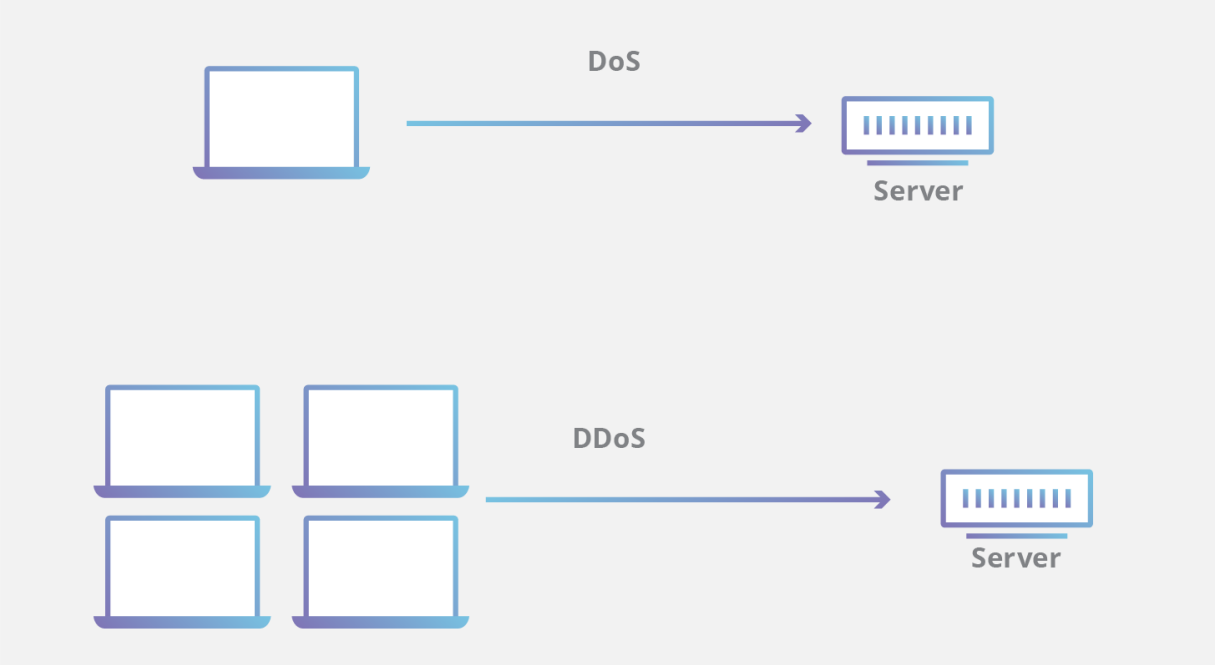
1) Buffer overflow attacks
2) Flood attacks

Στην πρώτη κατηγορία σκοπός του επιτιθέμενου είναι να καταναλώσει όλους τους διαθέσιμους πόρους του θύματος (αποθηκευτικός χώρος, μνήμη, επεξεργαστή). Αποτέλεσμα αυτού είναι τα θύματα να σταματούν να λειτουργούν σωστά και τελικά να διακόπτουν την λειτουργία τους.

Στην δεύτερη περίπτωση σκοπός είναι να υπερχειλίσει την χωρητικότητα του δικτύου. Αυτό επιτυγχάνεται με την αποστολή μεγάλου όγκου πακέτων οδηγώντας έτσι στην κατασπατάληση του εύρους ζώνης (bandwidth) των ζεύξεων του δικτύου του θύματος με αποτέλεσμα πακέτα νόμιμων χρηστών να απορρίπτονται και εν τέλει οι χρήστες να μην εξυπηρετούνται.

Συνεπώς, οι επιθέσεις DoS δε συνίστανται στην εγκατάσταση κακόβουλου κώδικα στο θύμα, αλλά στην εκμετάλλευση τρωτών σημείων των πρωτοκόλλων επικοινωνίας που χρησιμοποιούνται και στην αδύναμη υποδομή του θύματος (περιορισμένη φυσική μνήμη, επεξεργαστική ισχύς, ζεύξεις δικτύου κλπ).

Όταν η επίθεση δεν έχει σημείο εκκίνησης έναν υπολογιστή, αλλά πολλούς υπολογιστές, που βρίσκονται κατανεμημένοι σε διαφορετικά σημεία του διαδικτύου και προχωρούν σε συντονισμένη επίθεση, ονομάζεται κατανεμημένη επίθεση DoS (distributed DoS, DDoS). Οι επιθέσεις DDoS έχουν μεγάλη αποτελεσματικότητα καθώς εκμεταλλεύονται ευάλωτα υπολογιστικά συστήματα και τα αναγκάζουν να πάρουν μέρος στην επίθεση αυξάνοντας έτσι σε πολύ μεγάλο βαθμό τον όγκο της κακόβουλης κίνησης που καταλήγει στο θύμα.



Σχήμα 2.13 - Επιθέσεις DoS και DDoS.

2.2.2. Botnet

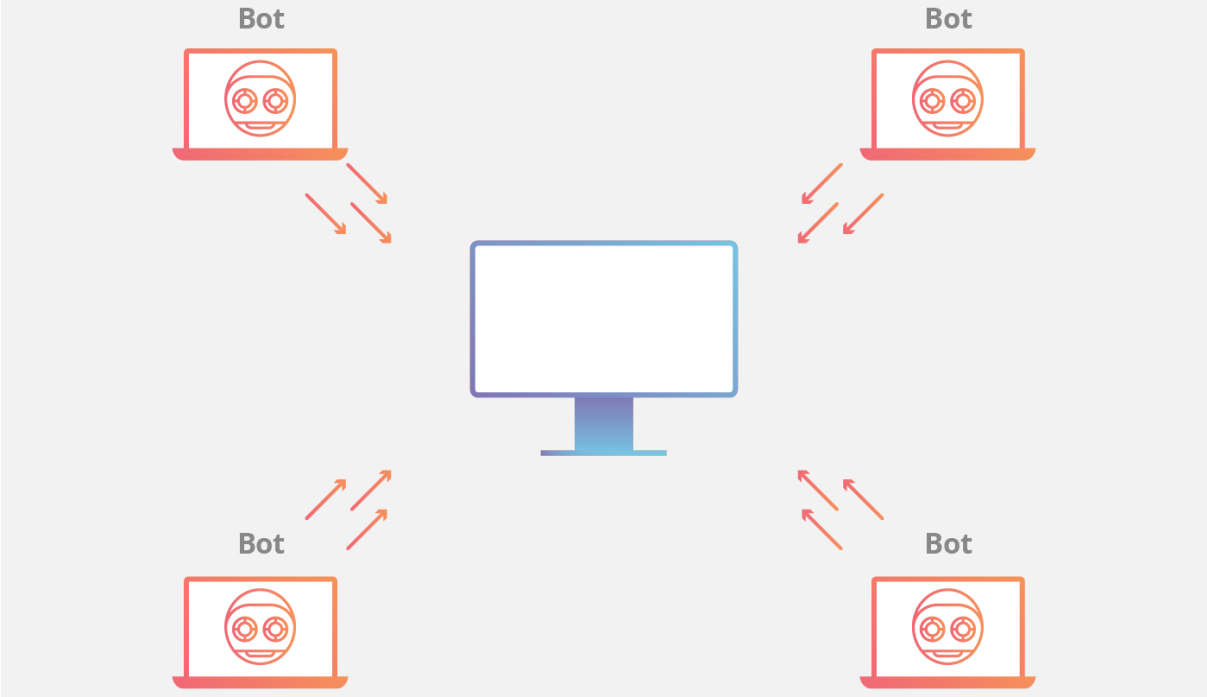
Ένας επιτιθέμενος μπορεί να έχει υπό τον έλεγχό του πολλά τέτοια υπολογιστικά συστήματα, δημιουργώντας έτσι ένα δίκτυο που εκτελεί πιστά τις εντολές του. Ένα τέτοιο δίκτυο ονομάζεται botnet και έχει καθοριστικό ρόλο στην ισχύ της επίθεσης. Ο ελεγκτής ενός botnet ονομάζεται botmaster.

Botnet ονομάζεται μία ομάδα από συστήματα με επεξεργαστική ισχύ τα οποία μπορούν να συνδέονται στο διαδίκτυο (υπολογιστές, συσκευές IoT), έχουν μολυνθεί από κάποιου είδους malware και είναι πλέον υπό τον έλεγχο ενός κακόβουλου χρήστη. Ο όρος botnet προήλθε από τον συνδυασμό των λέξεων *robot* και *network*, με κάθε μολυσμένη συσκευή να ονομάζεται bot. Τα δίκτυα αυτά μπορούν να σχεδιαστούν έτσι ώστε να εκτελέσουν διάφορες ενέργειες όπως αποστολή ανεπιθύμητων μηνυμάτων (spam), κλοπή δεδομένων, διασπορά ιών, και επιθέσεις DDoS [15].

Η επίδραση malware σε έναν υπολογιστή συνήθως έχει ορατά αποτελέσματα, όπως χαμηλή απόδοση αλλά ένα DDoS botnet malware μπορεί να λειτουργεί και διαφορετικά. Κάποια σχεδιάζονται έτσι ώστε να έχουν πλήρη έλεγχο του συστήματος που μολύνουν ενώ άλλα παραμένουν αδρανή μέχρι να λάβουν κάποια εντολή.

Τα botnet μπορούν και αυξάνουν τον αριθμό των bots τους εκμεταλλεύοντας αδυναμίες σε ιστοσελίδες, μολύνοντας με ιούς (Trojan) άλλους υπολογιστές ή αποκτώντας πρόσβαση σε συστήματα με αδύναμη προστασία. Από την στιγμή που

ένα σύστημα μολυνθεί μπορεί να ξεκινήσει να μολύνει άλλα συστήματα στο δίκτυό του. Εκτιμήσεις σχετικά με τον αριθμό των bots σε ένα botnet κυμαίνονται από μερικές χιλιάδες έως και πάνω ένα εκατομμύριο.



Σχήμα 2.14 - Μόλυνση ενός υπολογιστή από άλλα bots.

Τα botnets δημιουργούνται για διάφορους λόγους, από ακτιβισμό μέχρι και για χρήση από κρατικούς οργανισμούς. Η ενοικίαση ενός botnet είναι σχετικά φθηνή σχετικά με την ζημιά που μπορούν να προκαλέσουν [15].

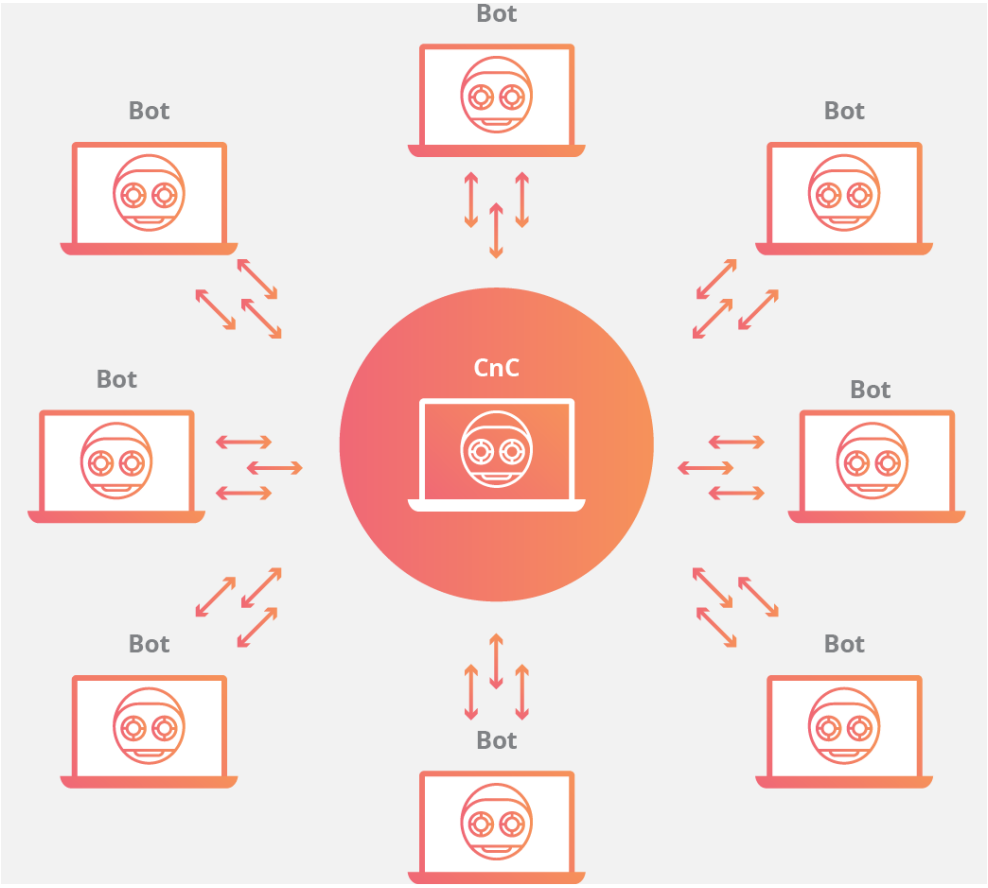
Το κύριο χαρακτηριστικό των botnets είναι η ικανότητα να λαμβάνουν ενημερώσεις και εντολές από τον ελεγκτή τους. Αυτή η ικανότητα των bots επιτρέπει στον botmaster να αλλάζει απομακρυσμένα τον τρόπο και το στόχο της επίθεσης, να σταματάει την επίθεση ή να εκτελεί άλλες ενέργειες.

Η γενική δομή του τρόπου επικοινωνίας των bots με τους ελεγκτές τους περιγράφεται από δύο μοντέλα [15] :

1) client/Server botnet model
2) peer-to-peer botnet model

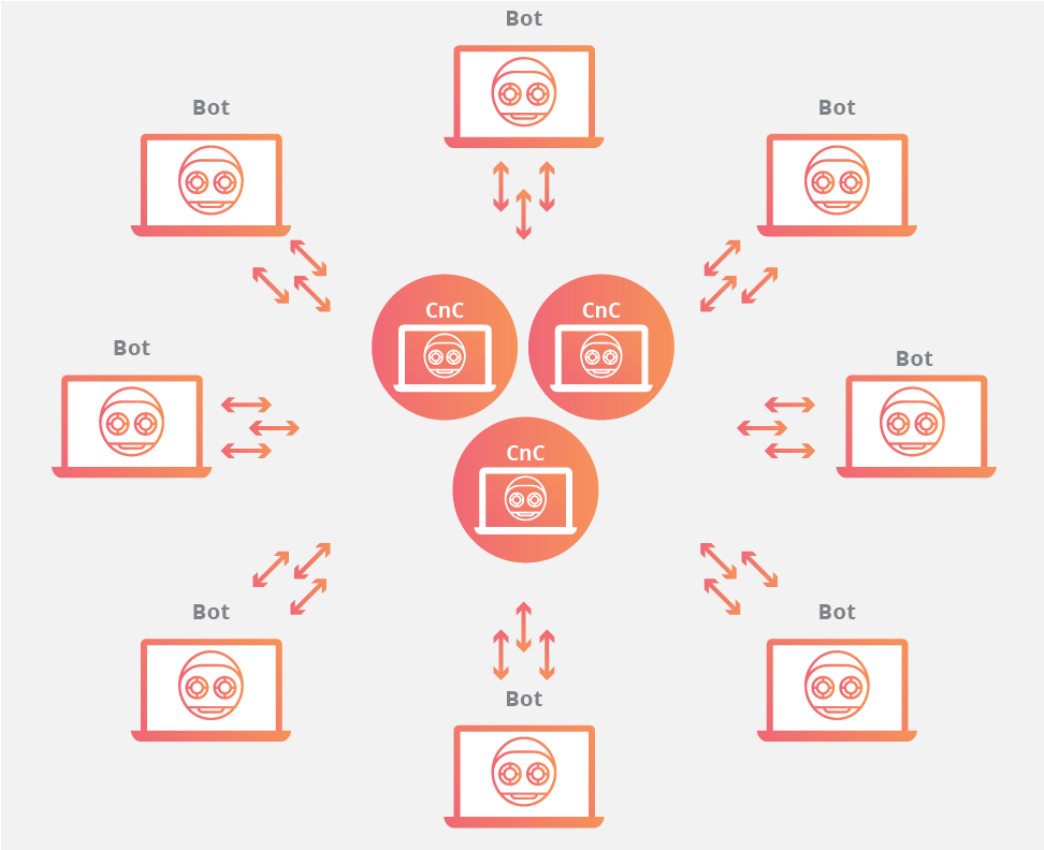
Το πρώτο μοντέλο έχει την κλασική δομή όπου κάθε συσκευή συνδέεται σε έναν κεντρικό Server έτσι ώστε να λάβει πληροφορίες. Αυτοί οι κεντρικοί Servers που στέλνουν τις πληροφορίες στα bots ονομάζονται Command and Control Servers (CnC). Οι πιο γνωστές τοπολογίες αυτού του μοντέλου είναι οι εξής:

1) Τοπολογία Αστέρα



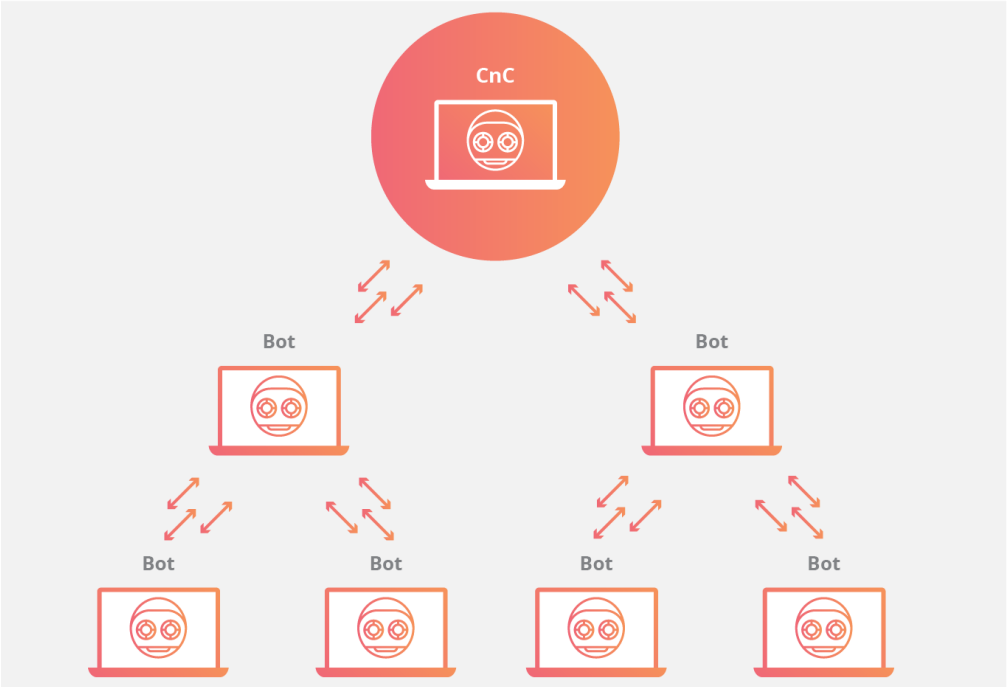
Σχήμα 2.15 - Botnet με τοπολογία Αστέρα.

2) Τοπολογία πολλαπλών Server



Σχήμα 2.16 - Botnet με τοπολογία πολλαπλών Server.

3) Ιεραρχική τοπολογία

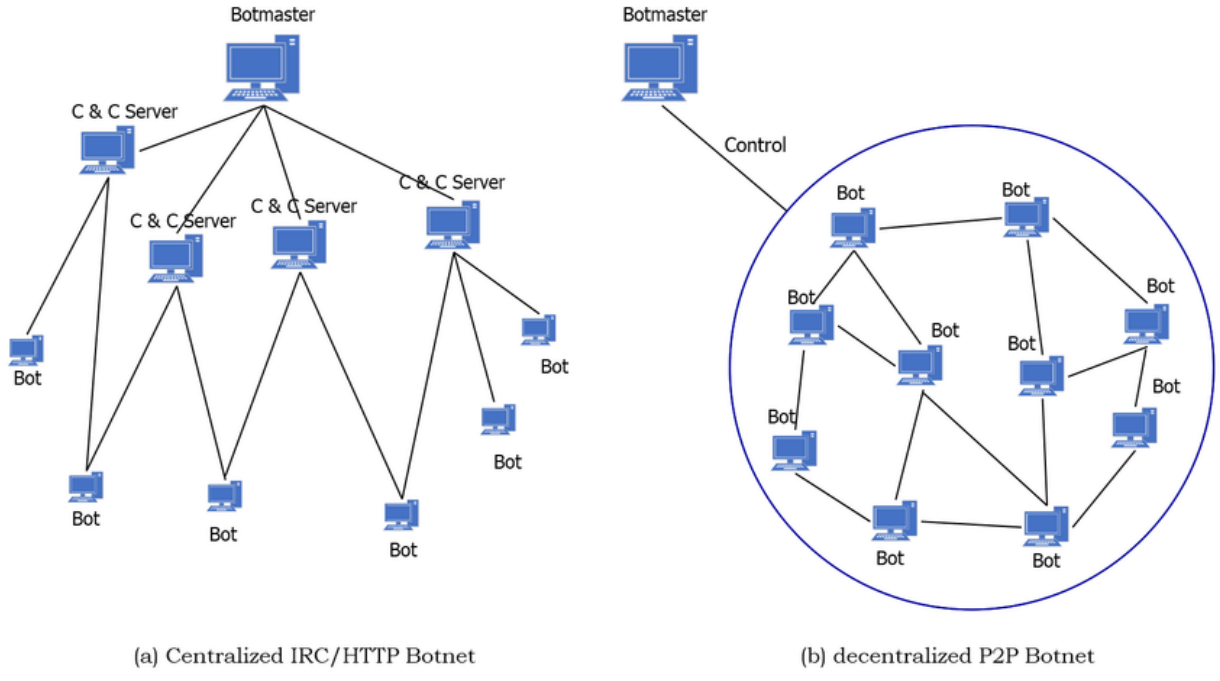


Σχήμα 2.17 - Botnet με ιεραρχική τοπολογία.

Η βασική αδυναμία αυτού του μοντέλου είναι ότι εάν εντοπιστεί και αφαιρεθεί ο CnC Server ή αποκοπεί η επικοινωνία του με τα bots, αυτά δεν θα μπορούν να λάβουν νέες οδηγίες. Γι' αυτό οι δημιουργοί των botnets ανέπτυξαν το peer-to-peer μοντέλο.

Ένα δίκτυο υπολογιστών peer-to-peer (ή P2P) είναι ένα δίκτυο που επιτρέπει σε δύο ή περισσότερους υπολογιστές να μοιράζονται τους πόρους τους ισοδύναμα [55]. Το δίκτυο αυτό χρησιμοποιεί την επεξεργαστική ισχύ, τον αποθηκευτικό χώρο και το εύρος ζώνης (bandwidth) των κόμβων. Όλοι οι κόμβοι του δικτύου έχουν ίσα δικαιώματα. Πληροφορίες που βρίσκονται στον ένα κόμβο, ανάλογα με τα δικαιώματα που καθορίζονται, μπορούν να διαβαστούν από όλους τους άλλους και αντίστροφα.

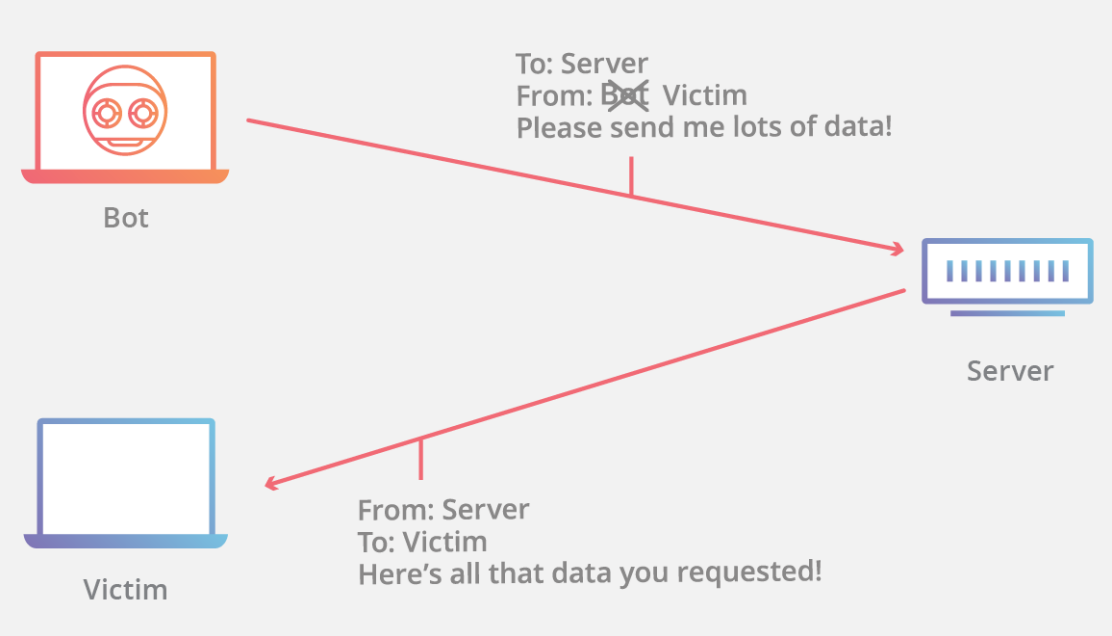
Έτσι, ενσωματώνεται η δομή ελέγχου μέσα στο ίδιο το botnet και αφαιρείται το πρόβλημα του μοντέλου πελάτη-εξυπηρετητή. Τα bots ενός P2P δικτύου μπορούν να είναι ταυτόχρονα και πελάτες και CnC Servers. Ακόμη, έχουν αποθηκευμένη μία λίστα από bots που εμπιστεύονται για να επικοινωνούν με αυτά και να λαμβάνουν ενημερώσεις και εντολές. Λόγω της απουσίας κάποιας κεντρικής αρχής το botnet μπορεί πιο εύκολα να ελεγχθεί από κάποιον εντός του δικτύου παρά από τον δημιουργό του. Γι' αυτό συνήθως τέτοιου τύπου botnets χρησιμοποιούν κρυπτογράφηση έτσι ώστε η πρόσβαση από τρίτους να αποκόπτεται.



Σχήμα 2.18 - Centralized vs P2P botnets.

2.2.3. IP spoofing

Μία τεχνική που χρησιμοποιείται στις επιθέσεις DDoS ονομάζεται IP Spoofing [17]. Με αυτή την τεχνική ο επιτιθέμενος έχει την δυνατότητα να αποκρύψει την πραγματική του διεύθυνση IP και ταυτόχρονα να την αλλάζει με κάποια άλλη. Αυτό συμβαίνει διότι ο τρόπος λειτουργίας του πρωτοκόλλου IP δεν περιέχει κάποιο έλεγχο ως προς την ταυτότητα του αποστολέα ενός πακέτου. Κάθε πακέτο IP περιέχει μια επικεφαλίδα στην οποία αναφέρονται πληροφορίες σχετικά με τον αποστολέα και τον παραλήπτη. Αλλάζοντας την IP του αποστολέα στο πακέτο ένας κακόβουλος χρήστης μπορεί να αποκρύψει την δικιά του διεύθυνση, να μιμηθεί κάποιον άλλο νόμιμο χρήστη ή όπως θα δούμε και σε επόμενα κεφάλαια να ανακατευθύνει ανεπιθύμητη κίνηση προς ένα θύμα.



Σχήμα 2.19 - Αλλαγή διεύθυνσης αποστολέα (IP spoofing).

2.2.4. Κίνητρα και Συνέπειες

Η πλειονότητα των κινήτρων των DDoS επιθέσεων ανήκουν στις παρακάτω κατηγορίες [18] :

- **Οικονομικό όφελος.** Οι επιθέσεις DDoS εναντίον ιστότοπων και τραπεζών ηλεκτρονικού εμπορίου είναι μια αυξανόμενη τάση. Ακόμη, μία επίθεση μπορεί να ως στόχο τον εκβιασμό ενός χρήστη ή ενός οργανισμού. Είναι μία συνηθισμένη τεχνική για κακόβουλους χρήστες οι οποίοι απαιτούν πληρωμή, συνήθως μέσω κρυπτονομισμάτων (Bitcoin) για να σταματήσουν την επίθεση.
- **Ιδεολογική πίστη.** Ορισμένοι επιτιθέμενοι έχουν κίνητρα για να επιτεθούν σε πολιτικούς στόχους λόγω των ιδεολογικών πεποιθήσεών τους εναντίον

πολιτικών εθνικού κράτους ή κυβέρνησης. Είναι επίσης γνωστό και ως χακτιβισμός (hacktivism).

- **Διανοητική πρόκληση.** Μερικοί επιτιθέμενοι στοχεύουν ιστότοπους για να αποδείξουν τις τεχνικές τους ικανότητες. Πειραματίζονται με τις τελευταίες τεχνολογίες και χρησιμοποιούν τις γνώσεις τους πολλές φορές για επίδειξη δύναμης.
- **Προσωπική απόλαυση.** Το συγκεκριμένο κίνητρο ανήκει στην κατηγορία του διαδικτυακού εκφοβισμού. Ο επιτιθέμενος έχει σαν σκοπό η επίθεσή του να είναι κάτι είτε διασκεδαστικό είτε εκδικητικό.
- **Κυβερνοπόλεμος.** Ο κυβερνοπόλεμος συνδέεται συνήθως με εθνικά κράτη, και χρησιμοποιείται για πολιτικό και στρατιωτικό πλεονέκτημα. Μία επίθεση με αυτό το κίνητρο εκτελείται για να επιφέρει οικονομικές ή φυσικές επιπτώσεις στους στόχους της. Οι ομάδες που χρησιμοποιούν στρατηγικές και τακτικές πολέμου στον κυβερνοχώρο είναι καλά εκπαιδευμένες, οργανωμένες και ανήκουν σε κυβερνητικούς στρατούς ή τρομοκρατικές οργανώσεις.

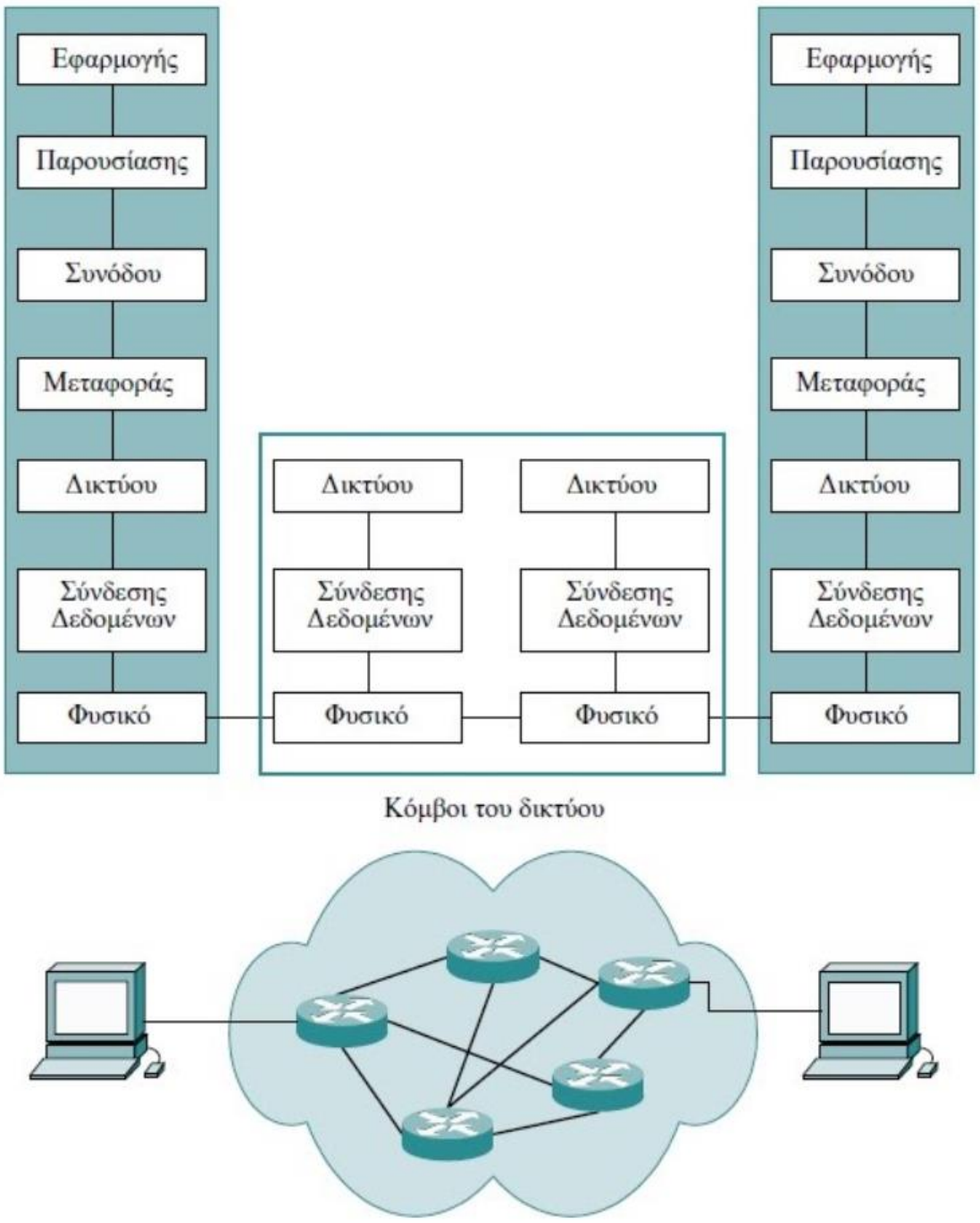
Οι συνέπειες μιας επιτυχημένης επίθεσης DoS/DDoS αφορούν κυρίως άμεσες ή έμμεσες οικονομικές επιβαρύνσεις. Τα έσοδα μια επιχείρησης μπορεί να συνδέονται άμεσα με τις δικτυακές τις υποδομές και όταν αυτές δέχονται επιθέσεις, με αποτέλεσμα να μην μπορούν να λειτουργήσουν σωστά, αυτό μεταφράζεται σε ζημία για την επιχείρηση αυτή. Ωστόσο, ακόμη και αν το αντίκτυπο μιας επίθεσης δεν είναι τόσο άμεσο για το θύμα, η ενίσχυση της άμυνας των συστημάτων που δέχτηκαν την επίθεση καθώς και το πλήγμα στην αξιοπιστία και στην εικόνα μια επιχείρησης αποτελούν εξίσου σοβαρές συνέπειες. Τέλος, η εργασία στην σύγχρονη εποχή απαιτεί την πρόσβαση των εργαζομένων στο διαδίκτυο, σε απομακρυσμένους εξυπηρετητές, σε υπηρεσίες σύννεφου (Cloud) και άλλες δικτυακές υπηρεσίες. Επομένως, μία DDoS επίθεση εμποδίζει και την παραγωγικότητα μιας εταιρείας.

2.2.5. OSI model & είδη DDoS

Οι επιθέσεις DDoS ποικίλουν ως προς τον τρόπο εκτέλεσής τους αλλά και ως προς τον τελικό τους στόχο. Για να γίνουν καλύτερα κατανοητές ακολουθεί μία σύντομη αναφορά στο μοντέλο OSI.

Το μοντέλο OSI [19, 20] βασίζεται σε μια πρόταση, που ανέπτυξε ο Οργανισμός Διεθνών Προτύπων ISO, ως ένα πρώτο βήμα προς την κατεύθυνση της διεθνούς προτυποποίησης των πρωτοκόλλων που χρησιμοποιούνται στα διάφορα στρώματα. Το μοντέλο αποκαλείται μοντέλο αναφοράς OSI (Open Systems Interconnection) του ISO, επειδή αφορά ανοικτά συστήματα, δηλαδή συστήματα ανοικτά στην επικοινωνία

με άλλα συστήματα. Το μοντέλο αυτό έχει επτά στρώματα καθένα από τα οποία εκτελεί συγκεκριμένες λειτουργίες και επικοινωνεί με τα επίπεδα που είναι ακριβώς από πάνω και από κάτω του. Τα τέσσερα χαμηλότερα επίπεδα ασχολούνται με τον έλεγχο της μετάδοσης των μηνυμάτων μέσα στο δίκτυο, ενώ τα τρία ανώτερα επίπεδα παρέχουν την αξιόπιστη μεταβίβαση των δεδομένων μεταξύ των τελικών χρηστών. Έτσι, και τα επτά επίπεδα υλοποιούνται μόνο στους υπολογιστές που λειτουργούν ως τερματικοί σταθμοί.



Σχήμα 2.20 - Μοντέλο OSI.

Αναλυτικότερα για το κάθε επίπεδο:

- Φυσικό (Physical) : Στο φυσικό επίπεδο καθορίζονται οι ηλεκτρικές, μηχανικές και λειτουργικές προδιαγραφές για τη μετάδοση των δεδομένων πάνω από ένα φυσικό μέσο(π.χ. η οπτική ίνα, το ομοαξονικό καλώδιο, η μικροκυματική ζεύξη κ.ά.). Στο επίπεδο αυτό τα δεδομένα γίνονται αντιληπτά ως μια «ακατέργαστη» ακολουθία bits και μόνο.
- Σύνδεσης Δεδομένων (Data Link): Το επίπεδο σύνδεσης δεδομένων παρέχει τα λειτουργικά και διαδικαστικά μέσα για τη μεταφορά δεδομένων από μια συσκευή ενός τοπικού δικτύου σε άλλη, αλλά και για την ανίχνευση και διόρθωση σφαλμάτων που συμβαίνουν στο φυσικό επίπεδο. Οι μη ιεραρχημένες διευθύνσεις των συσκευών εδώ είναι οι φυσικές (π.χ. MAC διευθύνσεις), δηλαδή είναι προκαθορισμένες και αποθηκευμένες στις κάρτες δικτύου των επικοινωνούντων κόμβων από το εργοστάσιο. Το πιο γνωστό πρότυπο αυτού του επιπέδου είναι το Ethernet, για τοπικά δίκτυα.
- Δικτύου (Network): Οι μονάδες δεδομένων που ανταλλάσσουν οι ομότιμες διεργασίες στο επίπεδο δικτύου καλούνται πακέτα. Στο Επίπεδο Δικτύου καθορίζεται ο τρόπος δρομολόγησης των πακέτων από τον αποστολέα στον παραλήπτη. Σε αυτό το επίπεδο υλοποιείται το σχήμα διευθυνσιοδότησης του δικτύου. Κάθε κόμβος που ανήκει σε ένα δίκτυο χαρακτηρίζεται μοναδικά από τη διεύθυνση δικτύου. Η δρομολόγηση των πακέτων γίνεται με βάση τη διεύθυνση δικτύου του παραλήπτη κόμβου.
- Μεταφοράς (Transport): Το επίπεδο μεταφοράς διεκπεραιώνει τη μεταφορά των δεδομένων από χρήστη σε χρήστη, απαλλάσσοντας έτσι τα ανώτερα επίπεδα από κάθε φροντίδα να προσφέρουν αξιόπιστη μεταφορά δεδομένων από το ένα άκρο της επικοινωνίας στο άλλο. Το επίπεδο μεταφοράς ελέγχει την αξιοπιστία ενός χρησιμοποιούμενου καναλιού με έλεγχο ροής, κατάτμηση και έλεγχο σφαλμάτων. Τα πιο γνωστά πρωτόκολλα μεταφοράς είναι το TCP και το UDP.
- Συνόδου (Session): Σε αυτό το επίπεδο διενεργούνται όλες οι απαραίτητες λειτουργίες για την εγκαθίδρυση, την επίβλεψη και τον τερματισμό των συνόδων (sessions) μεταξύ των τελικών εφαρμογών.
- Παρουσίασης (Presentation): Το επίπεδο παρουσίασης μετασχηματίζει τα δεδομένα σε τυπική μορφή που την αναμένει το επίπεδο εφαρμογών. Στο επίπεδο αυτό τα δεδομένα υφίστανται κρυπτογράφηση, συμπίεση, κωδικοποίηση MIME και όποια άλλη διαμόρφωση απαιτεί η μορφή δεδομένων ή ο σχεδιαστής του πρωτοκόλλου.
- Εφαρμογής (Application): Το επίπεδο εφαρμογών παρέχει στον χρήστη έναν τρόπο να προσπελάσει μέσω μιας εφαρμογής τις πληροφορίες ενός δικτύου. Αυτό το επίπεδο είναι η κύρια διασύνδεση του χρήστη με την εφαρμογή και, συνεπώς, με το δίκτυο. Παραδείγματα πρωτοκόλλων επιπέδου εφαρμογών αποτελούν τα Telnet, FTP, SMTP και http.

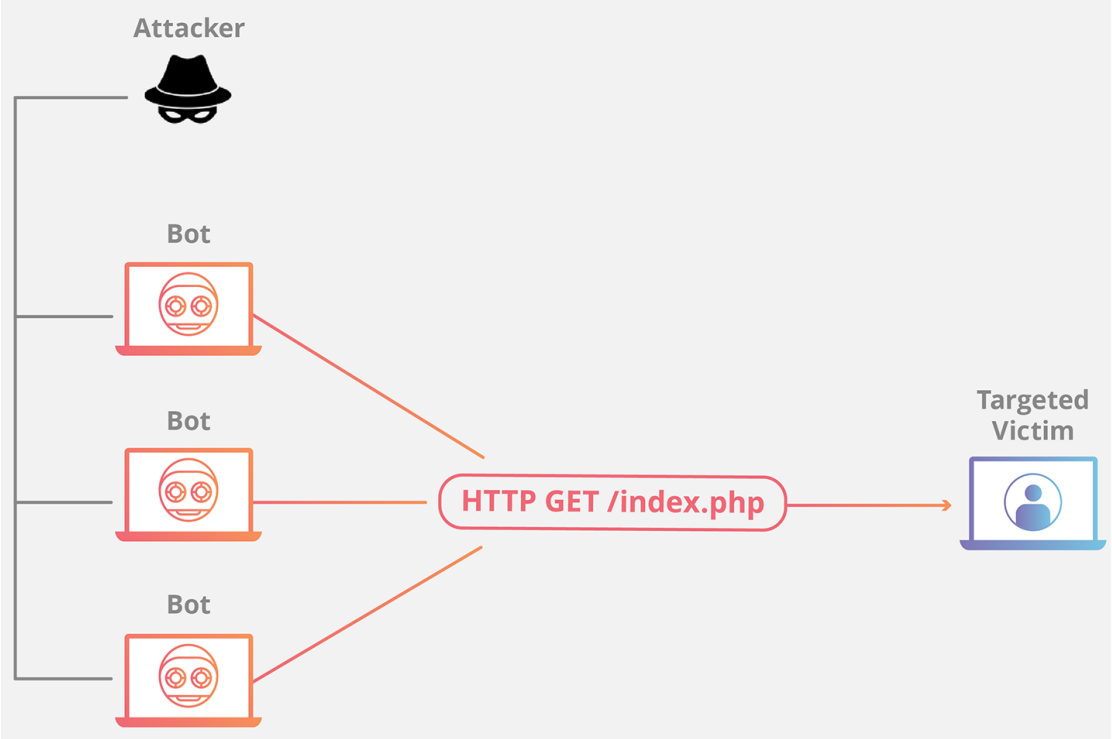
Οι επιθέσεις DDoS μπορούν να χωριστούν σε τρεις βασικές κατηγορίες [21], αν και σχεδόν όλες έχουν ως τελικό σκοπό να πλημμυρίσουν το θύμα με κίνηση που δεν μπορεί να διαχειριστεί. Ο επιτιθέμενος μπορεί να χρησιμοποιήσει έναν ή περισσότερους τρόπους επίθεσης, καθώς και να τους αλλάζει ανάλογα με τα μέτρα προστασίας που λαμβάνονται από το θύμα.

Οι κατηγορίες αυτές είναι:

1) Επιθέσεις στο επίπεδο εφαρμογής (application attacks) : Σκοπός αυτών των επιθέσεων είναι να εξαντλήσουν τους πόρους του θύματος για να προκαλέσουν άρνηση παροχής υπηρεσιών. Στόχος τους είναι το στρώμα του OSI στο οποίο παράγονται οι σελίδες του διαδικτύου από εξυπηρετητές, δηλαδή το στρώμα εφαρμογής (application layer). Μία σελίδα είναι εύκολο, από πλευράς επεξεργαστικού φόρτου, να ζητηθεί από έναν χρήστη αλλά μπορεί να είναι πιο δύσκολο για έναν Server ο οποίος μπορεί να χρειάζεται να φορτώσει αρχεία ή να λάβει πληροφορίες από κάποια βάση δεδομένων.

   Σε μία τέτοια επίθεση σκοπός του επιτιθέμενου είναι να υπερφορτώσει τον εξυπηρετητή, που διαχειρίζεται μία εφαρμογή σε μία ιστοσελίδα, με HTTP αιτήματα. Μπορεί να στοχεύσει μία ή και πολλές σελίδες τις εφαρμογής.

   Η συγκεκριμένη επίθεση είναι δύσκολο να αντιμετωπιστεί καθώς η κίνηση ενός κακόβουλου χρήστη δεν διαφέρει κατά πολύ από αυτή ενός νόμιμου χρήστη. Οι επιθέσεις αυτές μετριούνται σε requests/sec.

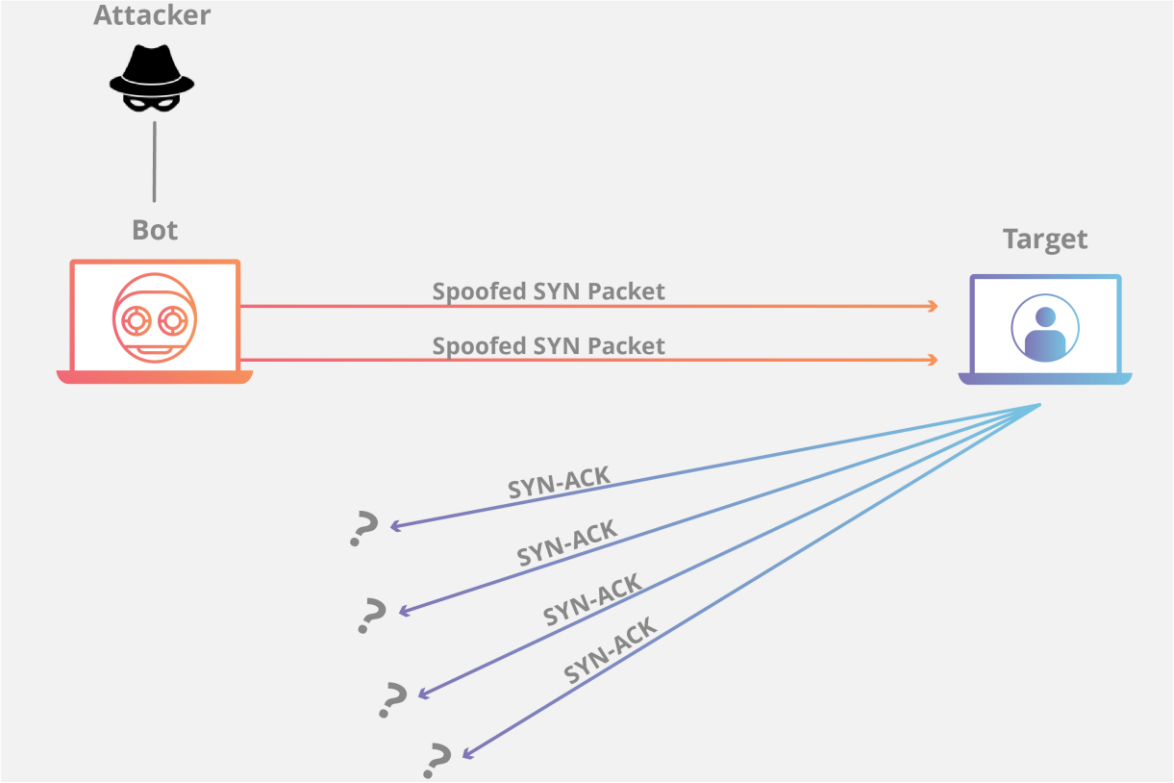


Σχήμα 2.21 - Επίθεση στο επίπεδο εφαρμογής με κατακλυσμό HTTP GET requests.

2) Επιθέσεις στο πρωτόκολλο (protocol attacks) :

Τέτοιες επιθέσεις έχουν ως στόχο την κατασπατάληση υπολογιστικών πόρων, όπως είναι η χρήση του επεξεργαστή, η διαθέσιμη μνήμη, ο αριθμός των διαθέσιμων sockets, ο αριθμός προσβάσεων στο δίσκο, κ.λπ.. Εκμεταλλεύονται αδυναμίες στα επίπεδα 3 (δικτύου) και 4 (μεταφοράς) του OSI. Οι επιθέσεις αυτές μετριούνται, συνήθως, σε packets/sec.

Μία γνωστή επίθεση πρωτοκόλλου είναι η SYN flood. Χρησιμοποιεί την τριμερή χειραψία του TCP, η οποία πραγματοποιείται πριν από οποιαδήποτε σύνδεση TCP μεταξύ δύο υπολογιστών. Στέλνοντας πολλά πακέτα αίτησης αρχικής σύνδεσης (SYN) με spoofed IP διεύθυνση, το θύμα αναγκάζεται να απαντήσει σε κάθε ένα από αυτά και στη συνέχεια περιμένει για το τελευταίο βήμα της σύνδεσης, κάτι το οποίο δεν συμβαίνει ποτέ. Αποτέλεσμα αυτού είναι το θύμα να εξαντλήσει τον αριθμό διαθέσιμων sockets και να μην μπορεί να δεχθεί συνδέσεις από νόμιμους χρήστες.



Σχήμα 2.22 - Επίθεση SYN flood.

3) Επιθέσεις κατά του εύρους ζώνης (volumetric/bandwidth attacks) :

   Έχουν στόχο να κατασπαταλήσουν το εύρος ζώνης του δικτύου στο οποίο βρίσκεται το θύμα, δηλαδή να προκαλέσουν συμφόρηση (congestion). Έτσι, θα εμποδίσουν τη μεταφορά μηνυμάτων από και προς τον εξυπηρετητή, καθώς θα απορρίπτονται λόγω των πλημμυρισμένων ζεύξεων. Οι επιθέσεις αυτές μετριούνται, συνήθως, σε bits/sec. Μία τέτοια επίθεση είναι DNS Amplification, η οποία θα αναλυθεί στο επόμενο κεφάλαιο.

2.2.6. Επιθέσεις DNS

Σε αυτό το κεφάλαιο θα αναλυθούν οι πιο σημαντικές επιθέσεις στο DNS.

DNS flood

Πρόκειται για μία επίθεση DDoS, με θύματα έναν ή περισσότερους Authoritative εξυπηρετητές μιας ζώνης DNS και αποτελεί μια παραλλαγή της πλημμύρας UDP (UDP flood) κατά της υπηρεσίας.

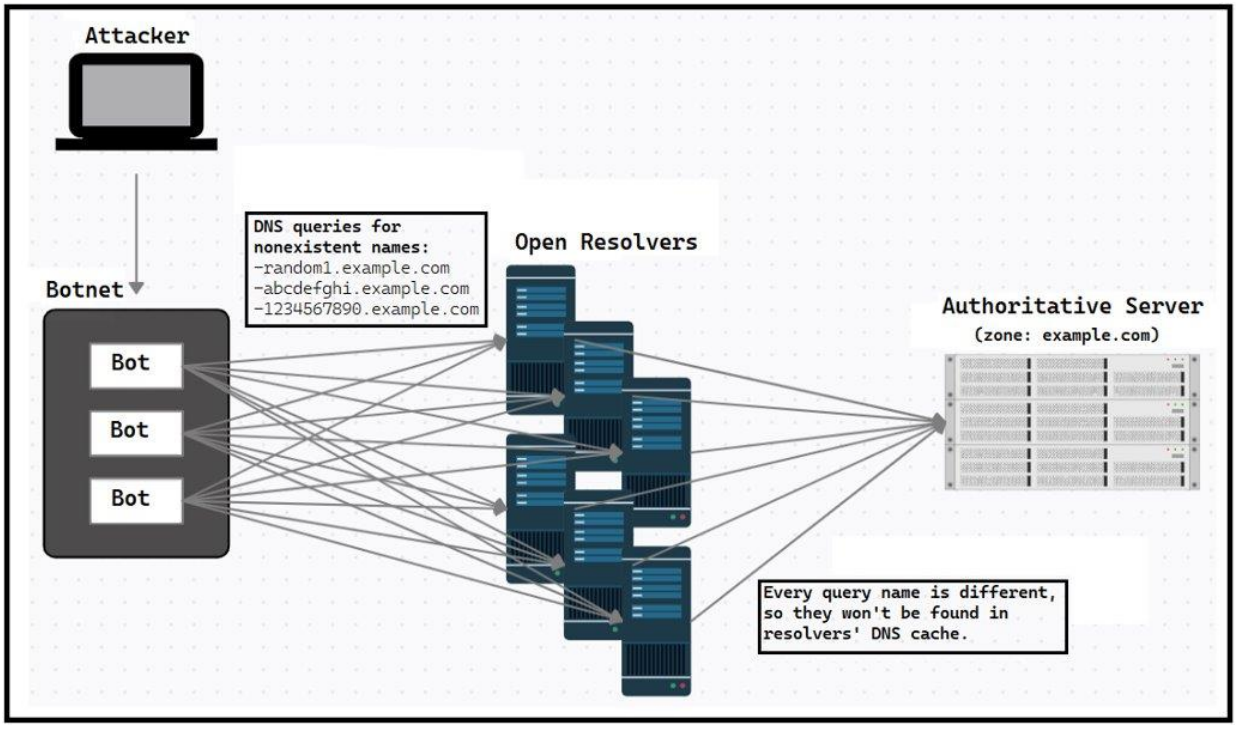
Ο επιτιθέμενος αποκρύπτοντας την IP του με την τεχνική του IP Spoofing, που προαναφέρθηκε, δίνει εντολές σε ένα botnet που ελέγχει και το προγραμματίζει να στέλνει μεγάλο όγκο κίνησης DNS που αποτελείται είτε από έγκυρα είτε από άκυρα ερωτήματα DNS. Αυτό έχει σαν αποτέλεσμα να εξαντλείται το εύρος ζώνης του δικτύου του εξυπηρετητή και να μονοπωλούνται οι πόροι του. Έτσι, η απόδοση του σταδιακά μειώνεται και αδυνατεί να απαντήσει σε μηνύματα νόμιμων χρηστών.

Water Torture

Πρόκειται για μια βελτιωμένη έκδοση της DNS flood, που σκοπό έχει να εξαντλήσει την επεξεργαστική ισχύ ενός Authoritative Server [22].

Ο επιτιθέμενος προετοιμάζεται για την επίθεση, δημιουργώντας το botnet του και συγκεντρώνει μια λίστα με διαθέσιμους open Resolvers του διαδικτύου. Στη συνέχεια, διατυπώνει έναν πολύ μεγάλο αριθμό από IP spoofed ερωτήματα DNS τυχαίας μορφής προς τους open Resolvers που έχει καταγράψει. Οι open Resolvers, σύμφωνα με τον τρόπο λειτουργίας που περιγράφηκε, προωθούν αυτά τα ερωτήματα στον αντίστοιχο Authoritative Server.

Τα ερωτήματα είναι κατασκευασμένα με τέτοιο τρόπο ώστε να μην μπορούν να βρεθούν στην προσωρινή μνήμη των Resolvers. Με χρήση ενός προγράμματος που παράγει μη επαναλαμβανόμενα τυχαία ονόματα, ο επιτιθέμενος εξασφαλίζει ότι σε κάθε αίτημα το όνομα προς επίλυση θα είναι διαφορετικό από τα προηγούμενα και επομένως ο Resolver θα αναγκαστεί να επικοινωνήσει με τον Authoritative Server.



Σχήμα 2.23 - Επίθεση DNS Water Torture.

Έτσι, ο Authoritative Server σπαταλά επεξεργαστική ισχύ για να βρει το όνομα που του έχει ζητηθεί και καταλήγει να απαντήσει με NXDOMAIN (το όνομα δεν υπάρχει). Ο μεγάλος όγκος των ερωτημάτων αυτών όμως αυξάνει το ποσοστό χρήσης του επεξεργαστή του Server σε βαθμό που να μην μπορεί να εξυπηρετήσει άλλους χρήστες.

Ένα ακόμη αποτέλεσμα αυτής της επίθεσης είναι η μειωμένη απόδοση των open Resolvers που χρησιμοποιούνται για την επίθεση. Εφόσον η προσωρινή τους μνήμη γεμίζει με ερωτήματα που δέχονται από τα bots, κάθε νέο έγκυρο ερώτημα από νόμιμους χρήστες πιθανόν να μην υπάρχει αποθηκευμένο εκεί, και ως εκ τούτου οι Resolvers δεν απαντούν απευθείας αλλά χρειάζεται να ακολουθήσουν ολόκληρη την διαδικασία επίλυσης ονόματος. Οπότε ακόμη και σε ερωτήματα που δέχθηκαν πρόσφατα αργούν να απαντήσουν.

Ως μέθοδος αντιμετώπισης χρησιμοποιείται η απόρριψη μηνυμάτων, όταν αυτά ξεπεράσουν ένα όριο μέσα σε ένα ορισμένο χρονικό διάστημα. Ωστόσο, αυτή η προσέγγιση έχει το μειονέκτημα ότι τα πακέτα που απορρίπτονται μπορεί να προέρχονται από νόμιμους χρήστες.

Reflection-Amplification

Η επίθεση αυτή [23] έχει σκοπό να πλήξει το εύρος ζώνης του δικτύου του θύματος. Αυτό το επιτυγχάνει στέλνοντας ένα πολύ μεγάλο αριθμό μηνυμάτων στο θύμα εκμεταλλεύοντας τους open Resolvers. Στόχος αυτής τη επίθεσης μπορεί να είναι οποιοσδήποτε υπολογιστής και όχι απαραίτητα ένας DNS Server.

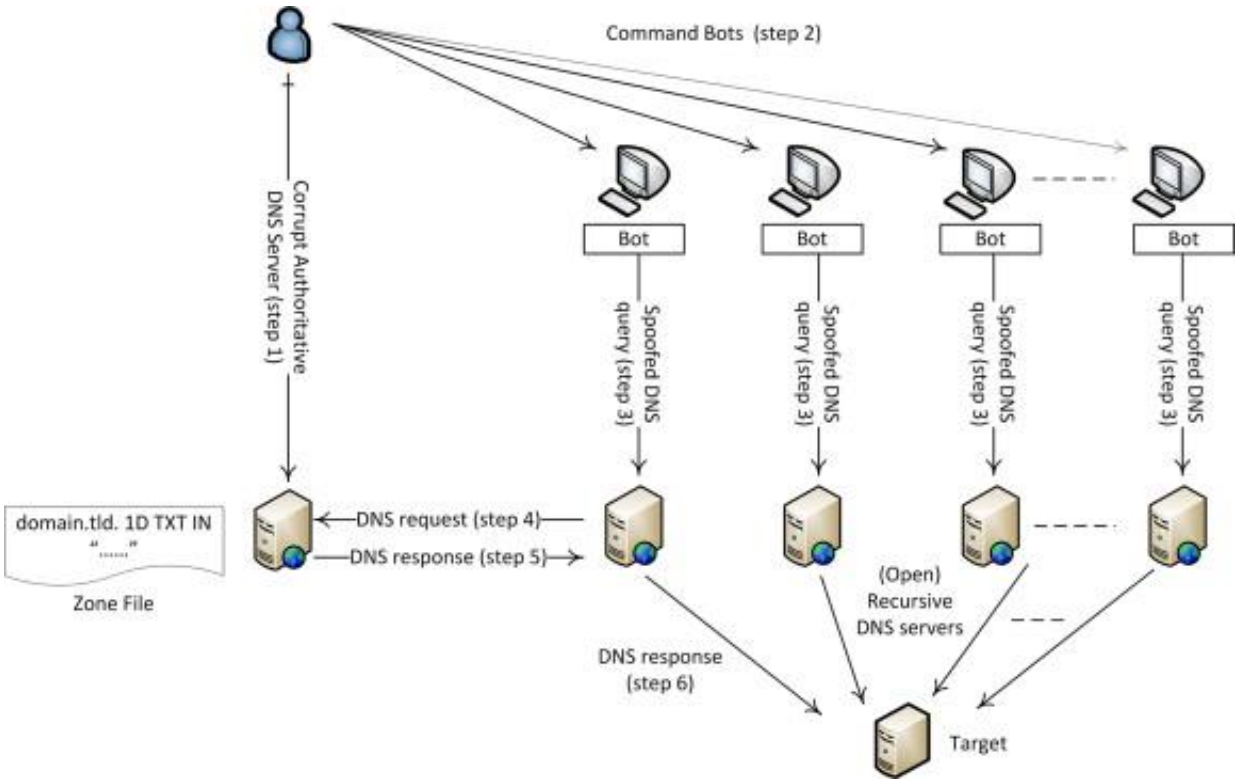
Η συγκεκριμένη επίθεση DNS χρησιμοποιεί τους Resolvers ως ανακλαστήρες (reflectors) για να κατευθύνει την κίνηση δικτύου DNS προς έναν στόχο. Ουσιαστικά, ένας κακόβουλος χρήστης στέλνει spoofed αιτήματα DNS σε αναδρομικούς αναλυτές. Χρησιμοποιώντας IP spoofing, ο επιτιθέμενος θέτει ως IP προέλευσης των πακέτων αυτών τη διεύθυνση IP του θύματος. Επομένως, οι αντίστοιχες απαντήσεις κατευθύνονται στον στόχο αντί για τον αποστολέα του αιτήματος. Η αποτελεσματικότητα αυτής της επίθεσης έγκειται στο γεγονός ότι ένα μικρό αίτημα DNS μπορεί να προκαλέσει πολύ μεγαλύτερη απάντηση. Ο λόγος του μεγέθους της απάντησης προς το μεγέθους του αιτήματος ονομάζεται παράγοντας ενίσχυσης (amplification factor). Αυτός ο παράγοντας είναι μια άμεση ένδειξη του αντίκτυπου της επίθεσης. Όσο μεγαλύτερος είναι ο παράγοντας ενίσχυσης, τόσο πιο γρήγορα κατακλύζεται με κίνηση το εύρος ζώνης του θύματος.

Ο επιτιθέμενος έχει στην διάθεσή του ένα μεγάλο botnet, δηλαδή, πολλούς υπολογιστές που μπορούν να στέλνουν αρκετά spoofed αιτήματα ταυτόχρονα. Επίσης, για να αυξήσει κατά πολύ τον παράγοντα ενίσχυσης, κατά το στάδιο προετοιμασίας της επίθεσης, έχει αναζητήσει αιτήματα DNS, τα οποία για μικρό μέγεθος ερώτησης επιστρέφουν πολύ μεγάλο μέγεθος απάντησης. Τέτοια ερωτήματα είναι, κυρίως, αυτά που χρησιμοποιούν DNSSec ή αιτήματα τύπου ANY. Τα αιτήματα τύπου ANY, επιστρέφουν απαντήσεις με όλους τους τύπους εγγραφών (A, AAAA, MX κλπ.). Ένας άλλος τρόπος είναι ο επιτιθέμενος να τοποθετήσει μια μεγάλη εγγραφή πόρων TXT (μερικά KB) σε έναν DNS Server που ελέγχει. Αυτό σημαίνει ότι είτε ο επιτιθέμενος κατέχει το Domain για το οποίο ο DNS Server είναι Authoritative είτε αποκτά πρόσβαση στο αρχείο ζώνης του και το αλλάζει.

Με τέτοιες τεχνικές ο παράγοντας ενίσχυσης παίρνει μεγάλες τιμές που μπορεί να φτάσει και το 70.

Η τεχνική του reflection στην επίθεση, βοηθάει τον επιτιθέμενο να αποκρύψει την ταυτότητά του ακόμα περισσότερο, αφού δε χρησιμοποιεί μόνο το botnet ως κάλυψη, αλλά και τους open Resolvers. Στην περίπτωση που ο επιτιθέμενος δε χρησιμοποιούσε reflection, το θύμα θα μπορούσε να χρησιμοποιήσει, στα άκρα του δικτύου του, τεχνικές φιλτραρίσματος διευθύνσεων IP. Αυτό σημαίνει ότι θα απέρριπτε πακέτα με κριτήριο τη διεύθυνση IP, καθώς θα γνώριζε ποιες είναι εκείνες που του στέλνουν κακόβουλη κίνηση και θα αντιμετώπιζε την επίθεση αποτελεσματικά. Ωστόσο, όταν χρησιμοποιείται reflection, το θύμα βλέπει ως αποστολείς των μηνυμάτων DNS τους open Resolvers οι οποίοι δεν

χρησιμοποιούνται μόνο από τον επιτιθέμενο, αλλά και από νόμιμους χρήστες. Είναι δύσκολο, λοιπόν, για εκείνο να διαχωρίσει την καλόβουλη και την κακόβουλη κίνηση με κριτήριο τις διευθύνσεις IP προέλευσης.



Σχήμα 2.24 - Επίθεση Reflection-Amplification.

Cache poisoning

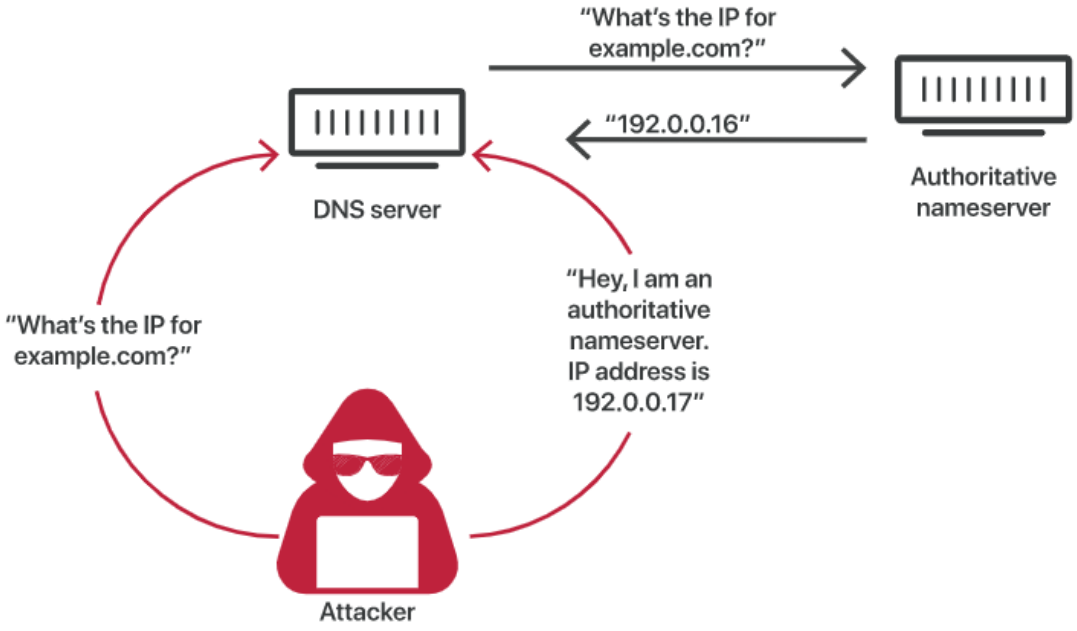
Η επίθεση DNS Cache poisoning [24] λειτουργεί με τρόπο που εκμεταλλεύεται την δομή της επικοινωνίας μεταξύ εξυπηρετητών DNS και Resolver. Όταν ένας DNS Resolver επιχειρεί να πραγματοποιήσει αναζήτηση ενός Domain, λόγω ενός ερωτήματος που του έχει τεθεί, θα ακολουθήσει την γνωστή διαδικασία επίλυσης ονόματος. Δεδομένου ότι δεν υπάρχει η απάντηση στην προσωρινή του μνήμη, δεν γνωρίζει ποιος DNS Server είναι υπεύθυνος για τον τομέα και δεν γνωρίζει την πλήρη διαδρομή προς κάθε Authoritative Server, δέχεται απαντήσεις στα ερωτήματά του από οπουδήποτε, αρκεί η απάντηση να ταιριάζει με το ερώτημα και να έχει μορφοποιηθεί σωστά.

Ένας κακόβουλος χρήστης μπορεί να εκμεταλλευτεί αυτόν τον σχεδιασμό απαντώντας πριν τον πραγματικό Authoritative Server στον Resolver και σαν αποτέλεσμα, ο τελευταίος θα χρησιμοποιήσει αυτή την εγγραφή DNS αντί για την

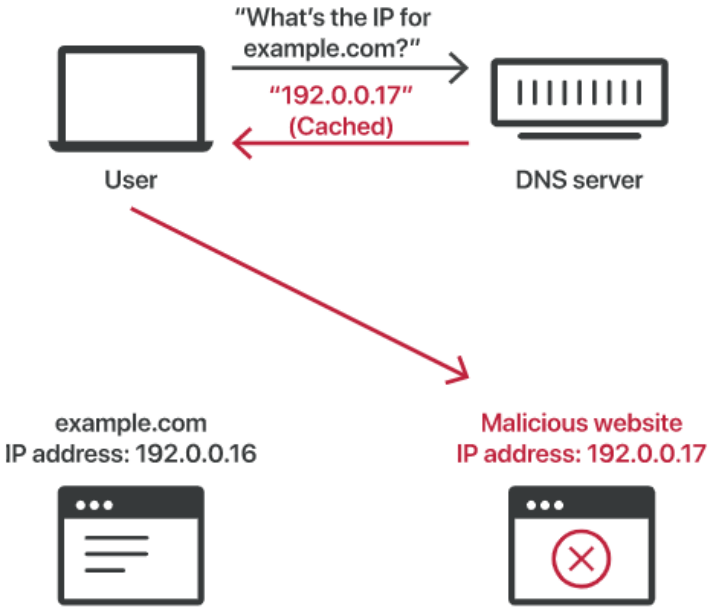
πραγματική απάντηση. Λόγω της φύσης του DNS, ο αναδρομικός αναλυτής δεν έχει τρόπο να προσδιορίσει ποια απάντηση είναι πραγματική και ποια είναι ψεύτικη.

Αυτή η επίθεση επιδεινώνεται λόγω του dns caching των Resolvers. Όπως έχει αναφερθεί οι αναδρομικοί αναλυτές αποθηκεύουν προσωρινά τις αναζητήσεις εσωτερικά, ώστε να μην χρειάζεται να χάνουν χρόνο και να υποβάλλουν ερωτήματα στους DNS Servers κάθε φορά που ζητείται ένα Domain. Αυτό έχει σαν αποτέλεσμα, η ψεύτικη εγγραφή του επιτιθέμενου να αποθηκευτεί στην προσωρινή μνήμη του Resolver κάτι που σημαίνει ότι σε κάθε ερώτημα που του γίνεται θα δοθεί η πλαστή εγγραφή ως απάντηση, η οποία ενδεχομένως να ανακατευθύνει τους χρήστες στον ιστότοπο του επιτιθέμενου.

DNS Cache Poisoning Process:



Poisoned DNS Cache:



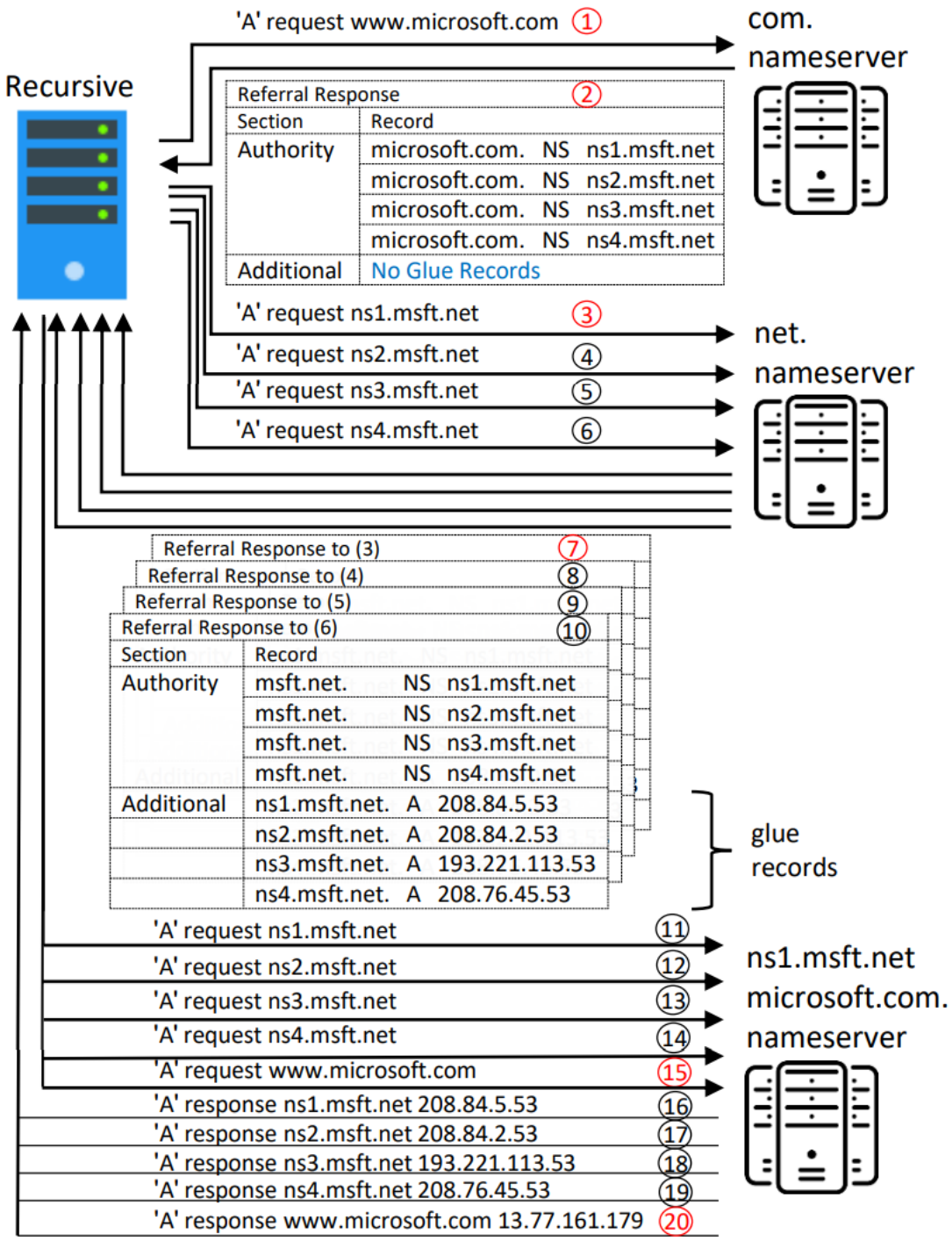
Σχήμα 2.25 - DNS Cache Poisoning.

NXNS Attack Amplification

Η NXNS Attack [25] είναι μια νέα επίθεση που εκμεταλλεύεται τον τρόπο λειτουργίας των αναδρομικών αναλυτών DNS κατά τη λήψη απάντησης παραπομπής (referral response) που περιέχουν name Servers αλλά χωρίς τις αντίστοιχες διευθύνσεις IP τους. Ο αριθμός των μηνυμάτων DNS που ανταλλάσσονται σε μια τυπική διαδικασία ανάλυσης μπορεί να είναι πολύ υψηλότερος στην πράξη από ότι αναμένεται θεωρητικά, κυρίως λόγω μιας προληπτικής ανάλυσης των διευθύνσεων IP των name Servers.

Όταν ένας Resolver εκτελεί επίλυση ενός ονόματος διατρέχει την ιεραρχία DNS, ξεκινώντας από το Root, προκειμένου να φτάσει στον Authoritative Server για το ζητούμενο Domain. Κάθε DNS Server, απαντάει με ένα μήνυμα παραπομπής σε άλλους name Servers, δηλαδή λέει στον Resolver να ζητήσει την πληροφορία που ψάχνει σε κάποιους name Servers.

Προκειμένου να υπάρχει ανοχή σε σφάλματα ενός name Server, παρέχονται παραπάνω από ένας (ns1, ns2 κλπ) και η απάντηση παραπομπής περιέχει και τις IP διευθύνσεις τους (glue records). Αυτές παρέχονται σε τύπου A Resource Record, αλλά το DNS δεν έχει κάποιους σαφείς κανονισμούς σχετικά με το αν αυτές πρέπει ή όχι να υπάρχουν. Εάν λοιπόν, δεν δίνονται οι διευθύνσεις για κάποιους name Servers, τότε ο Resolver πρέπει να κάνει επίλυση των ονομάτων τους. Επομένως, η επίλυση ενός ονόματος στην πράξη, έχει την παρακάτω μορφή [25].



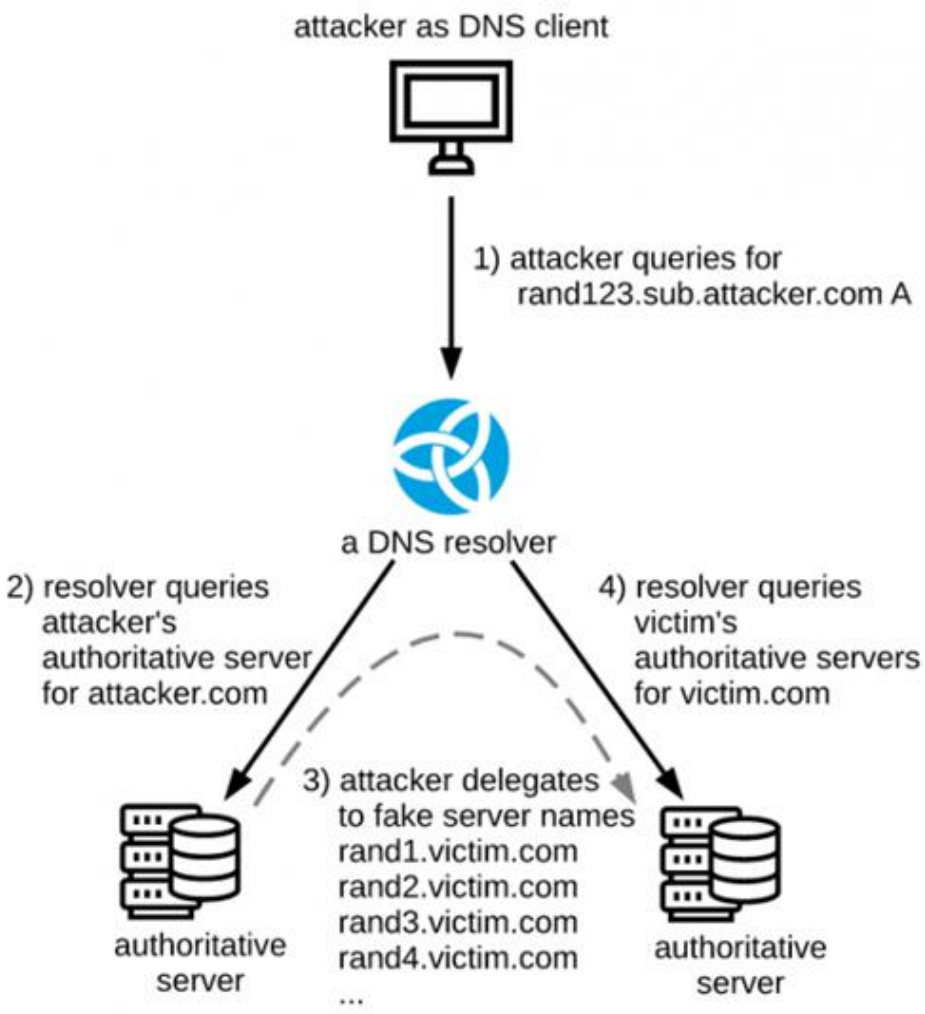
Σχήμα 2.26 - Η διαδικασία επίλυσης ονόματος αναλυτικά.

Η επίθεση το χρησιμοποιεί αυτό προκειμένου να πλημμυρίσει έναν TLD ή και έναν second level Domain Authoritative Server. Αρχικά, ο επιτιθέμενος δημιουργεί ένα δικό του Domain, έστω attacker.com και τοποθετεί σε αυτό κατάλληλα Resource Records.

Στη συνέχεια στέλνει σε έναν Resolver αίτημα για επίλυση ενός ονόματος στο Domain που δημιούργησε. Ο Authoritative Server του επιτιθέμενου απαντάει με τα

Resource Records που τοποθέτησε τα οποία περιέχουν ψεύτικες εγγραφές όπως fake1.victim.com, fake2.victim.com, κλπ. Αυτό οδηγεί σε μια εγγραφή που δεν περιέχει καμία διεύθυνση IP. Επομένως, ο Resolver πρέπει να συνδεθεί στον Authoritative εξυπηρετητή του θύματος (victim.com) για να αναζητήσει τα ψεύτικα ονόματα σε όλα τα Subdomains, καθώς και στο victim.com. Καθώς οι εγγραφές είναι ψεύτικες, ο Server θα απαντήσει με ένα μήνυμα σφάλματος [56].

Το πλήθος των επιτρεπόμενων ψεύτικων εγγραφών, σε συνδυασμό με ένα botnet και πολλούς Resolver να βομβαρδίζουν με ερωτήματα τους DNS Servers, οδηγεί σε παράγοντα ενίσχυσης μεγαλύτερο από 1620. Λόγω της πολύ υψηλής τιμής του παράγονται ενίσχυσης οι συνέπειες είναι αρκετά σοβαρές. Επιβαρύνονται τόσο οι Authoritative Servers που έχουν στοχοποιηθεί όσο και οι Resolvers που χρησιμοποιούνται για την επίθεση. Το αποτέλεσμα είναι κατασπατάληση του εύρους ζώνης και των πόρων τους.



Σχήμα 2.27 - Επίθεση NXNS Amplification.

2.3. eBPF & XDP

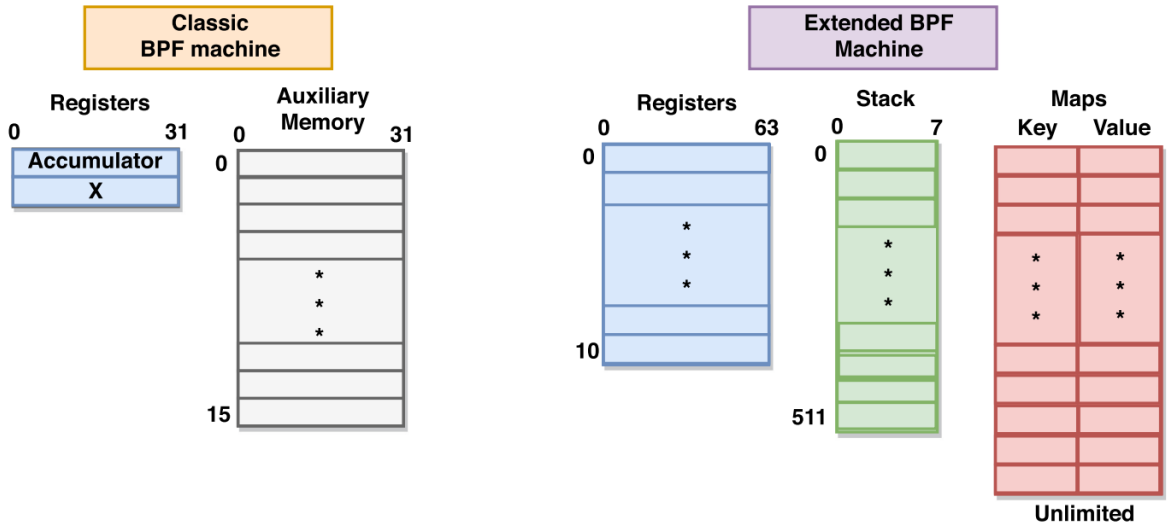
Σε αυτό το κεφάλαιο θα αναλυθούν οι κύριες θεωρητικές και θεμελιώδεις πτυχές του extended Berkeley Packet Filter (eBPF) και του eXpress Data Path (XDP), για να γίνει κατανοητή η γενική λειτουργία και η χρήση των δύο τεχνολογιών.

2.3.1. eBPF

Το eBPF είναι επέκταση του Berkeley Packet Filter (BPF) [59], το οποίο προτάθηκε από τους Steven McCanne και Van Jacobson, το 1992 και είχε ως κύρια εφαρμογή το φιλτράρισμα πακέτων στον πυρήνα σε συστήματα Unix BSD. Το BPF περιείχε ένα σύνολο εντολών και μια εικονική μηχανή (VM) για την εκτέλεση προγραμμάτων, γραμμένων στο σύνολο αυτό [26]. Ένα πολύ γνωστό εργαλείο που χρησιμοποιεί BPF είναι η βιβλιοθήκη libcap, που χρησιμοποιείται από το tcpdump [46].

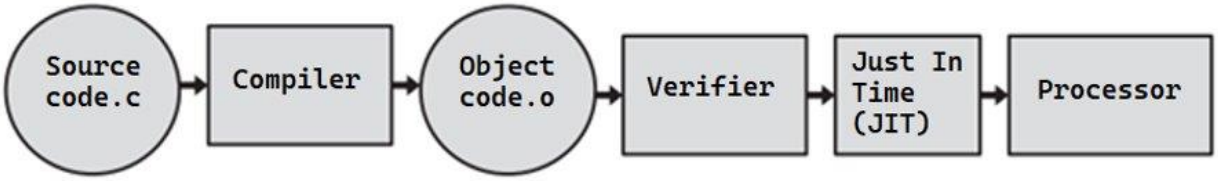
Το ιδιαίτερο χαρακτηριστικό του BPF ήταν η ικανότητά του να εκτελεί προγράμματα χρήστη στον πυρήνα χωρίς να χρειάζεται να τροποποιηθεί ο κώδικάς του. Αυτό οδήγησε στο συμπέρασμα ότι υπάρχουν και άλλες εφαρμογές που θα μπορούσαν να εκμεταλλευτούν αυτή την ικανότητα. Έτσι, βελτιώνοντας την εικονική μηχανή και την συνολική αρχιτεκτονική του BPF, το οποίο στην συνέχεια ονομάστηκε cBPF (classic BPF), προέκυψε το eBPF.

Στο παρακάτω σχήμα φαίνονται οι διαφορές των δύο τεχνολογιών. Αυξήθηκαν οι καταχωρητές και το μέγεθός τους, προστέθηκε στοίβα καθώς και άλλες βοηθητικές δομές δεδομένων.



Σχήμα 2.28 - Χαρακτηριστικά των cBPF και eBPF.

Το σύστημα eBPF αποτελείται από μια σειρά στοιχείων για την μεταγλώττιση , την επαλήθευση και την εκτέλεση του πηγαίου κώδικα των εφαρμογών. Η τυπική ροή του συστήματος eBPF απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 2.29 - Ροή προγράμματος eBPF.

Χρησιμοποιείται ένα υποσύνολο της γλώσσας C για τον κώδικα των προγραμμάτων. Το πρόγραμμα μεταγλωττίζεται με τον clang compiler [69], και προκύπτει έτσι το αρχείο κώδικα αντικειμένου. Πριν αυτό θα φορτωθεί σε έναν επεξεργαστή για να εκτελεστεί περνάει από μια σειρά ελέγχων για να εξασφαλιστεί ότι η εκτέλεσή του δεν θα προκαλέσει προβλήματα στην λειτουργία του πυρήνα. Εάν περάσει αυτούς τους ελέγχους μετατρέπεται σε γλώσσα μηχανής από έναν JIT (Just In Time) compiler και φορτώνεται στον πυρήνα ή σε προγραμματιζόμενες κάρτες δικτύου (SmartNICs). Σε αντίθετη περίπτωση, απορρίπτεται από τον επαληθευτή.

Μεταγλωττιστής και Επαληθευτής

Η τρέχουσα υλοποίηση του eBPF επιβάλει κάποιους κανονισμούς που πρέπει να τηρούνται από τα προγράμματα.

Οι πιο βασικοί είναι οι εξής [26]:

- Το eBPF μπορεί να χρησιμοποιήσει μόνο ένα υποσύνολο βιβλιοθηκών της γλώσσας C. Για παράδειγμα, η συνάρτηση printf δεν είναι διαθέσιμη. Για τύπωση σφαλμάτων και μηνυμάτων, ωστόσο, μπορεί να χρησιμοποιηθεί μια βοηθητική συνάρτηση, που παρέχεται από το ebpf, η bpf_trace_printk() [70]
- Δεν επιτρέπονται μη στατικές καθολικές (global) μεταβλητές.
- Επιτρέπονται μόνο οριοθετημένοι βρόχοι. Ατέρμονοι βρόχοι, ή βρόχοι των οποίων το μέγεθος είναι μεταβλητό και όχι προκαθορισμένο δεν επιτρέπονται. Αυτό γίνεται και για προστασία της λειτουργίας του πυρήνα, καθώς ένας βρόχος που δεν τερματίζει μονοπωλεί τον πυρήνα και δημιουργεί προβλήματα στην σωστή λειτουργία του.

- Η στοίβα, που προστέθηκε κατά την επέκταση του BPF, περιορίζεται σε 512 bytes.
- Το πλήθος των εντολών assembly του προγράμματος έχει όριο τις 4096.

Ο επαληθευτής , με στατική ανάλυση κώδικα, ουσιαστικά ελέγχει αν τηρούνται κάποιοι κανονισμοί, συμπεριλαμβανομένων των παραπάνω. Η υλοποίησή του είναι διαθέσιμη στο αρχείο kernel/bpf/verifier.c στον πηγαίο κώδικα του πυρήνα. Όπως αναφέρθηκε ο verifier καλείται αφού ο κώδικας έχει μεταγλωττιστεί και κατά τη διαδικασία φόρτωσης του προγράμματος στον πυρήνα.

Ένας ακόμη έλεγχος που εκτελείται από τον επαληθευτή, έχει να κάνει με τις προσβάσεις μνήμης του προγράμματος. Για να διασφαλιστεί η ακεραιότητα και η ασφάλεια του πυρήνα δεν επιτρέπει την πρόσβαση σε θέσεις μνήμης πέρα από τις τοπικές μεταβλητές και τα όρια πακέτων. Σε κάθε πρόσβαση σε οποιαδήποτε byte στο πακέτο, είναι πάντα απαραίτητος ο συνοριακός έλεγχος (boundary check), έτσι ώστε να είναι βέβαιο ότι οι θέσεις μνήμης είναι μέσα στα επιτρεπόμενα όρια. Εάν το πρόγραμμα eBPF δεν κάνει αυτόν τον τύπο ελέγχου, τότε ο επαληθευτής το απορρίπτει και έτσι δεν μπορεί να φορτωθεί.

Υπάρχουν επίσης προσπάθειες από έργα ανοιχτού κώδικα για την εφαρμογή μεταγλωττιστή P4 για το eBPF [50, 51]. Ακόμη, το BPF Compiler Collection (BCC) [38] παρέχει επιπλέον λειτουργίες από τον τυπικό κώδικα σε C για τη διευκόλυνση της γραφής και της αλληλεπίδρασης με προγράμματα eBPF, καθώς και η libbpf [60], η κύρια βιβλιοθήκη για αλληλεπίδραση του χώρου χρήστη με το eBPF [26].

Χάρτες eBPF

Οι χάρτες eBPF είναι δομές δεδομένων που παρέχονται από το eBPF, λειτουργούν σαν αποθηκευτικοί χώροι γενικού σκοπού, και χρησιμοποιούν την δομή κλειδιού-τιμής (Key/Value store). Τα κλειδιά και οι τιμές επιτρέπουν την αποθήκευση δομών και τύπων δεδομένων που καθορίζονται από το χρήστη, των οποίων τα μεγέθη πρέπει να ορίζονται κατά την δημιουργία του χάρτη.

Οι χάρτες έχουν διάφορους τύπους, καθένας από τους οποίους προσφέρει και διαφορετική λειτουργικότητα. Για παράδειγμα, ένας χάρτης μπορεί να χρησιμοποιηθεί ως απλός πίνακας (BPF_MAP_TYPE_ARRAY) ή πίνακας κατακερματισμού (BPF_MAP_TYPE_HASH) καθώς και να έχει ένα αντίγραφο για κάθε επεξεργαστικό πυρήνα (BPF_MAP_TYPE_PERCPU_ARRAY) δίνοντας έτσι την δυνατότητα στον κάθε πυρήνα να διαβάζει ή να γράφει σε ξεχωριστό στιγμιότυπο του χάρτη. Μερικοί από αυτούς τους τύπους είναι :

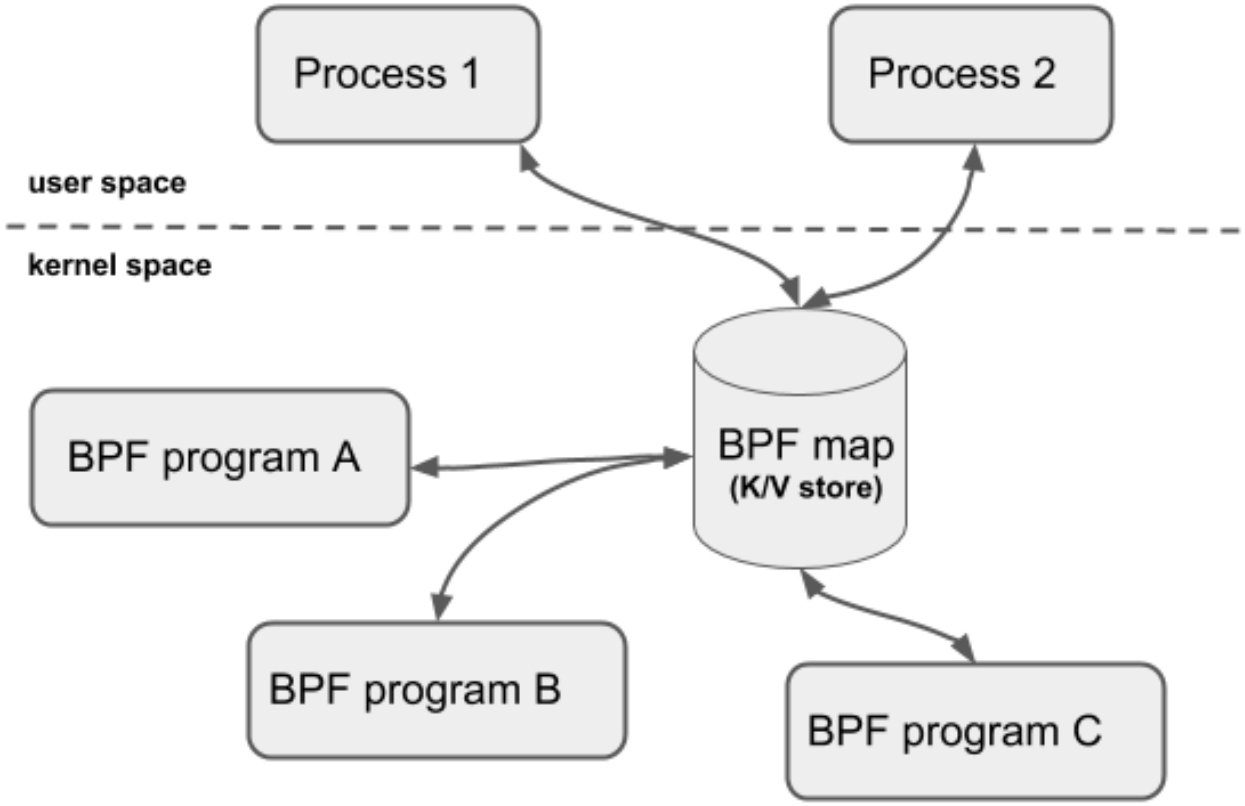
- BPF_MAP_TYPE_UNSPEC

- BPF_MAP_TYPE_HASH
- BPF_MAP_TYPE_ARRAY
- BPF_MAP_TYPE_PROG_ARRAY
- BPF_MAP_TYPE_PERF_EVENT_ARRAY
- BPF_MAP_TYPE_PERCPU_HASH
- BPF_MAP_TYPE_PERCPU_ARRAY
- BPF_MAP_TYPE_STACK_TRACE
- BPF_MAP_TYPE_CGROUP_ARRAY

Μια διεργασία που εκτελείται σε χώρο χρήστη μπορεί να δημιουργήσει πολλαπλούς χάρτες οι οποίοι να είναι προσβάσιμοι και από τις διεργασίες του χώρου χρήστη αλλά και από προγράμματα eBPF που φορτώνονται στον πυρήνα, επιτρέποντας έτσι την ανταλλαγή δεδομένων μεταξύ των δύο περιβαλλόντων.

Στο eBPF, κάθε αντικείμενο (πρόγραμμα, χάρτης, κ.λπ.) έχει έναν μετρητή (Refcnt) που δείχνει πόσες αναφορές έχει και διατηρείται από τον πυρήνα ο οποίος αυξάνει τον μετρητή κάθε φορά που φορτώνεται ένα νέο πρόγραμμα eBPF που χρησιμοποιεί τον χάρτη και τον μειώνει αντίστοιχα κάθε φορά που ένα από αυτά τερματίζει. Όταν ένα refcnt φτάσει στο μηδέν, απελευθερώνεται η αντίστοιχη θέση μνήμης, καταστρέφοντας το αντικείμενο eBPF που σχετίζεται με τον μετρητή [26].

Επιτρέπεται η κοινή χρήση των χαρτών μεταξύ πολλών προγραμμάτων ταυτόχρονα αφού ο χάρτης διατηρείται ανοιχτός όσο υπάρχει η μητρική του διεργασία ή κάποιο πρόγραμμα eBPF που τον χρησιμοποιεί.



Σχήμα 2.30 - Χρήση χάρτη eBPF από πολλαπλά προγράμματα χώρου χρήστη και χώρου πυρήνα.

Εφαρμογές eBPF

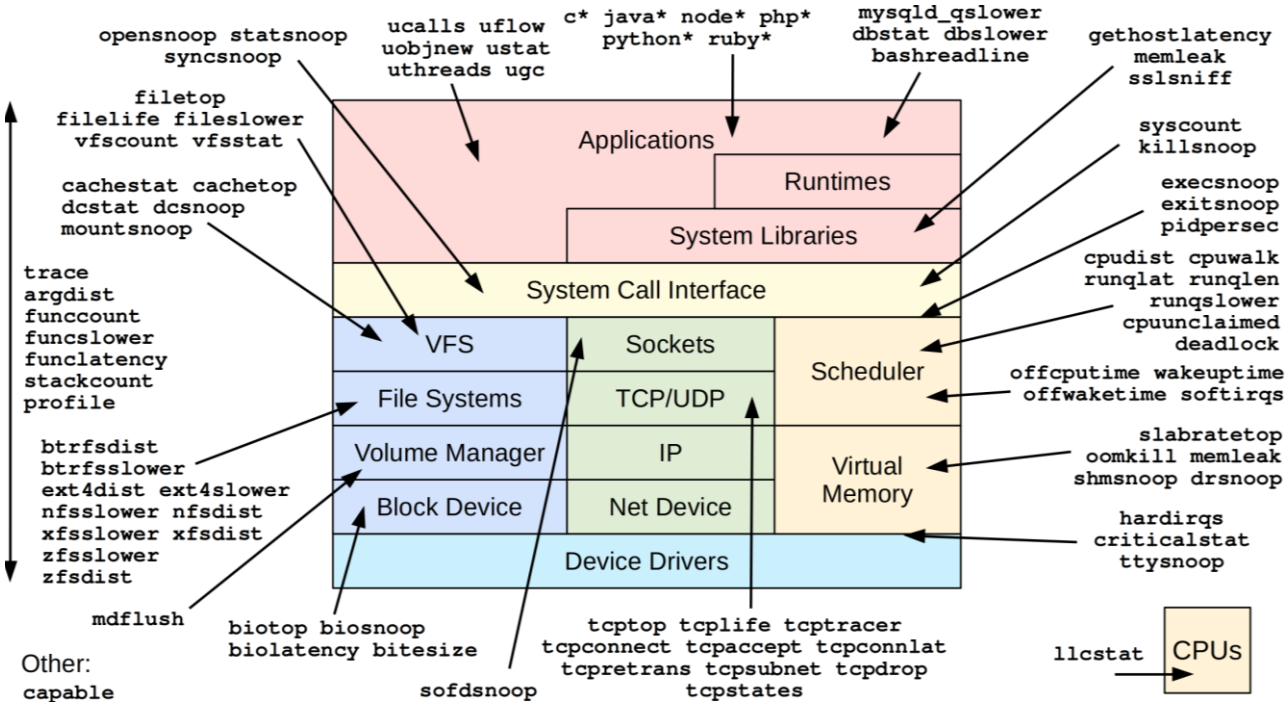
Τα προγράμματα eBPF έχουν διάφορους τύπους. Μερικοί από αυτούς είναι [71]:

- BPF_PROG_TYPE_SOCKET_FILTER
- BPF_PROG_TYPE_KPROBE
- BPF_PROG_TYPE_TRACEPOINT
- BPF_PROG_TYPE_XDP
- BPF_PROG_TYPE_PERF_EVENT
- BPF_PROG_TYPE_CGROUP_SKB
- BPF_PROG_TYPE_CGROUP_SOCK
- BPF_PROG_TYPE_SOCK_OPS
- BPF_PROG_CGROUP_DEVICE

Ο τύπος του προγράμματος σχετίζεται με το σημείο στο οποίο αυτά τα προγράμματα προσαρτώνται. Ανάλογα με το σημείο προσάρτησης, τα προγράμματα eBPF

προσφέρουν διαφορετικά οφέλη και μπορούν να χρησιμοποιηθούν για διαφορετικές εφαρμογές :

- **Ανίχνευση (Tracing) :** Τα προγράμματα μπορούν να τοποθετηθούν σε σημεία ανίχνευσης (trace points) έτσι ώστε να παρέχουν πληροφορίες σχετικά με εφαρμογές του πηρύνα [71, 72]. Αυτό βοηθάει στον καλύτερο έλεγχο εφαρμογών και παρέχει πληροφορίες σχετικά με αυτές.
- **Παρακολούθηση (Monitoring) :** Τα προγράμματα eBPF μπορούν να συλλέγουν στατιστικά σχετικά με την λειτουργία διάφορων εφαρμογών και έτσι να δίνουν μια καλύτερη εικόνα σχετικά με τις επιδόσεις ενός συστήματος [73].
- **Ασφάλεια συστημάτων (System securtiy) :** Λόγω της ικανότητάς του να παρακολουθεί κλήσεις συστήματος (system calls), να φιλτράρει την δικτυακή κίνηση και να παρέχει πληροφορίες για το περιεχόμενο εκτέλεσης κάθε εφαρμογής του πυρήνα το eBPF δίνει την δυνατότητα στους προγραμματιστές να έχουν πολύ καλύτερο έλεγχο των συστημάτων τους και να υλοποιούν μηχανισμούς που ενισχύουν την ασφάλειά τους [74].



Σχήμα 2.31 - Εργαλεία ανίχνευσης (tracing) Linux bcc/BPF.

2.3.2. XDP

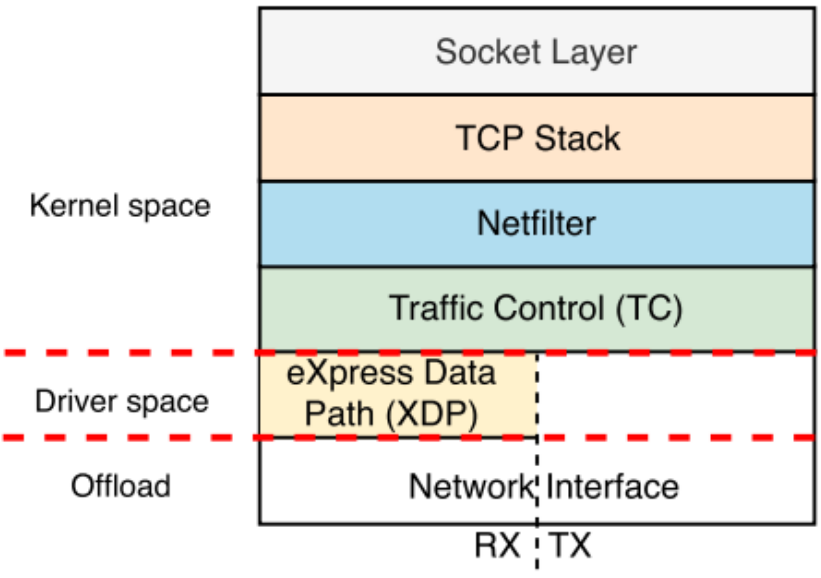
Τα προγράμματα eBPF, όπως αναφέρθηκε, μπορούν να προσαρτηθούν σε διάφορα σημεία ενός συστήματος και να εκτελούνται κάθε φορά που συμβαίνει ένα γεγονός

(π.χ. λήψη πακέτου). Αυτά τα σημεία ονομάζονται hooks. Είναι μηχανισμοί που μπορούν να χρησιμοποιηθούν για την παρακολούθηση πακέτων πριν από την κλήση ή κατά την εκτέλεση του λειτουργικού συστήματος. Ο πυρήνας του Linux διαθέτει πολλά hooks και ένα από αυτά είναι το eXpress Data Path (XDP).

Το eXpress Data Path (XDP) [58, 62] είναι το χαμηλότερο επίπεδο της στοίβας δικτύου του πυρήνα του Linux. Είναι παρόν μόνο στη διαδρομή RX (ληφθέντα πακέτα), υλοποιείται μέσα στο πρόγραμμα οδήγησης (driver) μιας κάρτας δικτύου και επιτρέπει την επεξεργασία πακέτων στο αρχικό σημείο της στοίβας δικτύου. Σε αυτό το σημείο δεν έχουν γίνει ακόμη ενέργειες οι οποίες έχουν κόστος στους πόρους του συστήματος, όπως κατανομή μνήμης από το λειτουργικό σύστημα για να προωθηθεί το πακέτο στα επόμενα στρώματα της στοίβας. Έτσι, τα προγράμματα eBPF που προσαρτώνται στο XDP hook, μπορούν να επεξεργαστούν τα εισερχόμενα πακέτα την στιγμή που γίνονται διαθέσιμα στον επεξεργαστή.

Αυτό καθιστά το XDP ως το καλύτερο hook όσον αφορά την ταχύτητα απόδοσης για εφαρμογές όπως η αντιμετώπιση των επιθέσεων DDoS, αφού ενέργειες όπως η απόρριψη μεγάλου όγκου κακόβουλων πακέτων γίνεται ταχύτατα [63], αποτελεσματικά και χωρίς να επιβαρύνουν το λειτουργικό σύστημα.

Τα πακέτα που εισέρχονται στο λειτουργικό σύστημα υποβάλλονται σε επεξεργασία από διάφορα επίπεδα στον πυρήνα, όπως φαίνεται και στο ακόλουθο σχήμα.



Σχήμα 2.32 - Στοίβα δικτύου του πυρήνα του Linux.

Τα πακέτα που προορίζονται για μια εφαρμογή χώρου χρήστη, περνούν από όλα αυτά τα επίπεδα και μπορούν να τροποποιηθούν κατά τη διάρκεια αυτής της διαδικασίας.

Έτσι, τα προγράμματα eBPF τύπου XDP, αφού λάβουν ένα εισερχόμενο πακέτο, μπορούν να εξάγουν πληροφορίες από αυτό ή και να το τροποποιήσουν.

Το πακέτο εισέρχεται στο XDP με την μορφή μια δομής struct xdp_md του ebpf.

```
struct xdp_md {
        __u32 data;
        __u32 data_end;
        __u32 data_meta;
        /* Below access go through struct xdp_rxq_info */
        __u32 ingress_ifindex; /* rxq->dev->ifindex */
        __u32 rx_queue_index;  /* rxq->queue_index  */
};
```

Σχήμα 2.33 - Η δομή που δέχεται ένα πρόγραμμα eBPF τύπου XDP.

Περιέχει δείκτες στην αρχή και στο τέλος του πακέτου, οι οποίες χρησιμοποιούνται για τους συνοριακούς ελέγχους που απαιτούνται. Επίσης, περιέχει τον δείκτη data_meta, ο οποίος περιέχει τη διεύθυνση μιας περιοχής μνήμης, ελεύθερη για χρήση από προγράμματα XDP για ανταλλαγή μεταδεδομένων των πακέτων με άλλα επίπεδα.

Μετά την επεξεργασία ενός πακέτου, ένα πρόγραμμα XDP επιστρέφει μια ενέργεια, η οποία αντιπροσωπεύει την τελική ετυμηγορία σχετικά με το τι πρέπει να γίνει με το πακέτο μετά την έξοδο του προγράμματος.

Ο παρακάτω πίνακας παραθέτει όλες τις πιθανές ενέργειες XDP, τις τιμές τους και την περιγραφή τους. Η ενέργεια καθορίζεται ως κωδικός επιστροφής του προγράμματος.

Τιμή	Ενέργεια	Περιγραφή
0	XDP_ABORTED	Σφάλμα. Απόρριψη πακέτου.
1	XDP_DROP	Απόρριψη πακέτου.
2	XDP_PASS	Αποδοχή πακέτου. Συνέχεια στη στοίβα του πυρήνα (kernel stack)
3	XDP_TX	Ανακατεύθυνση και αποστολή από την διεπαφή που ήρθε.

4	XDP_REDIRECT	Ανακατεύθυνση του πακέτου σε άλλη διεπαφή.

Πίνακας 2.3 - Κωδικοί επιστροφής XDP και η ερμηνεία τους.

Το XDP έχεις τρεις τρόπους λειτουργίας [62]:

1) **Native XDP**. Είναι ο προκαθορισμένος τρόπος λειτουργίας όπου οι drivers της κάρτας δικτύου επεξεργάζονται τα πακέτα στο κατώτατο σημείο της στοίβας δικτύου.
2) **Generic XDP**. Όταν οι drivers της κάρτας δικτύου δεν υποστηρίζουν το XDP, ο πυρήνας παρέχει αυτόν τον τρόπο λειτουργίας για να προσομοιώσει την λειτουργικότητα του XDP. Αυτό σημαίνει ότι το πακέτο επεξεργάζεται σε ένα σημείο της στοίβας δικτύου πολύ αργότερα από ότι στην περίπτωση του Native mode. Η απόδοση σε αυτήν την περίπτωση είναι αρκετά χαμηλότερη αφού χρειάζεται να γίνουν κάποιες κατανομές μνήμης από τον πυρήνα και γι' αυτό δεν προτείνεται για χρήση σε πραγματικά συστήματα. Αυτή η λειτουργία μπορεί να χρησιμοποιηθεί κατά την ανάπτυξη προγραμμάτων σε testing περιβάλλοντα.
3) **Offloaded XDP**. Σε αυτή την περίπτωση το πρόγραμμα εκφορτώνεται στην κάρτα δικτύου (σε όσες το υποστηρίζουν, π.χ. SmartNICs) και δεν εκτελείται από τον επεξεργαστή του συστήματος. Έτσι, το ήδη χαμηλό επεξεργαστικό κόστος του Native XDP, μεταφέρεται στην κάρτα δικτύου, κάνοντας την απόδοση του XDP ακόμα καλύτερη.

Το XDP μπορεί να χρησιμοποιηθεί σε αρκετές εφαρμογές, όπως [62] :

- **Αντιμετώπιση επιθέσεων DDoS**. Όπως αναφέρθηκε και προηγουμένως, οι ενέργειες που χρειάζεται να πραγματοποιηθούν κατά τη διάρκεια μια κατανεμημένης επίθεσης άρνησης παροχής υπηρεσιών πρέπει να εκτελούνται με πολύ μεγάλη ταχύτητα και αποτελεσματικότητα και αυτό καθιστά το XDP μια πολύ καλή επιλογή σε τέτοιου είδους προβλήματα [75].
- **Προώθηση πακέτων και εξισορρόπηση δικτυακής κίνησης**. Το XDP μπορεί να ανακατευθύνει ή να στείλει πίσω πακέτα πολύ γρήγορα και ταυτόχρονα να τα επεξεργάζεται. Αυτό μπορεί να βοηθήσει στην εξισορρόπηση της κίνησης που δέχεται ένα κέντρο δικτύου, αφού με την χρήση XDP μπορεί να κατανεμηθεί καλύτερα μεταξύ πολλών και διαφορετικών υπολογιστών, επιτυγχάνοντας έτσι καλύτερη χρήση των διαθέσιμων πόρων [76].
- **Δειγματοληψία και παρακολούθηση κίνησης**. Εφόσον το XDP μπορεί να επεξεργαστεί και να αναλύσει τα εισερχόμενα πακέτα, μπορεί επίσης να χρησιμοποιήσει αυτές τις πληροφορίες που δέχεται για την εξαγωγή

στατιστικών στοιχείων (π.χ. πόσα πακέτα δέχεται ανά IP, που κατευθύνονται αυτά τα πακέτα κ.λπ.)

2.4. Μηχανική Μάθηση

Η μηχανική μάθηση (Machine Learning ή ML) [27] είναι η μελέτη αλγορίθμων υπολογιστών που μαθαίνουν αυτόματα μέσω της εμπειρίας και της χρήσης δεδομένων. Θεωρείται μέρος της τεχνητής νοημοσύνης. Οι αλγόριθμοι μηχανικής μάθησης δημιουργούν ένα μοντέλο βασισμένο σε δείγματα δεδομένων, γνωστά ως "δεδομένα εκπαίδευσης", προκειμένου να κάνουν προβλέψεις ή να παίρνουν αποφάσεις χωρίς να έχουν προγραμματιστεί ρητά να το κάνουν. Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται σε μια ευρεία ποικιλία εφαρμογών, όπως στην ιατρική, το φιλτράρισμα email και την όραση υπολογιστών, όπου είναι δύσκολο ή ανέφικτο να αναπτυχθούν συμβατικοί αλγόριθμοι για την εκτέλεση των απαιτούμενων εργασιών.

Η μηχανική μάθηση περιλαμβάνει υπολογιστές που ανακαλύπτουν πώς μπορούν να εκτελούν εργασίες χωρίς να έχουν προγραμματιστεί ρητά να το κάνουν. Περιλαμβάνει υπολογιστές που μαθαίνουν από δεδομένα που παρέχονται έτσι ώστε να εκτελούν συγκεκριμένες εργασίες. Για απλές εργασίες που έχουν ανατεθεί σε υπολογιστές, είναι δυνατόν να προγραμματιστούν αλγόριθμοι που να λένε στον υπολογιστή πώς να εκτελέσει όλα τα βήματα που απαιτούνται για την επίλυση του προβλήματος που αντιμετωπίζει. Από την πλευρά του υπολογιστή, δεν απαιτείται μάθηση. Για πιο προηγμένες εργασίες, μπορεί να είναι δύσκολο για έναν άνθρωπο να δημιουργήσει χειροκίνητα τους απαραίτητους αλγόριθμους. Στην πράξη, μπορεί να αποδειχθεί πιο αποτελεσματικό για τον υπολογιστή να αναπτύξει τον δικό του αλγόριθμο, παρά να ζητήσει από τους προγραμματιστές να καθορίσουν κάθε απαραίτητο βήμα.

Η πειθαρχία της μηχανικής μάθησης χρησιμοποιεί διάφορες προσεγγίσεις για να διδάξει στους υπολογιστές πως να εκτελούν εργασίες όπου δεν υπάρχει πλήρως ικανοποιητικός αλγόριθμος. Σε περιπτώσεις όπου υπάρχει τεράστιος αριθμός πιθανών απαντήσεων, μια προσέγγιση είναι να χαρακτηριστούν ορισμένες από τις σωστές απαντήσεις ως έγκυρες. Αυτό μπορεί στη συνέχεια να χρησιμοποιηθεί ως δεδομένο εκπαίδευσης για τη βελτίωση του αλγορίθμου, που χρησιμοποιεί ο υπολογιστής για τον προσδιορισμό των σωστών απαντήσεων. Για παράδειγμα, για την εκπαίδευση ενός συστήματος για την αναγνώριση ενός ψηφιακού χαρακτήρα, έχει χρησιμοποιηθεί συχνά το σύνολο δεδομένων MNIST με χειρόγραφα ψηφία.

Οι προσεγγίσεις μηχανικής μάθησης χωρίζονται σε τρεις ευρείες κατηγορίες, ανάλογα με τη φύση του "σήματος" ή της "ανατροφοδότησης" που είναι διαθέσιμα στο σύστημα μάθησης:

- Εποπτευόμενη εκμάθηση (supervised learning) : Το υπολογιστικό πρόγραμμα δέχεται τις παραδειγματικές εισόδους καθώς και τα επιθυμητά αποτελέσματα και ο στόχος είναι να μάθει έναν γενικό κανόνα προκειμένου να αντιστοιχίσει τις εισόδους με τα αποτελέσματα.
- Μη εποπτευόμενη εκμάθηση (unsupervised learning) : Χωρίς να παρέχονται ζεύγη εισόδων-εξόδων στον αλγόριθμο μάθησης, πρέπει να βρει την δομή των δεδομένων εισόδου. Η μη επιτηρούμενη μάθηση μπορεί να είναι αυτοσκοπός (ανακαλύπτοντας κρυμμένα μοτίβα σε δεδομένα) ή μέσο για ένα τέλος.
- Ενισχυτική Μάθηση (reinforcement learning) : Ένα πρόγραμμα υπολογιστή αλληλοεπιδρά με ένα δυναμικό περιβάλλον στο οποίο πρέπει να επιτευχθεί ένας συγκεκριμένος στόχος (όπως η οδήγηση ενός οχήματος). Χρησιμοποιείται ένα σύστημα επιβράβευσης και τιμωρίας από έναν agent, ανάλογα με τις επιλογές του προγράμματος σε αυτό το δυναμικό περιβάλλον. Ένα άλλο παράδειγμα είναι να μάθει να παίζει ένα παιχνίδι εναντίον κάποιου αντιπάλου (σκάκι).

Η κατηγοριοποίηση των προβλημάτων μηχανικής μάθησης προκύπτει όταν κάποιος ορίσει το επιθυμητό αποτέλεσμα του συστήματος μηχανικής μάθησης :

- Στην ταξινόμηση, τα δεδομένα εισόδου χωρίζονται σε δύο ή περισσότερες κλάσεις, και η μηχανή πρέπει να κατασκευάσει ένα μοντέλο, το οποίο θα αντιστοιχίζει τα δεδομένα σε μία ή περισσότερες (multi-label ταξινόμηση) κλάσεις. Αυτό συνήθως εμπίπτει στην επιτηρούμενη μάθηση. Τα φίλτρα Spam είναι ένα παράδειγμα ταξινόμησης, όπου οι είσοδοι είναι τα emails ή άλλα μηνύματα και οι κλάσεις είναι "spam" και "όχι spam".
- Στην παλινδρόμηση, επίσης πρόβλημα επιτηρούμενης μάθησης, τα αποτελέσματα είναι συνεχή και όχι διακριτά.
- Στην συσταδοποίηση (Clustering), ένα σύνολο εισόδων πρόκειται να χωριστεί σε ομάδες. Σε αντίθεση με την ταξινόμηση, οι ομάδες δεν είναι γνωστές εκ των προτέρων, καθιστώντας αυτόν τον διαχωρισμό τυπική εργασία μη επιτηρούμενης μάθησης.
- Σε προβλήματα μείωσης διαστασιμότητας (dimensionality reduction), τα δεδομένα απλοποιούνται και αντιστοιχίζονται σε ένα χώρο λιγότερων διαστάσεων.

Εποπτευόμενη Εκμάθηση

Η εποπτευόμενη εκμάθηση, λοιπόν, είναι μία κατηγορία μηχανικής μάθησης, στόχος της οποίας είναι ο χαρακτηρισμός δεδομένων με βάση κάποια δεδομένα εκπαίδευσης. Το σύνολο δεδομένων εκπαίδευσης (Training dataset) αποτελείται από παραδείγματα τα οποία χρησιμοποιούνται για εκπαίδευση μοντέλων. Στην επιβλεπόμενη μάθηση, κάθε παράδειγμα αποτελείται από ένα σύνολο εισόδου (συνήθως ένα διάνυσμα από χαρακτηριστικά) και μια επιθυμητή τιμή εξόδου. Οι αλγόριθμοι επιβλεπόμενης μάθησης αναλύουν τα δεδομένα εκπαίδευσης και παράγουν ένα μοντέλο το οποίο μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει νέα παραδείγματα. Χρησιμοποιώντας σύνολα δεδομένων εξέτασης (Testing datasets), των οποίων είναι γνωστές οι επιθυμητές έξοδοι, ελέγχεται η απόδοση του μοντέλου. Το βέλτιστο σενάριο επιτρέπει στον αλγόριθμο να καθορίσει σωστά την ετικέτα της κατηγορίας για άγνωστα μέχρι τώρα παραδείγματα. Για να επιτευχθεί αυτό, απαιτείται ο αλγόριθμος μάθησης να γενικεύει από τα δεδομένα εκπαίδευσης σε αθέατες καταστάσεις με ένα "λογικό" τρόπο.

Ταξινομητής Naive Bayes

Ένα από τα προβλήματα που λύνεται με μεθόδους επιβλεπόμενης μάθησης, όπως αναφέρθηκε, είναι η ταξινόμηση. Ο ταξινομητής Naive Bayes [29] μπορεί να χρησιμοποιηθεί σε τέτοια προβλήματα.

Η βασική ιδέα λειτουργίας του ταξινομητή είναι ο γνωστός νόμος του Bayes,

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

και η (naive) υπόθεση ότι τα χαρακτηριστικά είναι όλα υπό συνθήκη ανεξάρτητα μεταξύ τους. Με δεδομένα μια μεταβλητή κατηγορίας y και ένα εξαρτώμενο διάνυσμα χαρακτηριστικών x_1 μέχρι x_n , σύμφωνα με το θεώρημα του Bayes θα ισχύει :

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Ισχύει ότι :

$$P(x_1, \ldots, x_i, \ldots, x_n \mid y) = \prod_{i=1}^{n} P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

και γίνεται η αφελής υπόθεσης ότι το χαρακτηριστικό x_i για κάθε i εξαρτάται μόνο από την κατηγορία y και όχι από οποιοδήποτε άλλο χαρακτηριστικό :

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y)$$

Επομένως, προκύπτει η παρακάτω σχέση :

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

Με δεδομένη είσοδο, ο παρονομαστής του δεύτερου μέλους είναι σταθερός. Οπότε προκύπτει ο ακόλουθος κανόνας ταξινόμησης :

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

Το P(y) είναι η υπόθεση και ισούται με τη σχετική συχνότητα της κατηγορίας y στο σύνολο δεδομένων εκπαίδευσης. Το P(x|y) είναι η πιθανοφάνεια δηλαδή η πιθανότητα του δείγματος με δεδομένη την υπόθεση που έγινε (y) και μπορεί επίσης να υπολογιστεί απλά από τα δεδομένα εκπαίδευσης.

3. Περιγραφή μηχανισμού

Η μεγάλη σημασία που κατέχει το DNS για την εύρυθμη λειτουργία του σύγχρονου διαδικτύου καθιστά αναγκαία τη λήψη μέτρων για την προστασία του. Η υποδομή του DNS αποτελεί συχνό στόχο των επιτιθέμενων, καθώς τα τρωτά σημεία του πρωτοκόλλου παρέχουν τη δυνατότητα εκτέλεσης πολύ αποτελεσματικών επιθέσεων.

Σε αυτό το κεφάλαιο θα αναλυθεί η δομή και ο τρόπος λειτουργίας του μηχανισμού που υλοποιήθηκε στα πλαίσια της παρούσης διπλωματικής εργασίας. Ο μηχανισμός αυτός, έχει ως στόχο την αντιμετώπιση της επίθεσης DNS Water Torture και χρησιμοποιείται από τους αναδρομικούς εξυπηρετητές. Με την χρήση επιβλεπόμενης μάθησης (ταξινομητής Naive Bayes), υλοποιημένη σε XDP, ένας αναδρομικός εξυπηρετητής ταξινομεί τα ερωτήματα που δέχεται σε έγκυρα και άκυρα και προωθεί μόνο τα έγκυρα. Έτσι, ο Authoritative Server, ο οποίος είναι το θύμα αυτής της επίθεσης, προστατεύεται από τον μεγάλο όγκο κακόβουλης κίνησης και μπορεί να εξυπηρετήσει καλόβουλους χρήστες.

Θα μελετηθούν τρεις καταστάσεις λειτουργίας του αναδρομικού εξυπηρετητή και θα συγκριθούν τα αποτελέσματα για κάθε μία από αυτές:

1) No mitigation :
   Σε αυτόν τον τρόπο λειτουργίας ο αναδρομικός εξυπηρετητής χρησιμοποιεί το λογισμικό BIND [34] και δεν διαθέτει κάποιον αμυντικό μηχανισμό.
2) Mitigation with Naive Bayes – XDP (με χρήση μέγιστου αριθμού αναζητήσεων σε eBPF maps) :
   Σε αυτή την περίπτωση, ο Resolver χρησιμοποιεί το BIND για την εξυπηρέτηση των ερωτημάτων που δέχεται αλλά έχει φορτωθεί στο XDP hook και ο μηχανισμός άμυνας που χρησιμοποιεί τον Naive Bayes για ταξινόμηση των ερωτημάτων. Ακόμη, πραγματοποιείται ο μέγιστος αριθμός αναζητήσεων σε χάρτες eBPF.
3) Mitigation with Naive Bayes – XDP (με χρήση ελάχιστου αριθμού αναζητήσεων σε eBPF maps) :
   Παρόμοια, με τον προηγούμενο τρόπο λειτουργίας, ο αναδρομικός εξυπηρετητής χρησιμοποιεί το BIND και παράλληλα διαθέτει τον αμυντικό μηχανισμό που αναπτύχθηκε. Ωστόσο, η διαφορά είναι πως σε αυτόν τον τρόπο λειτουργίας πραγματοποιείται μικρότερος αριθμός αναζητήσεων σε eBPF maps. Αυτό γίνεται για να εξετασθεί αν οι πολλές αναζητήσεις στους χάρτες έχουν αρνητικό αντίκτυπο στην απόδοση του μηχανισμού. Δεδομένου του τρόπου υλοποίησης του αμυντικού μηχανισμού αυτός ο αριθμός αναζητήσεων είναι και ο ελάχιστος, καθώς αν προσπαθήσουμε να ελαττώσουμε περισσότερο τις αναζητήσεις οδηγούμαστε σε υπέρβαση του ορίου εντολών που επιτρέπει το eBPF. Το πρόβλημα αυτό αναλύεται και στο επόμενο κεφάλαιο που αφορά τους περιορισμούς της υλοποίησης.

Στόχος της παρούσας εργασίας, όπως αναφέρθηκε και στο πρώτο κεφάλαιο, είναι η ανάλυση της απόδοσης του μηχανισμού και όχι η ακρίβεια του Naive Bayes στις ταξινομήσεις καθώς η τελευταία, έχει ήδη μελετηθεί [35] και τα αποτελέσματα δείχνουν υψηλά ποσοστά ακρίβειας στις προβλέψεις.

3.1. Επίπεδο Ελέγχου

Στο επίπεδο ελέγχου (control plane) γίνεται η εκπαίδευση του ταξινομητή, η μεταφορά των απαραίτητων τιμών σε χάρτες eBPF καθώς και η φόρτωση του προγράμματος στο XDP hook. Τα παραπάνω υλοποιούνται στο αρχείο bpf_nb_loader.py

Εκπαίδευση Ταξινομητή Naive Bayes

Η εκπαίδευση του ταξινομητή γίνεται χρησιμοποιώντας ως σύνολα δεδομένων εκπαίδευσης τα εξής :

1) 700.000 έγκυρα ονόματα [36], τα οποία ανήκουν στην λίστα με τις κορυφαίες 1.000.000 ιστοσελίδες σύμφωνα με το Alexa.com [64]
2) 700.000 άκυρα ονόματα, τα οποία παράχθηκαν με τυχαίο τρόπο [37, 65]

Στην συνέχεια εξάγονται επτά χαρακτηριστικά (features) από κάθε όνομα. Τα χαρακτηριστικά που χρησιμοποιήθηκαν είναι :

1) Μήκος ονόματος
2) Πλήθος αριθμών
3) Πλήθος συμφώνων
4) Πλήθος φωνηέντων
5) Μήκος μέγιστης ακολουθίας αριθμών
6) Μήκος μέγιστης ακολουθίας συμφώνων
7) Μήκος μέγιστης ακολουθίας φωνηέντων

Τέλος, υπολογίζεται το πλήθος εμφάνισης των διαφορετικών τιμών κάθε χαρακτηριστικού και διαιρείται με το πλήθος των δεδομένων εκπαίδευσης. Έτσι, προκύπτει η υπό συνθήκη πιθανότητα κάθε χαρακτηριστικού να πάρει κάποια τιμή δεδομένου ότι είναι valid ή invalid. Αυτές οι πιθανότητες χρησιμοποιούνται, όπως αναφέρθηκε, από τον Naive Bayes για να υπολογιστεί η τελική πιθανότητα ενός ονόματος να είναι έγκυρο ή άκυρο.

Επειδή, αυτές οι τιμές θα χρησιμοποιούνται από το επίπεδο δεδομένων (XDP), αποθηκεύονται σε χάρτες eBPF. Δημιουργούνται συνολικά 14 χάρτες για την αποθήκευση αυτών των τιμών, επτά για κάθε χαρακτηριστικό έγκυρων ονομάτων και επτά για κάθε ένα χαρακτηριστικό άκυρων ονομάτων.

Φόρτωση προγράμματος eBPF στο XDP

Για την φόρτωση του προγράμματος eBPF στο XDP hook, χρησιμοποιούνται οι συναρτήσεις που παρέχονται από το BPF Compiler Collection (BCC) [38]. Προσδιορίζονται από τον χρήστη, το τμήμα του προγράμματος eBPF που θα φορτωθεί και η κάρτα δικτύου (network interface) στην οποία θα φορτωθεί.

3.2. Επίπεδο Δεδομένων

Στο επίπεδο δεδομένων (data plane) πραγματοποιείται η επεξεργασία των εισερχόμενων πακέτων, η εξαγωγή των χαρακτηριστικών που αναφέρθηκαν πιο πάνω και η λήψη της απόφασης σχετικά με το εάν το πακέτο θα απορριφθεί ή θα περάσει.

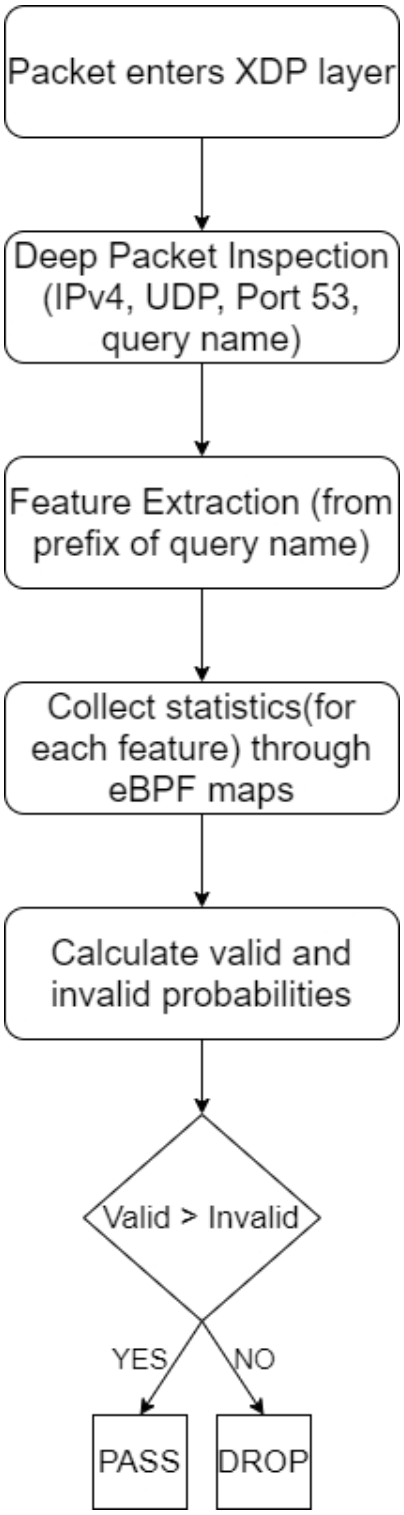
Παρακάτω φαίνεται ένα διάγραμμα ροής, το οποίο απεικονίζει τον τρόπο λειτουργίας του προγράμματος eBPF που φορτώνεται στο XDP hook.

Όπως φαίνεται, αρχικά το πακέτο εισέρχεται στο επίπεδο του XDP. Σε αυτό το σημείο γίνονται οι έλεγχοι σχετικά με το αν το πακέτο χρησιμοποιεί τα πρωτόκολλα Ethernet, IPv4 , αν έχει ως πρωτόκολλο μεταφοράς το UDP, εάν έχει ως θύρα προορισμού την θύρα 53 εάν πρόκειται για πακέτο DNS και τέλος, εάν είναι ερώτημα DNS. Αυτά τα χαρακτηριστικά επιβεβαιώνουν πως το πακέτο έχει την δομή ενός DNS ερωτήματος και θα το επεξεργαστεί το BIND. Επομένως, για αυτά τα πακέτα χρειάζεται να ελεγχθεί το όνομα του ερωτήματος και να ταξινομηθεί ως άκυρο ή έγκυρο. Σε περίπτωση που πρόκειται για διαφορετικού τύπου πακέτο, συνεχίζει κανονικά την πορεία του.

Στη συνέχεια, γίνεται ανάγνωση του ονόματος του DNS ερωτήματος (ως όνομα θεωρείται το prefix του FQDN του ερωτήματος, δηλαδή το πρώτο label από αριστερά), από το πακέτο. Για να τηρούνται οι κανόνες του BPF και να περάσει το πρόγραμμα τον έλεγχο του επαληθευτή για την ανάγνωση του ονόματος, επειδή δεν γνωρίζουμε το μήκος του εκ των προτέρων, ελέγχουμε για κάθε byte αν είναι εντός του ορίου που επιτρέπεται το πρόγραμμα να διαβάσει (boundary check).

Αφού βρεθεί το όνομα του ερωτήματος, γίνεται η εξαγωγή των επτά χαρακτηριστικών που αναφέρθηκαν πιο πάνω. Στην περίπτωση του max_ebpf_maps_searches για να ελεγχθεί εάν ένας χαρακτήρας είναι ψηφίο, φωνήεν ή σύμφωνο γίνεται αναζήτηση του χαρακτήρα προς διερεύνηση σε χάρτες ebpf που περιέχουν ψηφία, φωνήεντα ή σύμφωνα. Αυτοί οι χάρτες, παίρνουν τις τιμές τους από το control plane κατά την έναρξη του προγράμματος. Αυτό, επιφέρει ένα κόστος στην απόδοση του προγράμματος, όμως το σύνολο εντολών assembly είναι μικρότερο από την περίπτωση του min_ebpf_maps_searches, όπου για να γίνει ο παραπάνω έλεγχος συγκρίνουμε τον χαρακτήρα προς διερεύνηση με κάθε ψηφίο,

φωνήεν ή σύμφωνο. Όπως είχε αναφερθεί στο προηγούμενο κεφάλαιο υπάρχει ένα όριο στον αριθμό εντολών assembly ενός bpf προγράμματος και στην συγκεκριμένη περίπτωση φαίνεται πώς αυτό μπορεί να επηρεάσει τελικά την απόδοση ενός προγράμματος.



Σχήμα 3.1 - Διάγραμμα ροής του προγράμματος eBPF του αμυντικού μηχανισμού που αναπτύχθηκε.

Μετά την εξαγωγή χαρακτηριστικών, το πρόγραμμα μέσω αναζητήσεων στους χάρτες χαρακτηριστικών, τους οποίους γεμίζει με τιμές, από το training, το επίπεδο ελέγχου, υπολογίζει με έναν πολλαπλασιασμό (σύμφωνα με τον naibe bayes) την πιθανότητα το όνομα να είναι έγκυρο και άκυρο. Συγκρίνει τις δύο αυτές τιμές και λαμβάνει την ανάλογη απόφαση.

Αν μία τιμή ενός χαρακτηριστικού που εξάχθηκε από το όνομα δεν υπάρχει στους πίνακες, σημαίνει ότι στην εκπαίδευση που έγινε δεν βρέθηκε κανένα όνομα με ίδια τιμή για αυτό το χαρακτηριστικό. Σε αυτή την περίπτωση, επιστρέφεται ένας NULL (κενός) δείκτης από την συνάρτηση αναζήτησης σε ebpf maps. Εάν αυτό συμβεί κατά τον υπολογισμό και της έγκυρης και της άκυρης πιθανότητας τότε το πακέτο απορρίπτεται καθώς θεωρούμε ότι είναι πιο πιθανό ένα άκυρο όνομα να περιέχει τιμή σε ένα από τα χαρακτηριστικά που να μην βρέθηκε σε κανένα όνομα κατά την εκπαίδευση. Εάν επιστραφεί NULL δείκτης μόνο σε μία από τις δύο πιθανότητες τότε αυτή με τον κενό δείκτη τίθεται ίση με 0 και η άλλη πιθανότητα προφανώς υπερισχύει.

Η παραδοχή αυτή χρησιμοποιείται αφενός διότι, όπως αναφέρθηκε, σκοπός της εργασίας είναι η μελέτη της απόδοσης του μηχανισμού και όχι η ακρίβεια στις προβλέψεις και αφετέρου γιατί σύμφωνα με δοκιμές που έγιναν δεν επηρεάζει σημαντικά την συνολική ακρίβεια και συγκεκριμένα την βελτιώνει ελάχιστα.

3.3. Περιορισμοί

Παρακάτω περιγράφονται κάποιοι περιορισμοί που παρουσιάστηκαν κατά την υλοποίηση του μηχανισμού και σχετίζονται με το eBPF και ο τρόπος αντιμετώπισής τους.

Δεκαδικοί αριθμοί στο eBPF

Για να υπολογιστεί η τελική πιθανότητα για το αν ένα όνομα είναι έγκυρο ή άκυρο πολλαπλασιάζονται οι κατάλληλες τιμές των χαρακτηριστικών που έχουν παραχθεί κατά την εκπαίδευση. Οι πιθανότητες είναι αριθμοί μικρότεροι της μονάδας και ως εκ τούτου αναπαριστάνονται σαν δεκαδικοί αριθμοί. Ωστόσο, στο eBPF δεν παρέχεται υποστήριξη για δεκαδικούς αριθμούς επομένως δεν μπορούν να χρησιμοποιηθούν με την μορφή αυτή.

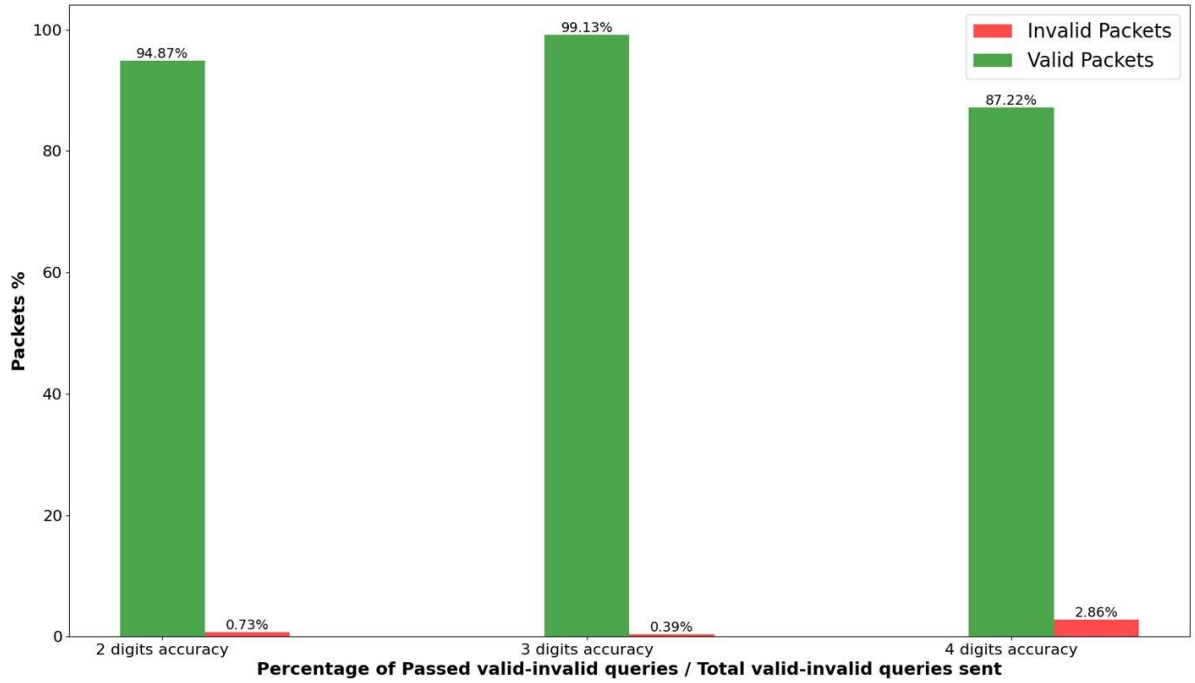
Η προσέγγιση που εφαρμόστηκε σε αυτό το πρόβλημα είναι η αναπαράσταση αυτών των δεκαδικών αριθμών σαν ακέραιους με την χρήση των τριών πρώτων δεκαδικών τους ψηφίων μόνο. Ουσιαστικά, κάθε ένας από αυτούς τους δεκαδικούς αριθμούς πριν αποθηκευτεί στους χάρτες eBPF πολλαπλασιάζεται με το 1000. Τα υπόλοιπα

δεκαδικά ψηφία αγνοούνται. Επίσης, εάν ο αριθμός είναι μικρότερος από 0,001, για να μην προκύψει μηδενική τελική πιθανότητα, αποθηκεύεται με την τιμή 1, έτσι ώστε να έχει την μικρότερη δυνατή πιθανότητα (αφού οι δυνατές τιμές θα είναι μεταξύ 1 και 999). Εάν χρησιμοποιηθεί η τιμή 0, η ακρίβεια μειώνεται περισσότερο.

Αυτό προφανώς έχει επιπτώσεις στην ακρίβεια των προβλέψεων, ωστόσο η μείωση της ακρίβειας είναι μόλις 0,1 – 0,2%, που σημαίνει ότι είναι ανεκτή.

Η επιλογή των 3 πρώτων δεκαδικών ψηφίων έγινε για τους εξής λόγους:

- Η χρήση περισσότερων δεκαδικών αυξάνει την ακρίβεια, ωστόσο αυξάνει και τις τιμές στους χάρτες. Μεγαλύτερες τιμές στους χάρτες οδηγούν σε υπερχειλίσεις στον πυρήνα κατά τον υπολογισμό της τελικής πιθανότητας (υπάρχει όριο στο πόσο μεγάλοι αριθμοί μπορούν να αποθηκευτούν στις μεταβλητές) και εξ αιτίας αυτού μειώνεται η ακρίβεια στις προβλέψεις.
- Η χρήση λιγότερων ψηφίων μειώνει περισσότερο την ακρίβεια.

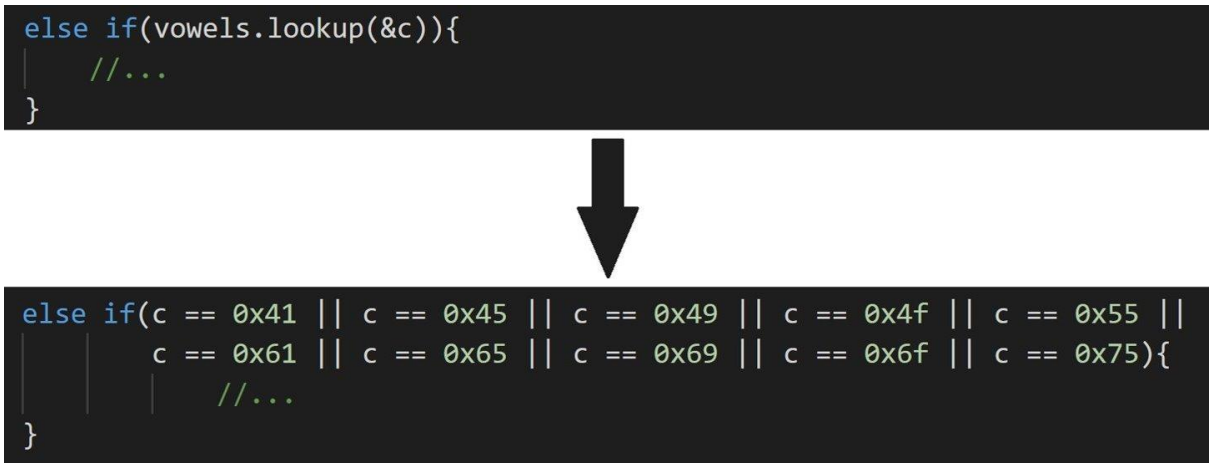


Σχήμα 3.2 - Ποσοστό έγκυρων/άκυρων ερωτημάτων που ταξινομούνται σαν έγκυρα ονόματα και προωθούνται από τον Resolver.

Στο διάγραμμα φαίνονται τα ποσοστά των έγκυρων και άκυρων πακέτων που διέρχονται από το φίλτρο του XDP. Όταν χρησιμοποιούνται τα 3 πρώτα δεκαδικά ψηφία από το σύνολο των έγκυρων ερωτημάτων ταξινομείται σωστά το 99.13% αυτών και από τα άκυρα ερωτήματα μόλις το 0.39% ταξινομείται λάθος (ως έγκυρα) και διέρχεται από το επίπεδο του XDP. Στις περιπτώσεις χρήσης τεσσάρων ή δύο δεκαδικών ψηφίων παρατηρούνται χειρότερα αποτελέσματα, οπότε επιλέγεται ως καλύτερη λύση η χρήση τριών δεκαδικών ψηφίων.

Όριο εντολών

Όπως αναφέρθηκε, το όριο εντολών μπορεί να δημιουργήσει προβλήματα στην απόδοση του μηχανισμού. Συγκεκριμένα, αναφέρθηκε το παράδειγμα της χρήσης αναζήτησης σε χάρτες για εξοικονόμηση εντολών που έχει σαν αποτέλεσμα την μείωση της απόδοσης του μηχανισμού.



Σχήμα 3.3 - Έλεγχος για το αν ένας χαρακτήρας είναι φωνήεν ή όχι, με αναζήτηση σε χάρτη eBPF και με σύγκριση με όλα τα φωνήεντα.

Το όριο εντολών είναι επίσης πρόβλημα όταν χρειάζεται να υλοποιηθεί μια σύνθετη συνάρτηση (π.χ. με βρόχους, αναδρομές). Τέτοιες συναρτήσεις μπορεί να βελτιώνουν την απόδοση ενός προγράμματος αλλά αν χρειάζονται πολλές εντολές για να υλοποιηθούν και ξεπεράσουν το όριο εντολών, θα οδηγήσουν στην απόρριψη του προγράμματος από τον επαληθευτή και ως αποτέλεσμα στην αδυναμία φόρτωσής του.

4. Περιγραφή και αξιολόγηση πειραμάτων

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση του εικονικού εργαστηρίου που δημιουργήθηκε για την εκτέλεση των πειραμάτων, περιγραφή του τρόπου διεξαγωγής τους καθώς ανάλυση των αποτελεσμάτων τους.

4.1. Εικονικό εργαστήριο

Το εικονικό εργαστήριο στήθηκε σε περιβάλλον VirtualBox (6.1.20) [39] και περιλαμβάνει συνολικά τρία εικονικά μηχανήματα (Virtual Machines ή VMs) τα οποία επικοινωνούν μεταξύ τους μέσω εσωτερικής δικτύωσης (εικονικό τοπικό δίκτυο). Τα χαρακτηριστικά των μηχανημάτων αυτών περιγράφονται στη συνέχεια.

Virtual Machines

Attacker

Η κακόβουλη κίνηση καθώς και η καλόβουλη κίνηση έχουν σαν πηγή αυτό το εικονικό μηχάνημα (για ευκολία, θεωρούμε πως και οι δύο στέλνονται από τον Attacker). Χρησιμοποιεί την διανομή Linux, Kali Linux [40] και διαθέτει 4 επεξεργαστικούς πυρήνες και 4 GB φυσικής μνήμης. Με την χρήση του εργαλείου tcprewrite [77] προετοιμάζονται τα πακέτα προς αποστολή στον αναδρομικό εξυπηρετητή, ρυθμίζοντας κατάλληλα τις διευθύνσεις IP και MAC καθώς και υπολογίζοντας ξανά το checksum του πακέτου. Στην συνέχεια, με το tcpreplay [41] ξεκινάει η επίθεση στέλνοντας ταυτόχρονα τα πακέτα καλόβουλης και κακόβουλης κίνησης, με τον επιθυμητό ρυθμό, στον Resolver [49].

Resolver

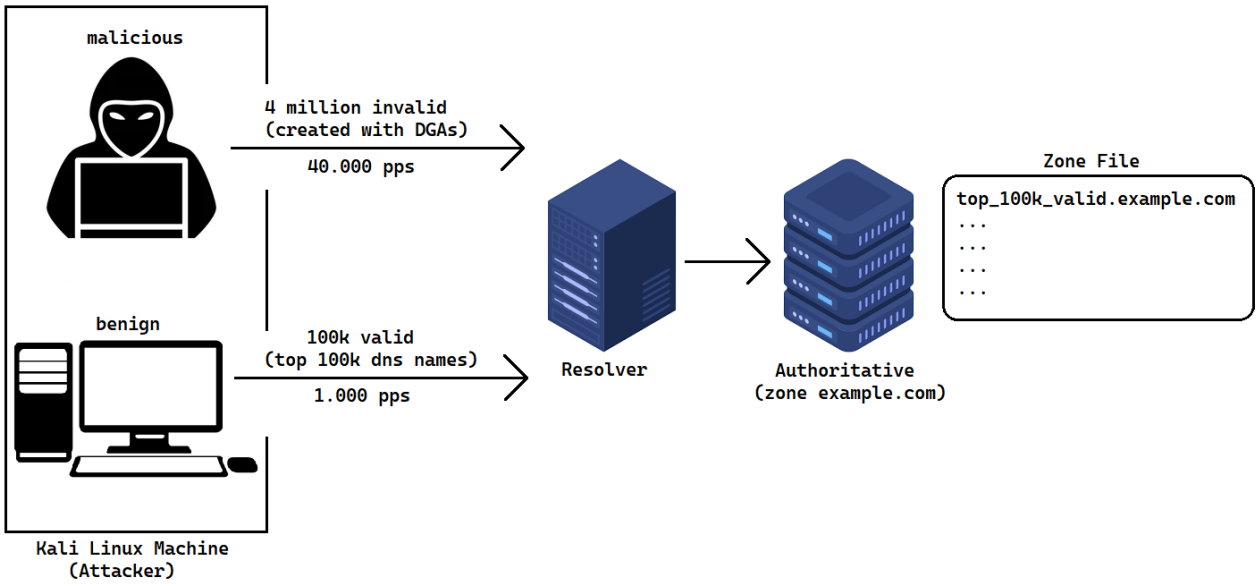
Ο αναδρομικός εξυπηρετητής χρησιμοποιεί την ένατη έκδοση του λογισμικού BIND και λειτουργεί σαν προωθητής των ερωτημάτων που του γίνονται στον Authoritative εξυπηρετητή. Χρησιμοποιεί την διανομή Linux, Ubuntu (18.04) [42] και είναι το μηχάνημα στο οποίο έχει εφαρμοστεί ο μηχανισμός άμυνας που αναπτύχθηκε. Διαθέτει επίσης 4 επεξεργαστικούς πυρήνες και 4 GB φυσικής μνήμης.

Authoritative

Ο Authoritative, εξυπηρετητής αποτελεί ένα αντίγραφο του εικονικού μηχανήματος του αναδρομικού εξυπηρετητή με τη διαφορά ότι έχει ρυθμιστεί να είναι υπεύθυνος για την ζώνη example.com. Δημιουργήθηκε ένα αρχείο ζώνης στον Authoritative

εξυπηρετητή το οποίο περιέχει μια λίστα από εγγραφές με έγκυρα ονόματα (τα κορυφαία 100.000 DNS ονόματα από μία λίστα με τα κορυφαία 10.000.000 ονόματα [44]). Έτσι, αυτά τα ονόματα ανήκουν στην ζώνη example.com και για κάθε ένα από αυτά τα ονόματα ο Authoritative εξυπηρετητής είναι υπεύθυνος να απαντάει σε ερωτήματα που του γίνονται. Η καλόβουλη κίνηση αποτελείται από αυτά τα ονόματα. Αυτό το εικονικό μηχάνημα διαθέτει 2 επεξεργαστικούς πυρήνες και 2 GB φυσικής μνήμης. Χρησιμοποιεί επίσης την ένατη έκδοση του λογισμικού BIND.

Παρακάτω απεικονίζεται το εικονικό εργαστήριο.



Σχήμα 4.1 - Διάταξη εικονικού εργαστηρίου σε VirtualBox.

Οι ρυθμοί αποστολής πακέτων καθώς και τα σύνολα έγκυρων/άκυρων ονομάτων που χρησιμοποιήθηκαν θα επεξηγηθούν στην συνέχεια.

4.2. Ανάλυση διεξαγωγής πειραμάτων

Εκτελέστηκαν 2 επιθέσεις DNS Water Torture με στόχο τον Authoritative Server της ζώνης example.com. Το εικονικό μηχάνημα του επιτιθέμενου στέλνει ταυτόχρονα καλόβουλη και κακόβουλη κίνηση χρησιμοποιώντας τα σύνολα ονομάτων και τους ρυθμούς κίνησης που αναφέρονται στη συνέχεια.

Σύνολα άκυρων/έγκυρων ονομάτων

Κάθε σύνολο άκυρων ονομάτων αποτελείται από 4 εκατομμύρια μη επαναλαμβανόμενα ονόματα με μέσο μήκος 24 για την πρώτη επίθεση και 14 για την δεύτερη. Αυτή είναι και η μόνη διαφορά μεταξύ των 2 επιθέσεων. Αυτό συμβαίνει καθώς έτσι εξετάζεται η απόδοση του μηχανισμού σε σύνολα ονομάτων που έχουν μέσο μήκος παρόμοιο με αυτό έγκυρων ονομάτων, κάνοντας έτσι την ταξινόμηση πιο δύσκολη. Το μέσο μήκος των 10.000.000 κορυφαίων ονομάτων [44] είναι περίπου 11, οπότε το δεύτερο σύνολο άκυρων ονομάτων, που έχει μέσο μήκος 14 είναι αρκετά κοντά σε αυτό. Αυτό έχει σαν αποτέλεσμα περισσότερα άκυρα πακέτα να ταξινομούνται σαν έγκυρα και να προωθούνται τελικά στον Authoritative Server. Στην δεύτερη επίθεση λοιπόν, εξετάζεται κατά πόσο η αύξηση των invalid misclassifications θα επηρεάσει την απόδοση του μηχανισμού και τι αντίκτυπο θα έχει στον αναδρομικό καθώς και στον Authoritative εξυπηρετητή.

Τα άκυρα ονόματα δημιουργήθηκαν από το συνδυασμό 5 διαφορετικών αλγορίθμων παραγωγής Domain (Domain Generation Algorithms ή DGAs) [43]. Οι DGAs που χρησιμοποιήθηκαν είναι οι :

1) Corebot
2) Morenodownloader
3) Newgoz
4) Reconyc
5) Qadars

Αυτοί οι αλγόριθμοι χρησιμοποιούνται από τα bots για να επικοινωνούν με τους χειριστές τους. Δημιουργώντας τυχαία Domain names, προσπαθούν να επικοινωνήσουν με αυτά για να λάβουν εντολές. Κάποια από αυτά τα Domains, ανήκουν στον χειριστή και μέσω αυτού στέλνει τις εντολές. Ο βαθμός τυχαιότητας σε αυτά τα ονόματα είναι αρκετά μεγάλος και είναι μη επαναλαμβανόμενα. Ταιριάζουν στα χαρακτηριστικά ονομάτων που μπορούν να χρησιμοποιηθούν σε μια DNS Water Torture επίθεση και για αυτό χρησιμοποιήθηκαν σαν σύνολα ονομάτων των επιθέσεων.

Και στις 2 επιθέσεις χρησιμοποιείται το ίδιο σύνολο έγκυρων ονομάτων και αυτό αποτελείται από τα 100.000 πιο γνωστά ονόματα στο διαδίκτυο. Αυτά τα ονόματα ανήκουν στα 10.000.000 κορυφαία ονόματα στο διαδίκτυο [44]. Χρησιμοποιήθηκαν αυτά τα ονόματα για την καλόβουλη κίνηση καθώς κάθε ένα από αυτά θα μπορούσε να αποτελεί έγκυρο ερώτημα από πραγματικούς χρήστες.

Ρυθμοί έγκυρης/άκυρης κίνησης

Ο ρυθμός αποστολής άκυρων ονομάτων, όπως φαίνεται και στο σχήμα τέθηκε ίσος με 40.000 πακέτα ανά δευτερόλεπτο, ενώ ο ρυθμός αποστολής πακέτων έγκυρων ονομάτων τέθηκε ίσος με 1000 πακέτα ανά δευτερόλεπτο.

Ο ρυθμός αποστολής της κακόβουλης κίνησης πήρε αυτή την τιμή καθώς είναι ικανός να προκαλέσει την απόρριψη καλόβουλων πακέτων λόγω του κατακλυσμού από πακέτα στον αναδρομικό εξυπηρετητή σε περίπτωση απουσίας αμυντικού μηχανισμού και έτσι να δημιουργήσει προβλήματα στη σωστή λειτουργία του Authoritative εξυπηρετητή αφού χάνει ένα μεγάλο ποσοστό της καλόβουλης κίνησης και καταναλώνει τους πόρους του σε εξυπηρέτηση άκυρων ερωτημάτων.

Ο ρυθμός των έγκυρων ερωτημάτων πήρε την τιμή 1000, αφενός για να υπάρχει μια αναλογία 40:1 σε σχέση με την κακόβουλη κίνηση, αφού σε μια πραγματική επίθεση ο ρυθμός των άκυρων μηνυμάτων είναι πολύ μεγαλύτερος από αυτόν των έγκυρων και αφετέρου διότι σύμφωνα με στατιστικά δεδομένα από DNS Servers [45] οι πραγματικοί ρυθμοί ερωτημάτων ανά δευτερόλεπτο έχουν παρόμοιες τιμές.

Συλλογή αποτελεσμάτων

Χρησιμοποιήθηκε το εργαλείο tcpdump [46] για την καταγραφή των εισερχόμενων πακέτων στον Authoritative εξυπηρετητή και το Wireshark [47] για την ανάλυση αυτών των πακέτων και τον υπολογισμό των άκυρων και έγκυρων ερωτημάτων. Συγκεκριμένα, στον Authoritative Server γίνεται καταγραφή των πακέτων με IP παραλήπτη ίδια με αυτήν του αναδρομικού εξυπηρετητή. Τα ερωτήματα τα οποία δέχεται είναι ίσα σε αριθμό με αυτά στα οποία απαντάει (δεν χάνει πακέτα κατά την διαδικασία της απάντησής τους). Επομένως, σε όσα ερωτήματα στέλνει ο αναδρομικός εξυπηρετητής αυτά στα οποία απαντάει με NXDomain ανήκουν στην κακόβουλη κίνηση, ενώ τα υπόλοιπα στην καλόβουλη.

Τέλος, συλλέγονται στατιστικά από την χρήση της RAM και του επεξεργαστή, στον αναδρομικό και στον Authoritative εξυπηρετητή. Αυτό επιτυγχάνεται με την χρήση ενός custom script σε python το οποίο καταγράφει το ποσοστό χρήσης φυσικής μνήμης και επεξεργαστή κάθε δευτερόλεπτο.

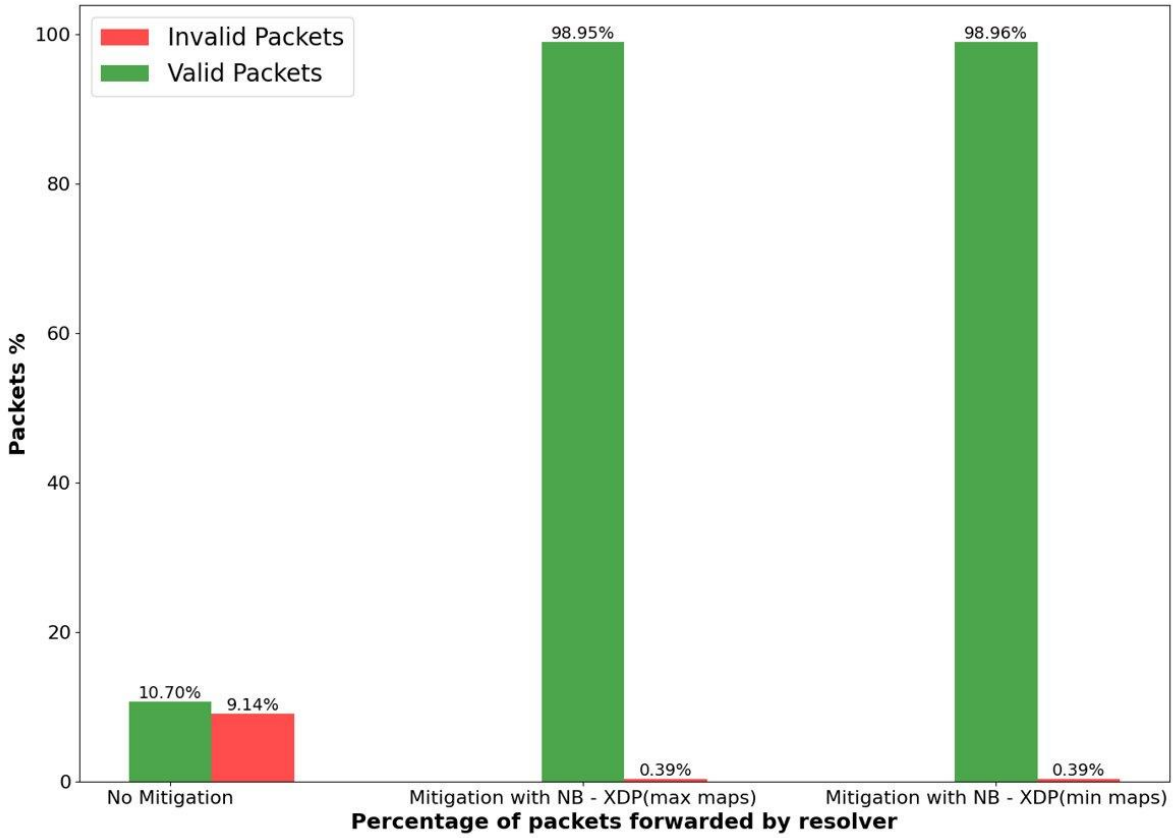
4.3. Παρουσίαση και ανάλυση αποτελεσμάτων

Αποτελέσματα πρώτου πειράματος (μέσο μήκος άκυρων ονομάτων : 24)

Στο πρώτο πείραμα ο επιτιθέμενος στέλνει, με ρυθμό 40.000 πακέτα ανά δευτερόλεπτο, άκυρα ονόματα με μέσο μήκος 24 χαρακτήρες. Το συγκεκριμένο dataset, με την εκπαίδευση που έγινε στον Naive Bayes, παρουσίασε ποσοστό misclassification, δηλαδή λάθος ταξινομήσεων (ταξινόμηση ως valid, ενώ το όνομα ήταν invalid), ίσο με 0.4%. Παράλληλα, όπως αναφέρθηκε, από το εικονικό μηχάνημα του επιτιθέμενου στέλνεται ταυτόχρονα και η καλόβουλη κίνηση με ρυθμό 1000 πακέτα το δευτερόλεπτο. Τα κορυφαία 100.000 dns ονόματα, είχαν ποσοστό misclassification 0.95%.

1)

Στο παρακάτω διάγραμμα φαίνονται τα ποσοστά των έγκυρων/άκυρων πακέτων που προωθήθηκαν από τον αναδρομικό εξυπηρετητή στον Authoritative Server και απαντήθηκαν από αυτόν, ως προς τον συνολικό αριθμό έγκυρων/άκυρων πακέτων που στάλθηκαν από τον επιτιθέμενο. Σκοπός του διαγράμματος είναι να δείξει κατά πόσο αλλάζει το ποσοστό των έγκυρων και άκυρων ερωτημάτων, που φτάνουν στον Authoritative Server, όταν ενεργοποιείται ο αμυντικός μηχανισμός.



Σχήμα 4.2 - Ποσοστό πακέτων που στάλθηκαν από τον Resolver (Πείραμα 1).

Στο πρώτο ζευγάρι στηλών, φαίνονται τα ποσοστά όταν δεν υπάρχει κάποιος αμυντικός μηχανισμός. Ο Resolver, επομένως καταφέρνει να προωθήσει μόλις 10.7% των αρχικών valid ερωτημάτων και παράλληλα στέλνει 9.14% των invalid. Υπενθυμίζεται ότι τα αρχικά valid πακέτα ήταν 100.000 σε πλήθος , ενώ τα invalid 4.000.000. Αυτό σημαίνει ότι ο Authoritative εξυπηρετητής λαμβάνει μεγάλο όγκο άκυρων ονομάτων και λόγω κατακλυσμού πακέτων στον Resolver, φτάνει ένα πολύ μικρό μέρος των έγκυρων ερωτημάτων. Επομένως, ο επιτιθέμενος επιτυγχάνει τον σκοπό του.

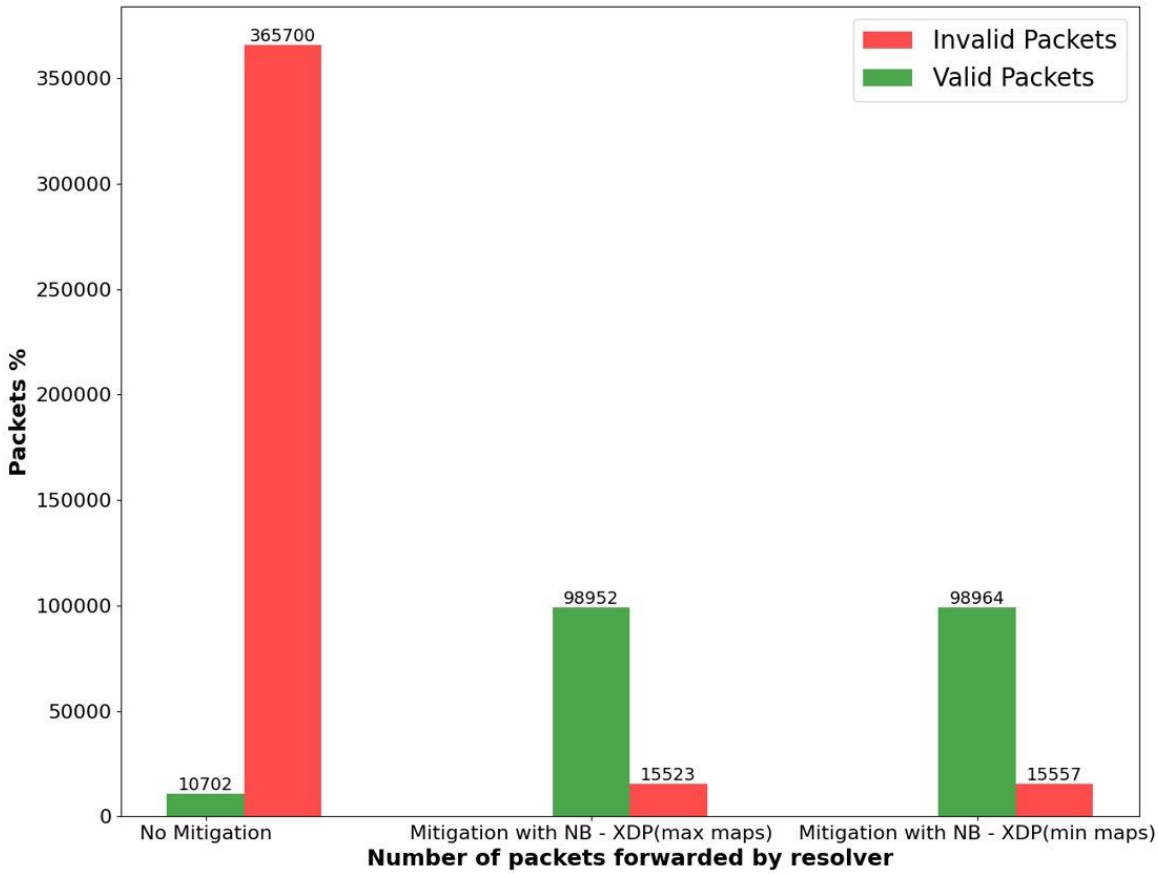
Αυτό αλλάζει, ωστόσο στο δεύτερο ζευγάρι στηλών, δηλαδή με την χρήση του αμυντικού μηχανισμού (naive bayes σε XDP). Παρατηρείται μια πολύ μεγάλη αύξηση

στο ποσοστό των έγκυρων ερωτημάτων που εξυπηρετούνται από τον Authoritative Server και ταυτόχρονα μία μείωση στο ποσοστό των πακέτων της κακόβουλης κίνησης. Τα 98.95% των valid πακέτων προωθείται κανονικά από τον Resolver, ενώ μόλις το 0.39% των invalid φτάνει τον Authoritative. Ο μηχανισμός πετυχαίνει τον σκοπό του σε αρκετά καλό βαθμό καθώς στο επίπεδο του Resolver απορρίπτονται τα περισσότερα άκυρα πακέτα, εξοικονομώντας έτσι πόρους για να επεξεργαστεί τα έγκυρα. Έτσι δικαιολογούνται και οι αλλαγές στα ποσοστά.

Τέλος, στην περίπτωση χρήσης ελάχιστων αναζητήσεων σε χάρτες eBPF (τρίτο ζευγάρι στηλών), φαίνεται μία μικρή αύξηση στο ποσοστό των valid πακέτων που εξυπηρετούνται από τον Authoritative Server και αυτό οφείλεται στην βελτίωση της απόδοσης του μηχανισμού. Η βελτίωση αυτή είναι μικρή, ωστόσο δείχνει πως λιγότερες αναζητήσεις αποδίδουν καλύτερα.

2)

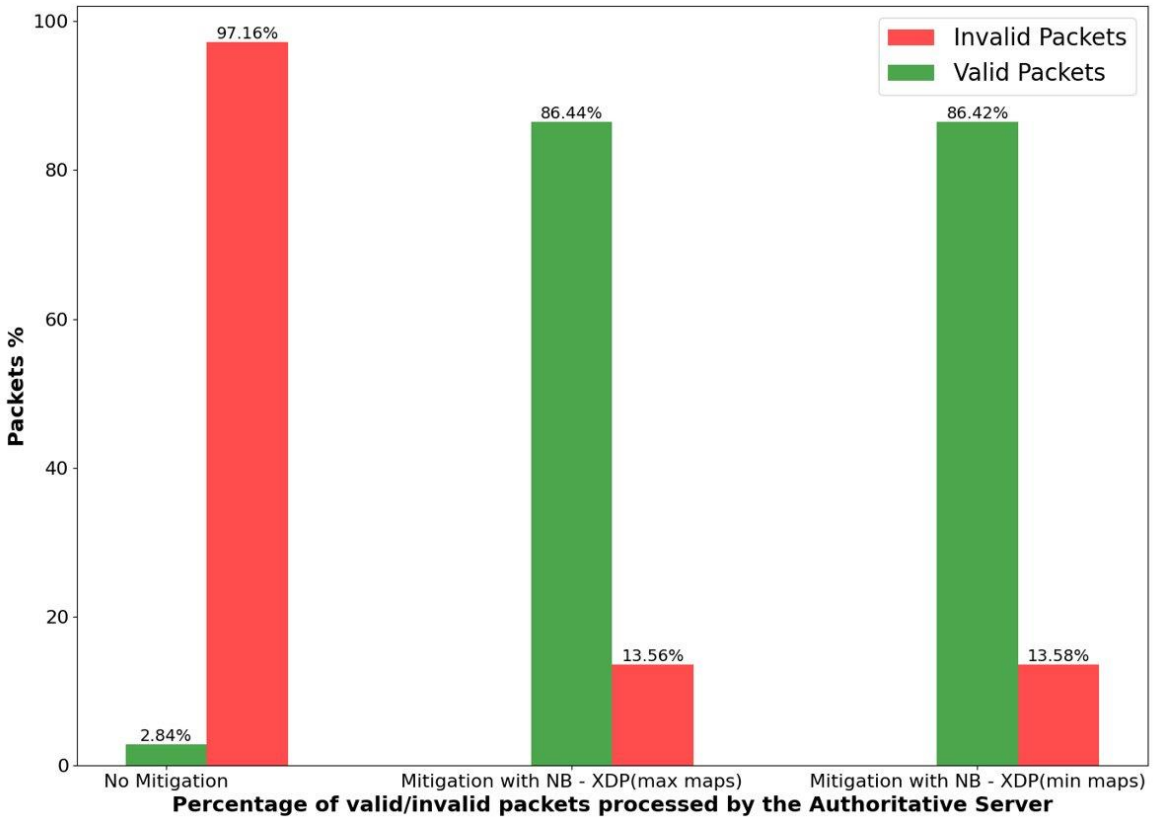
Το επόμενο διάγραμμα είναι παρόμοιο με το προηγούμενο, με την διαφορά ότι σε αυτό φαίνονται οι αριθμοί των πακέτων και όχι τα ποσοστά τους. Φαίνεται λοιπόν, πως σε περίπτωση απουσίας αμυντικού μηχανισμού, 365.700 (από τα 4.000.000) πακέτα κακόβουλης κίνησης προωθούνται από τον αναδρομικό εξυπηρετητή και μόλις 10.702 (από τα 100.000) έγκυρα πακέτα φτάνουν στον Authoritative. Με την χρήση του Naive Bayes σε XDP όμως, τα άκυρα πακέτα που προωθούνται είναι 23 φορές μικρότερα σε πλήθος και αυτό δείχνει την ικανότητα του XDP να απορρίπτει με μεγάλες ταχύτητες μεγάλο όγκο άκυρων πακέτων και παράλληλα να εξυπηρετεί τα έγκυρα χωρίς σοβαρές απώλειες.



Σχήμα 4.3 - Αριθμός πακέτων που στάλθηκαν από τον Resolver (Πείραμα 1).

3)

Στη συνέχεια φαίνονται τα ποσοστά των πακέτων που απαντήθηκαν από τον Authoritative Server. Από το σύνολο των πακέτων που εξυπηρέτησε ο Authoritative (στην πρώτη περίπτωση) το 97.16% αφορούσε κακόβουλα πακέτα ενώ μόνο το 2.84% ήταν καλόβουλη κίνηση. Η επίθεση επομένως ανάγκασε τον εξυπηρετητή να χρησιμοποιήσει τους πόρους για την επεξεργασία κυρίως κακόβουλων πακέτων. Στην περίπτωση χρήσης του αμυντικού μηχανισμού όμως, υπάρχει μια μεγάλη αλλαγή σε αυτά τα ποσοστά καθώς το 86.44% των ερωτημάτων στα οποία απάντησε ήταν μέρος της καλόβουλης κίνησης. Επομένως, χάρη στην χρήση του μηχανισμού ο Authoritative Server μειώνει την χρήση των πόρων του (αφού φτάνουν συνολικά πολύ λιγότερα πακέτα) και εξυπηρετεί κυρίως τα έγκυρα ερωτήματα.

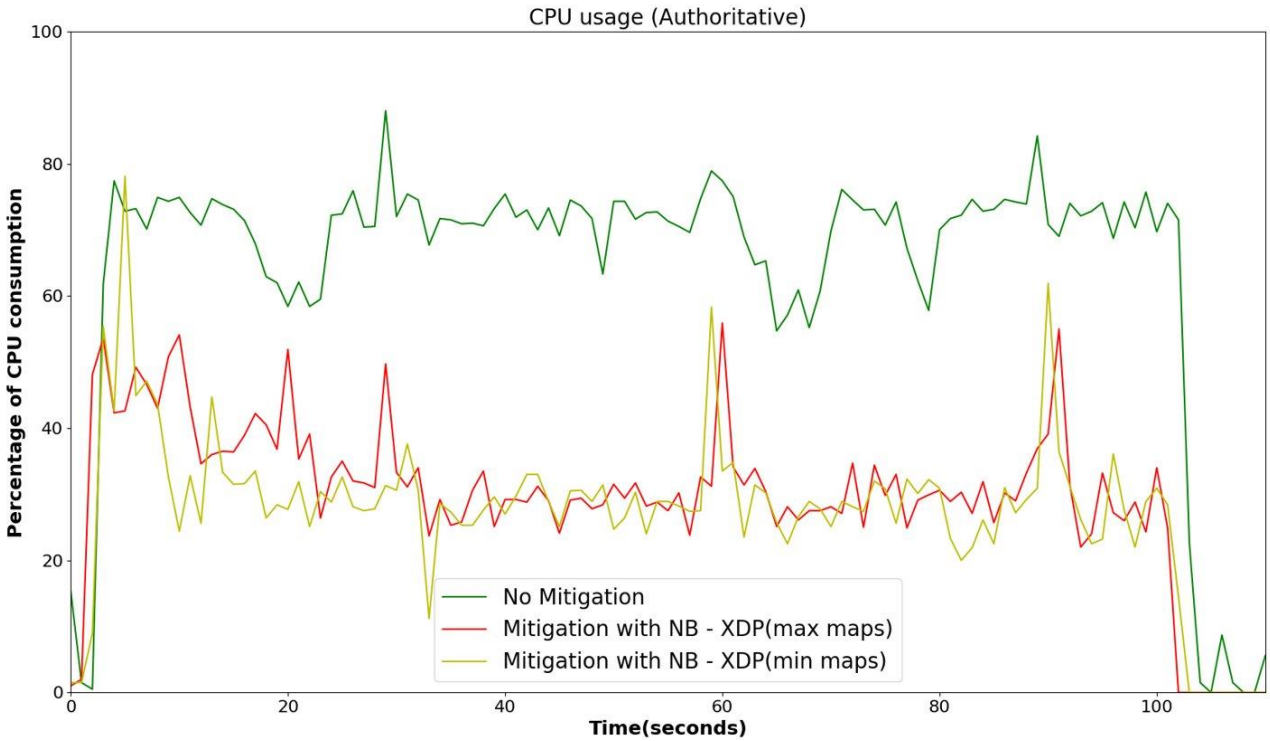


Σχήμα 4.4 - Ποσοστά έγκυρων/άκυρων ερωτημάτων σε σχέση με τα συνολικά ερωτήματα τα οποία επεξεργάστηκε ο Authoritative Server (Πείραμα 1).

4)

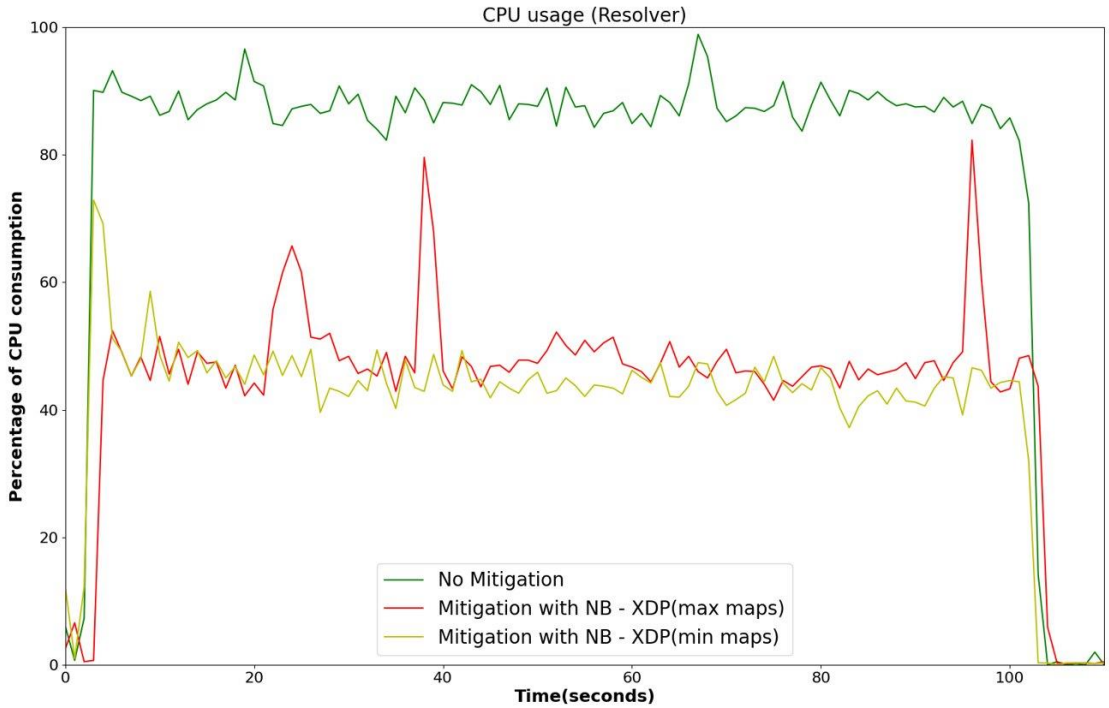
Όπως έχει αναφερθεί κάθε ερώτημα που στέλνεται στον Authoritative Server καταναλώνει πόρους του επεξεργαστή του. Παρακάτω φαίνεται πως στην περίπτωση απουσίας αμυντικού μηχανισμού (πράσινη γραμμή) το ποσοστό χρήσης του επεξεργαστή αγγίζει το 80% και σύμφωνα με το παραπάνω διάγραμμα το μεγαλύτερο μέρος αυτής της κατανάλωσης σχετίζεται με άκυρα ερωτήματα, κάτι που αποτελεί την κύρια συνέπεια της επίθεσης Water Torture.

Με την χρήση του Naive Bayes σε XDP, η χρήση του επεξεργαστή μειώνεται κατά περίπου 50%. Επομένως , ο Authoritative Server δεν σπαταλά τους πόρους του στην εξυπηρέτηση ερωτημάτων της κακόβουλης κίνησης και συγχρόνως καταναλώνει ένα μικρό ποσοστό τους με αποτέλεσμα να μην παρουσιάζει πρόβλημα στην λειτουργία του. Σε μια πραγματική επίθεση όπου το ποσοστό χρήσης έφτανε συνεχώς στο 100% (σε περίπτωση που δεν γινόταν χρήση κάποιου αμυντικού μηχανισμού), ο Authoritative εξυπηρετητής πιθανόν να μην μπορούσε να εκτελέσει αποτελεσματικά άλλες διεργασίες και αυτό θα δημιουργούσε σοβαρά προβλήματα στην λειτουργία του.



Σχήμα 4.5 - Ποσοστό χρήσης επεξεργαστή του Authoritative, για τις 3 καταστάσεις λειτουργίας του Resolver (Πείραμα 1).

Η χρήση του μηχανισμού έχει αντίστοιχα οφέλη και στην περίπτωση του αναδρομικού εξυπηρετητή. Η χρήση του CPU φτάνει το 90% στο πρώτο σενάριο και αυτό συμβαίνει διότι ο Resolver καλείται να εξυπηρετήσει έναν μεγάλο όγκο ερωτημάτων (κυρίως invalid). Ωστόσο στις περιπτώσεις που γίνεται χρήση του Naive Bayes σε XDP, έχουμε μείωση κατά περίπου 50% και έτσι ο Resolver μπορεί να εξυπηρετήσει σχεδόν όλα τα καλόβουλα πακέτα.



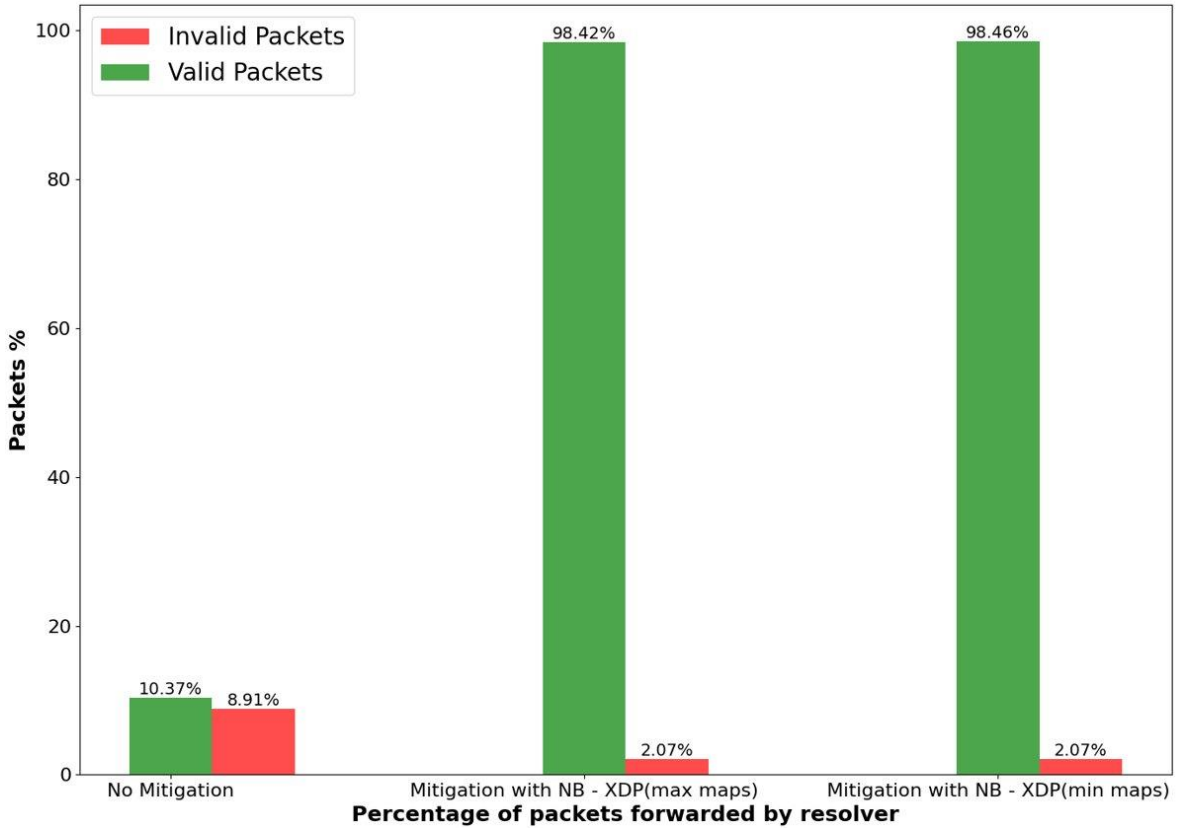
Σχήμα 4.6 - Ποσοστό χρήσης επεξεργαστή του Resolver, για τις 3 καταστάσεις λειτουργίας του (Πείραμα 1).

Αποτελέσματα δεύτερου πειράματος (μέσο μήκος άκυρων ονομάτων : 14)

Στο δεύτερο πείραμα ο επιτιθέμενος στέλνει και πάλι, με ρυθμό 40.000 πακέτα ανά δευτερόλεπτο, άκυρα ονόματα με μέσο μήκος 14 χαρακτήρες. Το ποσοστό των misclassifications αυξάνεται σε 2%. Αυτό συμβαίνει γιατί όπως αναφέρθηκε το μέσο μήκος ονομάτων είναι πλέον αρκετά κοντά σε αυτό των κανονικών ονομάτων οπότε η διάκριση μεταξύ έγκυρου και άκυρου ονόματος γίνεται πιο δύσκολη. Ωστόσο, όπως θα φανεί και στην συνέχεια αυτό δεν επιβαρύνει τους εξυπηρετητές σε βαθμό που να προκαλεί απώλεια καλόβουλων πακέτων.

1)

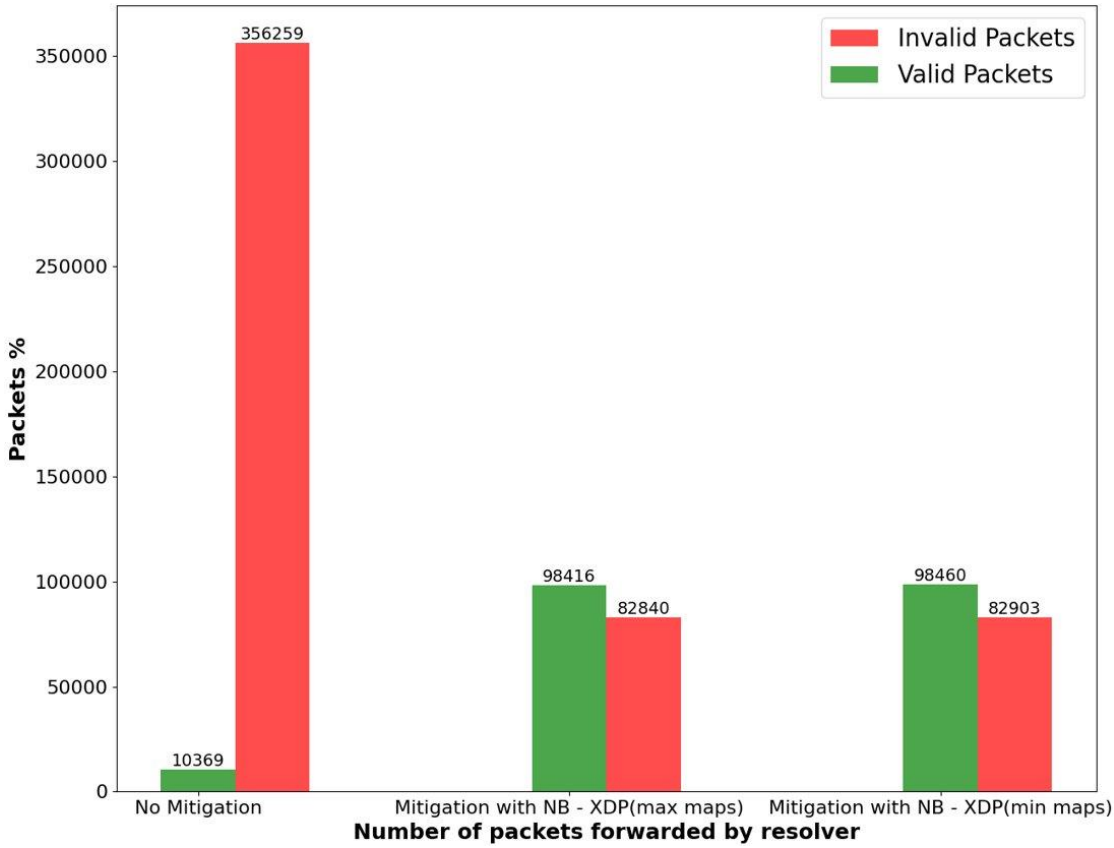
Όπως και στο αντίστοιχο διάγραμμα του πρώτου πειράματος το ποσοστό έγκυρων πακέτων αυξάνεται σε μεγάλο βαθμό όταν χρησιμοποιείται ο αμυντικός μηχανισμός. Η διαφορά με το πρώτο πείραμα φαίνεται στο ποσοστό των πακέτων της κακόβουλης κίνησης που αυξάνεται σε 2.07% και οφείλεται στην αύξηση των misclassifications του Naive Bayes.



Σχήμα 4.7 - Ποσοστό πακέτων που στάλθηκαν από τον Resolver (Πείραμα 2).

2)

Όπως φαίνεται στο διάγραμμα που ακολουθεί τα συνολικά πακέτα που προωθούνται από τον Resolver και περιέχουν άκυρα ονόματα (από τα 4.000.000 άκυρα πακέτα που στάλθηκαν) μειώνονται από 356.259 σε 82.840. Παράλληλα, το μεγαλύτερο ποσοστό των valid ερωτημάτων εξακολουθεί να εξυπηρετείται κανονικά από τον Authoritative Server.

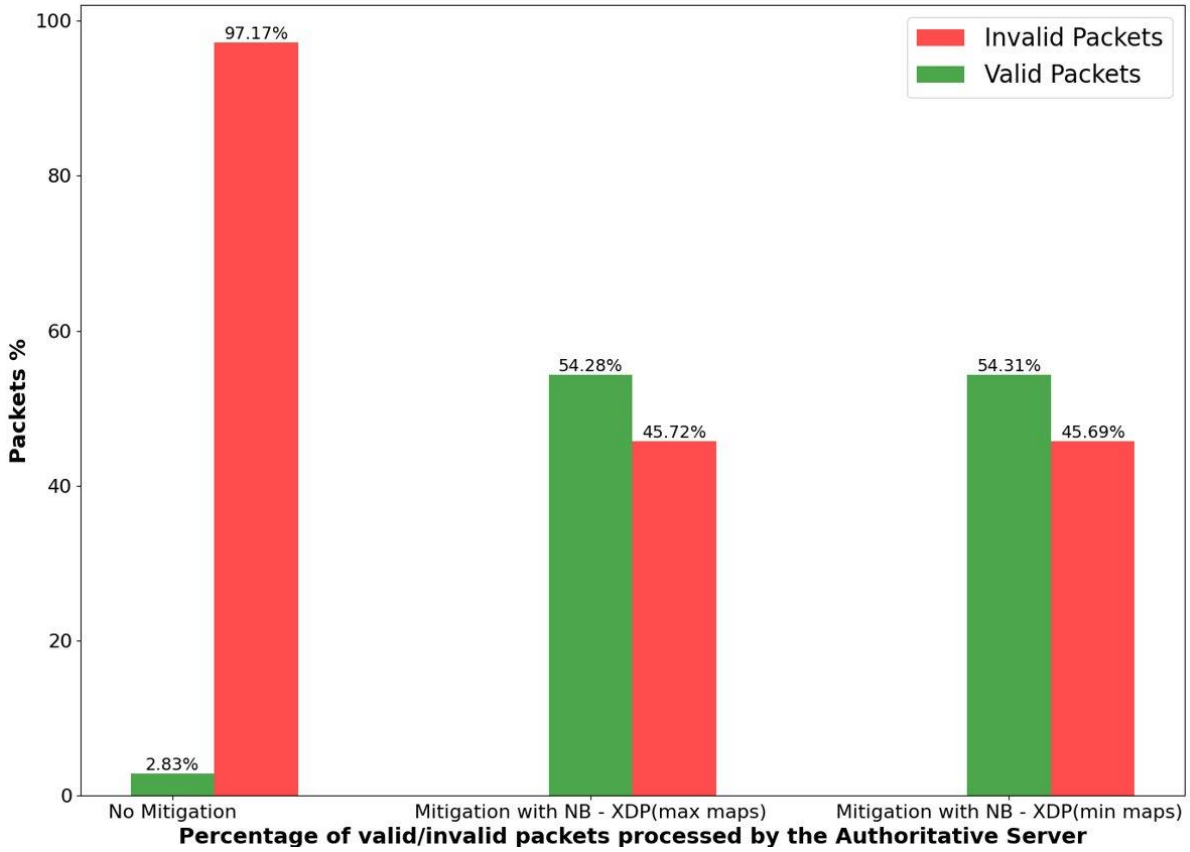


Σχήμα 4.8 - Αριθμός πακέτων που στάλθηκαν από τον Resolver (Πείραμα 2).

Η μικρή αύξηση των πακέτων που στέλνονται από τον Resolver, στην περίπτωση ελάχιστων αναζητήσεων σε χάρτες eBPF, δείχνει και πάλι πως η απόδοση του μηχανισμού βελτιώνεται αφού καταφέρνει να στείλει περισσότερα πακέτα στο ίδιο χρονικό διάστημα , σε σχέση με την περίπτωση που πραγματοποιούνται περισσότερες αναζητήσεις σε eBPF maps.

3)

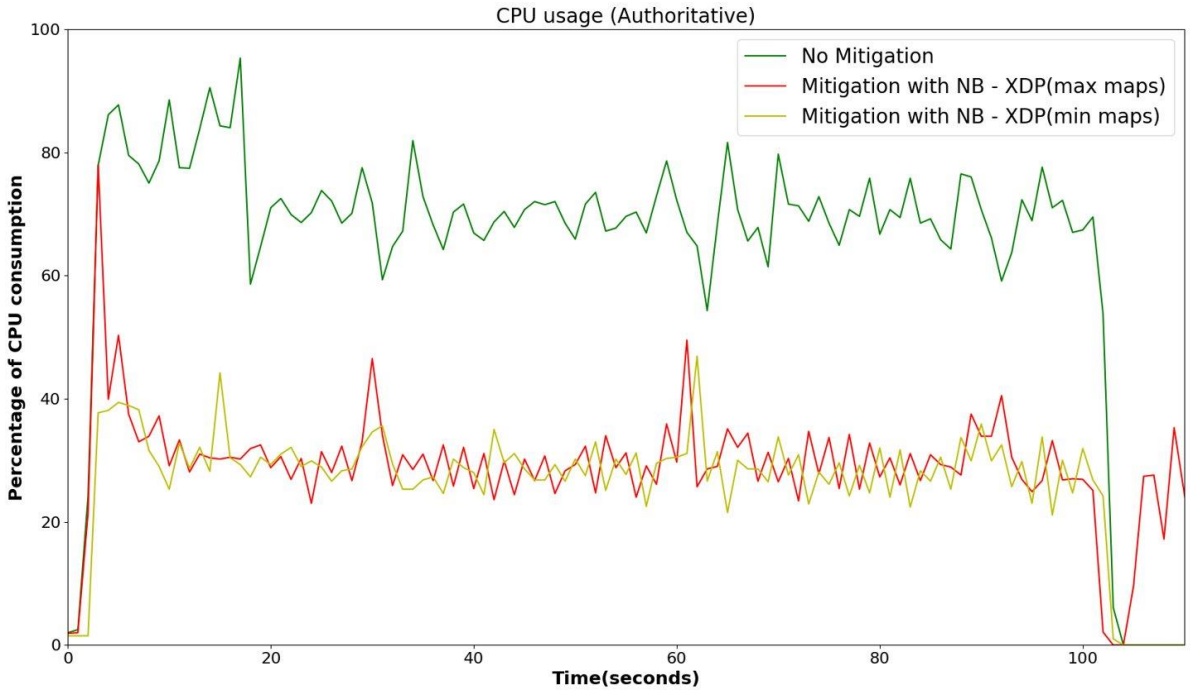
Στο επόμενο διάγραμμα βλέπουμε για το δεύτερο πείραμα, τα ποσοστά valid/invalid ερωτημάτων σε σχέση με τα συνολικά ερωτήματα που απάντησε ο Authoritative. Παρόμοια, με το πρώτο πείραμα, στην πρώτη περίπτωση, ο επιτιθέμενος επιτυγχάνει τον στόχο του αφού το 97.13% των ερωτημάτων που απάντησε ο εξυπηρετητής ήταν invalid. Στην περίπτωση χρήσης του αμυντικού μηχανισμού, όπως ήταν αναμενόμενο παρατηρείται αύξηση του ποσοστού των άκυρων, σε σχέση με το πρώτο πείραμα. Ωστόσο, όπως φαίνεται και στο επόμενο διάγραμμα (χρήση CPU), αυτό δεν έχει κάποια επίπτωση στην κατανάλωση των πόρων του εξυπηρετητή καθώς ο αριθμός των άκυρων πακέτων που περνάνε από τον Resolver δεν είναι πολύ υψηλός.



Σχήμα 4.9 - Ποσοστά έγκυρων/άκυρων ερωτημάτων σε σχέση με τα συνολικά ερωτήματα τα οποία επεξεργάστηκε ο Authoritative Server (Πείραμα 2).

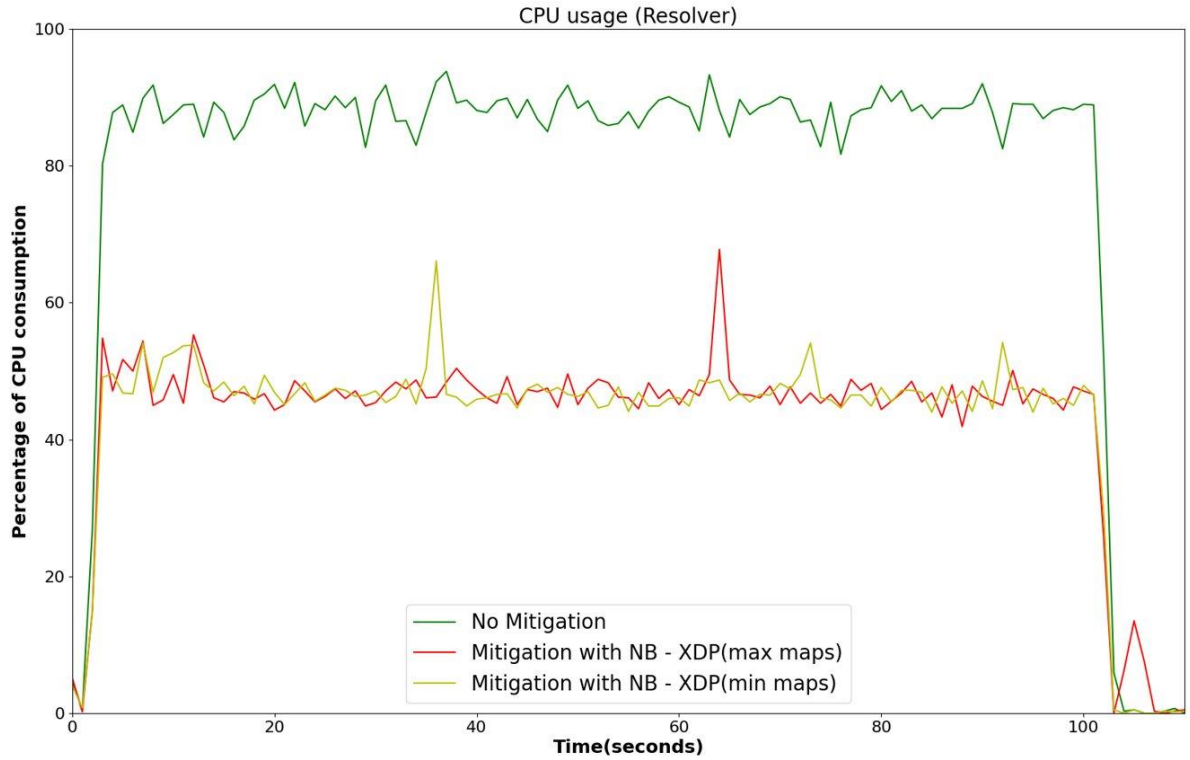
4)

Παρατηρείται μείωση παρόμοια με το πρώτο πείραμα, στην κατανάλωση του CPU του Authoritative. Επομένως, παρόλο που ο Authoritative χρησιμοποιεί τους πόρους του για την επεξεργασία περισσότερων άκυρων ερωτημάτων από ότι στο πρώτο πείραμα, δεν προκαλείται πρόβλημα στην λειτουργία του και δεν κατασπαταλούνται οι πόροι του, αφού το μεγαλύτερο μέρος της κακόβουλης κίνησης απορρίπτεται από τον αναδρομικό εξυπηρετητή.



Σχήμα 4.10 - Ποσοστό χρήσης επεξεργαστή του Authoritative, για τις 3 καταστάσεις λειτουργίας του Resolver (Πείραμα 2).

Αντίστοιχα αποτελέσματα προκύπτουν και για την χρήση του επεξεργαστή του αναδρομικού εξυπηρετητή.



Σχήμα 4.11 - Ποσοστό χρήσης επεξεργαστή του Resolver, για τις 3 καταστάσεις λειτουργίας του (Πείραμα 2).

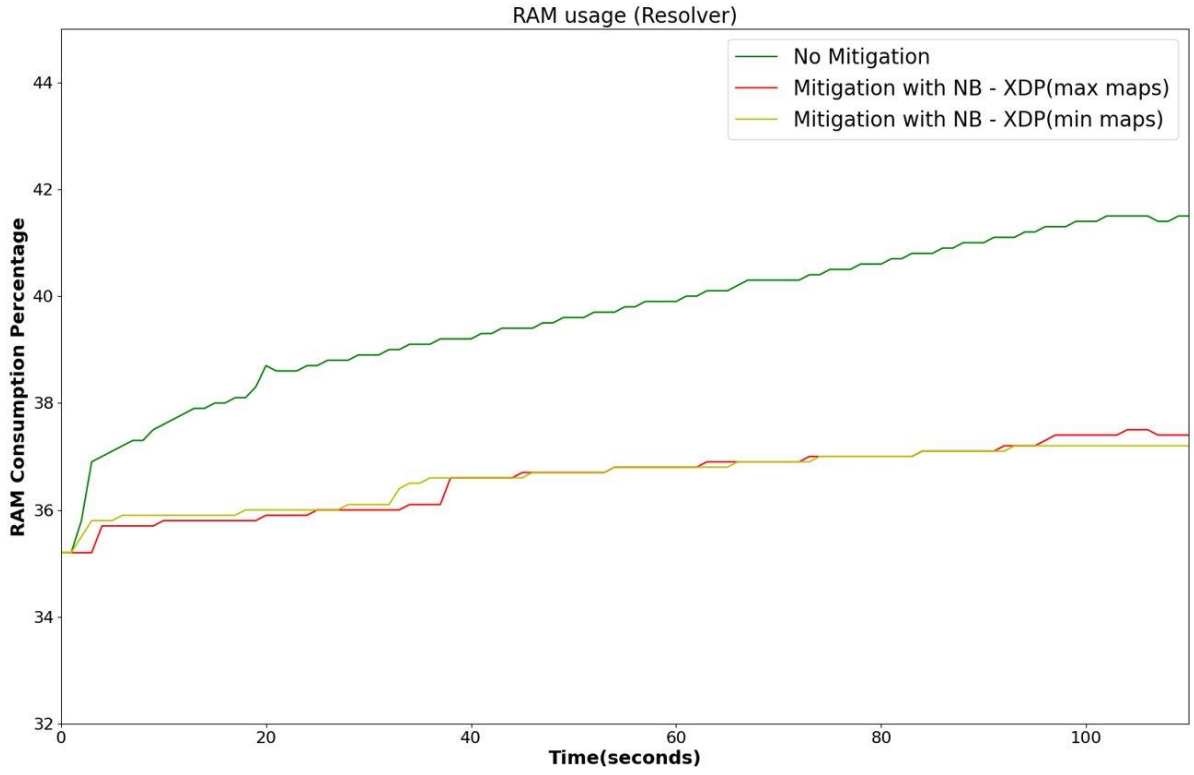
5)

Το τελευταίο διάγραμμα αφορά την κατανάλωση φυσικής μνήμης του Resolver.

Όπως είχε αναφερθεί σε προηγούμενα κεφάλαια, ο αναδρομικός εξυπηρετητής χρησιμοποιεί μία προσωρινή μνήμη (cache) για να αποθηκεύσει τις απαντήσεις σε ερωτήματα που του γίνονται, έτσι ώστε να απαντάει απευθείας, χωρίς να εκτελεί την γνωστή διαδικασία επίλυσης ονόματος. Αυτή η προσωρινή μνήμη βρίσκεται στην RAM του υπολογιστή.

Σε μία επίθεση DNS Water Torture, τα ονόματα που χρησιμοποιούνται είναι όλα διαφορετικά μεταξύ τους, έτσι ώστε ο αναδρομικός εξυπηρετητής να στέλνει πάντα τα ερωτήματα στον Authoritative Server που αποτελεί θύμα της επίθεσης. Αυτό όμως έχει σαν αποτέλεσμα η προσωρινή μνήμη του Resolver να γεμίζει την προσωρινή του μνήμη με τις απαντήσεις NXDomain που λαμβάνει από τον Authoritative και κατά συνέπεια να καταναλώνει περισσότερη φυσική μνήμη.

Στο διάγραμμα φαίνεται πως στην περίπτωση απουσίας αμυντικού μηχανισμού το ποσοστό χρήσης της RAM αυξάνεται κατά περίπου 6% ενώ στις άλλες δύο περιπτώσεις υπάρχει μια μικρή αύξηση κατά περίπου 1.5%. Αυτό δείχνει ένα ακόμη πλεονέκτημα του μηχανισμού, καθώς σε μια πραγματική επίθεση η φυσική μνήμη θα παρουσίαζε πολύ μεγάλη αύξηση που ενδεχομένως να δημιουργούσε προβλήματα στην σωστή λειτουργία του αναδρομικού εξυπηρετητή. Συνεπώς, η χρήση του αμυντικού μηχανισμού βοηθάει και τον Resolver στο να μην κατασπαταλάει την RAM του στην αποθήκευση πολλών άκυρων ερωτημάτων.



Σχήμα 4.12 - Ποσοστό χρήσης φυσικής μνήμης του Resolver και στις 3 καταστάσεις λειτουργίας του (Πείραμα 1).

5. Επίλογος

Στο κεφάλαιο αυτό παρατίθενται τα συμπεράσματα της εργασίας και παρουσιάζονται προτάσεις για την επέκταση του μηχανισμού που αφήνονται ως μελλοντική εργασία.

5.1. Συμπεράσματα

Στα πλαίσια αυτής της διπλωματικής εργασίας αναπτύχθηκε και αξιολογήθηκε ένας μηχανισμός αντιμετώπισης, στο επίπεδο του αναδρομικού εξυπηρετητή, της επίθεσης DNS Water Torture. Ο μηχανισμός χρησιμοποιεί επιβλεπόμενη μάθηση και συγκεκριμένα τον ταξινομητή Naive Bayes για να κρίνει εάν ένα όνομα, ενός ερωτήματος DNS που δέχεται, είναι έγκυρο ή άκυρο, δηλαδή αν υπάρχει ή όχι στο αρχείο ονομάτων του Authoritative Server. Στην πρώτη περίπτωση το ερώτημα προωθείται κανονικά στον Authoritative Server ενώ στην δεύτερη απορρίπτεται από τον αναδρομικό εξυπηρετητή. Ο ταξινομητής υλοποιήθηκε με την χρήση eBPF και ο μηχανισμός προσαρτήθηκε στο XDP hook για επίτευξη υψηλού ρυθμού επεξεργασίας πακέτων.

Σκοπός ήταν η μελέτη της απόδοσης του ταξινομητή, υλοποιημένου σε XDP, για την αποτελεσματική απόρριψη κακόβουλων πακέτων και την προώθηση όσο το δυνατό περισσότερων έγκυρων ερωτημάτων και όχι η ακρίβεια στις ταξινομήσεις του Naive Bayes, καθώς σύμφωνα με μελέτη [35] φαίνεται πως μπορεί να επιτευχθεί υψηλό ποσοστό ακρίβειας προβλέψεων. Σύμφωνα με τα αποτελέσματα ο μηχανισμός αποτελεί μία πολύ καλή λύση απέναντι σε μία τέτοια επίθεση καθώς επιτυγχάνει την προώθηση του μεγαλύτερου ποσοστού των έγκυρων ερωτημάτων και ταυτόχρονα την απόρριψη ενός μεγάλου τμήματος της κακόβουλης κίνησης. Παράλληλα, ο επεξεργαστικός φόρτος των εξυπηρετητών (Resolver και Authoritative) και η φυσική μνήμη που καταναλώνεται, λόγω caching των ονομάτων, κατά την διάρκεια της επίθεσης παρουσιάζουν σημαντική μείωση με τη χρήση του μηχανισμού.

5.2. Μελλοντικές επεκτάσεις

Ο μηχανισμός υλοποιήθηκε με eBPF και προσαρτήθηκε στο XDP hook. Θα μπορούσε να υλοποιηθεί σε P4 [66] ή μέσω του Data Plane Development Kit (DPDK) [61] και να συγκριθούν οι αποδόσεις.

Επίσης, ενδιαφέρουσα επέκταση αποτελεί η υλοποίηση άλλων αλγορίθμων μηχανικής μάθησης με σκοπό αντιμετώπιση της επίθεσης Water Torture ή και άλλων δικτυακών επιθέσεων.

Τέλος, όπως είχε αναφερθεί και στο κεφάλαιο 2, το XDP έχει και άλλους τρόπους λειτουργίας. Στην περίπτωση του Generic ο πυρήνας προσομοιώνει την λειτουργία του XDP και αυτό έχει ένα κόστος στην απόδοση. Στο Native το XDP υλοποιείται από τους drivers της κάρτας δικτύου με αποτέλεσμα την επίτευξη υψηλότερων επιδόσεων. Σαν μελλοντική επέκταση λοιπόν, αφήνεται η εκτέλεση του μηχανισμού με τη χρήση Native XDP και η μελέτη της βελτιωμένης απόδοσής του.

Βιβλιογραφία

[1] Domain Name System, Wikipedia,
https://en.wikipedia.org/wiki/Domain_Name_System

[2] DNS Hierarchy, novel.com,
https://www.novell.com/documentation/dns_dhcp/?page=/documentation/dns_dhcp/dhcp_enu/data/behdbhhj.html

[3] Paul Albitz and Cricket Liu,"DNS and BIND",4th Edition O 'Reilly,
https://docstore.mik.ua/orelly/networking_2ndEd/dns/index.htm

[4] Baojun Liu, Chaoyi Lu, Haixin Duan, Ying Liu, Zhou Li, Shuang Hao, Min Yang , "Who Is Answering My Queries: Understanding and CharacterizingInterception of the DNS Resolution Path ", Usenix Security,
https://www.researchgate.net/publication/330006223_Who_Is_Answering_My_Queries_Understanding_and_Characterizing_Interception_of_the_DNS_Resolution_Path

[5] DOMAIN NAMES - CONCEPTS AND FACILITIES, RFC,
https://www.ietf.org/rfc/rfc1034.txt

[6] DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION, RFC,
https://www.ietf.org/rfc/rfc1035.txt

[7] Lab 4: DSN Primer Notes, cs.duke,
https://courses.cs.duke.edu//fall16/compsci356/DNS/DNS-primer.pdf

[8] Introduction to DNS Privacy, internetsociety.org,
https://www.internetsociety.org/resources/deploy360/dns-privacy/intro/

[9] Suranjith Ariyapperuma, Chris J. Mitchell, "Security vulnerabilities in DNS and DNSSEC", https://ieeexplore.ieee.org/document/4159821

[10] DNS Security, Cloudflare, https://www.cloudflare.com/learning/dns/dns-security/

[11] Giuseppe Ateniese, Stefan Mangard, "A new approach to DNS security (DNSSEC)", https://dl.acm.org/doi/pdf/10.1145/501983.501996

[12] DNS over TLS, Cloudflare, https://www.cloudflare.com/learning/dns/dns-over-tls/

[13] Paul Schmitt, Anne Edmundson, Allison Mankin, and Nick Feamster , "Oblivious DNS: Practical Privacy for DNS Queries", https://arxiv.org/pdf/1806.00276.pdf

[14] Denial of Service, Cloudflare,
https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/

[15] Botnet, Cloudflare, https://www.cloudflare.com/learning/ddos/what-is-a-ddos-botnet/

[16] Botnet, Wikipedia, https://en.wikipedia.org/wiki/Botnet

[17] IP Spoofing, Cloudflare, https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/

[18] The Psychology Behind DDoS Attacks, https://www.perimeter81.com/blog/network/the-psychology-behind-ddos-attacks

[19] https://sites.google.com/site/eisagogestadiktyaypologiston1/architektonike-diktyou/montelo-anaphoras-osi

[20] OSI model, Wikipedia, https://en.wikipedia.org/wiki/OSI_model

[21] DDoS attack, Cloudflare, https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

[22] Xi Luo, Liming Wang, Zhen Xu, Kai Chen, Jing Yang, Tian Tian , “A Large Scale Analysis of DNS Water Torture Attack”, https://dl.acm.org/doi/pdf/10.1145/3297156.3297272

[23] Marios Anagnostopoulos, Georgios Kambourakis, Panagiotis Kopanos, Georgios Louloudakis, Stefanos Gritzalis, “DNS amplification attack revisited”, https://www.sciencedirect.com/science/article/pii/S0167404813001405?via%3Dihub

[24] “What is DNS cache poisoning?” , https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

[25] Yehuda Afek, Anat Bremler-Barr, Lior Shafir, “NXNSAttack: Recursive DNS Inefficiencies and Vulnerabilities”, https://cyber-security-group.cs.tau.ac.il/dns-ns-paper.pdf

[26] Marcos A. M. Vieira, Matheus S. Castanho, Racyus D. G. Pacífico, Elerson R. S. Santos, Eduardo P. M. Câmara Júnior, Luiz F. M. Vieira , “Fast Packet Processing with eBPF and XDP: Concepts, Code, Challenges, and Applications”, https://dl.acm.org/doi/pdf/10.1145/3371038

[27] Machine Learning, Wikipedia, https://en.wikipedia.org/wiki/Machine_learning

[28] I. Rish , “An empirical study of the naive Bayes classifier”, https://www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf

[29] Naive Bayes classifier, Wikipedia, https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[30] Stochastic Processes & Optimization in Machine Learning, Vasilis Maglaris, http://www.netmode.ntua.gr/courses/postgraduate/stochastic/2021/Stochastic_Processes_Optimization_Machine_Learning_VM_10_2021.pdf

[31] Facebook. 2018. Katran Source Code Repository. Retrieved from https://github.com/facebookincubator/katran

[32] David Beckett, Jaco Joubert, and Simon Horman. 2018. Host dataplane acceleration (HDA). In ACM SIGCOMM 2018 Tutorials (SIGCOMM'18). ACM, New York, NY.

[33] Gilberto Bertin. 2017. XDP in practice: Integrating XDP into our DDoS mitigation pipeline. In Proceedings of the Netdev 2.1 Technical Conference on Linux Networking. 1–5.

[34] BIND, https://www.isc.org/BIND/

[35] Takuro Yoshida, Kento Kawakami, Ryotaro Kobayashi, Masahiko Kato, Masayuki Okada, Hiroyuki Kishimoto , "Detection and Filtering System for DNS Water Torture Attacks Relying Only on Domain Name Information", https://www.jstage.jst.go.jp/article/ipsjjip/25/0/25_854/_pdf

[36] https://github.com/nkostopoulos/StochasticsLabPublic/blob/master/lab8/valid_training.txt

[37] https://github.com/nkostopoulos/StochasticsLabPublic/blob/master/lab8/invalid_training.txt

[38] BPF Compiler Collection, https://github.com/iovisor/bcc

[39] VirtualBox, https://www.virtualbox.org/

[40] Kali Linux, https://www.kali.org/get-kali/

[41] Tcpreplay, https://tcpreplay.appneta.com/

[42] Ubuntu, https://ubuntu.com/

[43] https://github.com/baderj/Domain_generation_algorithms

[44] https://www.domcop.com/top-10-million-Domains

[45] DNS statistics, nic, https://www.nic.ch/statistics/dns/

[46] Tcpdump, https://www.tcpdump.org/

[47] Wireshark, https://www.wireshark.org/

[48] Local DNS Attack Lab, seedsecuritylabs.org, https://web.ecs.syr.edu/~wedu/seed/Labs_12.04/Networking/DNS_Local/DNS_Local.pdf

[49] How to capture and replay network traffic on Linux, xmodulo, https://www.xmodulo.com/how-to-capture-and-replay-network-traffic-on-linux.html

[50] "p4c-ubpf: a New Back-end for the P4 Compiler" , https://opennetworking.org/news-and-events/blog/p4c-ubpf-a-new-back-end-for-the-p4-compiler/

[51] https://github.com/p4lang/p4c/tree/main/backends/ebpf

[52] “BPF and XDP Reference Guide” , https://docs.cilium.io/en/stable/bpf/

[53] “ODNS: Oblivious DNS”, https://odns.cs.princeton.edu/

[54] “Man in the middle (MITM) attack”, https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/

[55] Peer to peer, Wikipedia, https://el.wikipedia.org/wiki/Peer-to-peer

[56] “NXNS Attack Vulnerability in the DNS”, https://www.denic-services.de/fileadmin/anycast/nxns/Anycast_NXNS_Attack_pdf.pdf

[57] eBPF, https://ebpf.io/

[58] Toke Høiland-Jørgensen, Jesper Dangaard Brouer, Daniel Borkmann, John Fastabend, Tom Herbert, David Ahern, David Miller, “The eXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel”, https://dl.acm.org/doi/pdf/10.1145/3281411.3281443

[59] Steven McCanne, Van Jacobson, “The BSD Packet Filter: A New Architecture for User-level Packet Capture”, https://www.tcpdump.org/papers/bpf-usenix93.pdf

[60] libbpf, https://github.com/libbpf/libbpf

[61] DPDK, https://www.dpdk.org/]

[62] BPF and XDP Reference Guide, https://docs.cilium.io/en/stable/bpf/#xdp

[63] XDP benchmark baseline, https://github.com/tohojo/xdp-paper/blob/master/benchmarks/bench01_baseline.org#initial-data-from-jespers-runs

[64] Alexa Top 1 Million Sites, https://www.kaggle.com/cheedcheed/top1m

[65] generator.py, https://github.com/nkostopoulos/StochasticsLabPublic/blob/master/lab8/generator.py

[66] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, David Walker, “P4: programming protocol-independent packet processors”, https://dl.acm.org/doi/pdf/10.1145/2656877.2656890

[67] Jonathan Trostle, Bill Van Besien, Ashish Pujari , “Protecting against DNS cache poisoning attacks”, https://ieeexplore.ieee.org/document/5634454

[68] iovisor, eBPF use cases, https://www.iovisor.org/technology/use-cases

[69] clang - llvm, https://clang.llvm.org/

[70] bpf helper functions, https://man7.org/linux/man-pages/man7/bpf-helpers.7.html#HELPERS

[71] Brendan Gregg's Blog, “Linux MySQL Slow Query Tracing with bcc/BPF”, https://www.brendangregg.com/blog/2016-10-04/linux-bcc-mysqld-qslower.html

[72] openstack, “In-kernel Analytics and Tracing with eBPF for OpenStack Clouds”, https://www.openstack.org/videos/summits/barcelona-2016/in-kernel-analytics-and-tracing-with-ebpf-for-openstack-clouds

[73] https://www.brendangregg.com/blog/2021-07-03/how-to-add-bpf-observability.html

[74] https://www.slideshare.net/IOVisor/using-io-visor-to-secure-microservices-running-on-cloudfoundry-openstack-summit-austin-april-2016?qid=6ebb7aa3-30d8-4468-9b67-fc5826ed25c3&v=&b=&from_search=4

[75] https://blog.cloudflare.com/l4drop-xdp-ebpf-based-ddos-mitigations/

[76] https://cilium.io/blog/2021/05/20/cilium-110#standalonelb

[77] https://tcpreplay.appneta.com/wiki/tcprewrite-man.html

Παράρτημα

Α. Κώδικας

Ο κώδικας του αμυντικού μηχανισμού που υλοποιήθηκε σε αυτή τη διπλωματική εργασία είναι διαθέσιμος στο github repository **https://github.com/skorentis/dns_water_torture_xdp_mitigation**. Σε αυτόν τον σύνδεσμο βρίσκονται τα datasets για το training/testing του Naive Bayes καθώς και τα valid/invalid ονόματα που χρησιμοποιήθηκαν για την δημιουργία του valid/invalid traffic.

Β. Πηγές σχημάτων και πινάκων

Παρακάτω καταγράφονται οι πηγές των σχημάτων και των πινάκων που χρησιμοποιήθηκαν στην εργασία.

Β.1. Πηγές σχημάτων

- **Σχήμα 2.1** : https://commons.wikimedia.org/wiki/File:Root-historic.svg
- **Σχήμα 2.2** : https://blog.linkody.com/what-is-a-Root-Domain
- **Σχήμα 2.3** : https://en.wikipedia.org/wiki/Fully_qualified_Domain_name#/media/File:DNS_schema.svg
- **Σχήμα 2.4** : https://docstore.mik.ua/orelly/networking_2ndEd/dns/ch02_04.htm
- **Σχήμα 2.5** : Baojun Liu, Chaoyi Lu, Haixin Duan, Ying Liu, Zhou Li, Shuang Hao, Min Yang , “Who Is Answering My Queries: Understanding and CharacterizingInterception of the DNS Resolution Path “, Usenix Security, https://www.researchgate.net/figure/Domain-resolution-process-with-a-recursive-resolver_fig1_330006223
- **Σχήματα 2.6 - 2.9** : https://courses.cs.duke.edu//fall16/compsci356/DNS/DNS-primer.pdf
- **Σχήμα 2.10** : https://www.internetsociety.org/resources/deploy360/dns-privacy/intro/.
- **Σχήμα 2.11** : https://www.cloudflare.com/learning/dns/dns-over-tls/
- **Σχήμα 2.12** : Paul Schmitt, Anne Edmundson, Allison Mankin, and Nick Feamster , “Oblivious DNS: Practical Privacy for DNS Queries”, https://arxiv.org/pdf/1806.00276.pdf
- **Σχήμα 2.13** : https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/
- **Σχήματα 2.14 - 2.17** : https://www.cloudflare.com/learning/ddos/what-is-a-ddos-botnet
- **Σχήμα 2.18** : Riaz Ullah Khan, Xiaosong Zhang, Rajesh Kumar, Abubakar Sharif, Noorbakhsh Amiri Golilarz and Mamoun Alazab , “An Adaptive Multi-Layer Botnet Detection TechniqueUsing Machine Learning Classifiers”, https://www.researchgate.net/figure/Difference-between-centralized-and-decentralized-P2P-botnets_fig1_333706685
- **Σχήμα 2.19** : https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/
- **Σχήμα 2.20** : https://sites.google.com/site/eisagogestadiktyaypologiston1/architektonike-diktyou/montelo-anaphoras-osi
- **Σχήματα 2.21 - 2.22** : https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

- **Σχήμα 2.24** : https://ars.els-cdn.com/content/image/1-s2.0-S0167404813001405-gr1.jpg
- **Σχήμα 2.25** : https://www.cloudflare.com/learning/dns/dns-cache-poisoning/
- **Σχήμα 2.26** : Yehuda Afek, Anat Bremler-Barr, Lior Shafir, “NXNSAttack: Recursive DNS Inefficiencies and Vulnerabilities”, https://cyber-security-group.cs.tau.ac.il/dns-ns-paper.pdf
- **Σχήμα 2.27** : https://www.hackreports.com/content/images/2020/05/NXNS-Attack-.png
- **Σχήματα 2.28 - 2.29** : Marcos A. M. Vieira, Matheus S. Castanho, Racyus D. G. Pacífico, Elerson R. S. Santos, Eduardo P. M. Câmara Júnior, Luiz F. M. Vieira , “Fast Packet Processing with eBPF and XDP: Concepts, Code, Challenges, and Applications”, https://dl.acm.org/doi/pdf/10.1145/3371038
- **Σχήμα 2.30** : BPF and XDP Reference Guide, https://docs.cilium.io/en/stable/bpf/#maps
- **Σχήμα 2.31** : iovisor, bcc, https://raw.githubusercontent.com/iovisor/bcc/master/images/bcc_tracing_tools_2019.png
- **Σχήματα 2.32 - 2.33** : Marcos A. M. Vieira, Matheus S. Castanho, Racyus D. G. Pacífico, Elerson R. S. Santos, Eduardo P. M. Câmara Júnior, Luiz F. M. Vieira , “Fast Packet Processing with eBPF and XDP: Concepts, Code, Challenges, and Applications”, https://dl.acm.org/doi/pdf/10.1145/3371038

Β.2. Πηγές πινάκων

- **Πίνακας 2.1** : DNS Hierarchy, novel.com, https://www.novell.com/documentation/dns_dhcp/?page=/documentation/dns_dhcp/dhcp_enu/data/behdbhhj.html
- **Πίνακες 2.2 - 2.3** : Marcos A. M. Vieira, Matheus S. Castanho, Racyus D. G. Pacífico, Elerson R. S. Santos, Eduardo P. M. Câmara Júnior, Luiz F. M. Vieira , “Fast Packet Processing with eBPF and XDP: Concepts, Code, Challenges, and Applications”, https://dl.acm.org/doi/pdf/10.1145/3371038