Πανεπιστήμιο Δυτικής Αττικής

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΗΧΑΝΙΣΜΟΙ ΕΛΕΓΧΟΥ ΠΡΟΣΒΑΣΗΣ ΜΕ ΧΡΗΣΗ ΣΥΝΘΗΜΑΤΙΚΩΝ

Ιωαννίδου Βασιλική

71347682

Επιβλέπουσα Καθηγήτρια

Καντζάβελου Ιωάννα, Επίκουρη Καθηγήτρια

Αθήνα

Σεπτέμβριος, 2022

Η Τριμελής Εξεταστική Επιτροπή

Η Επιβλέπουσα Καθηγήτρια, Καντζάβελου Ιωάννα

Καντζάβελου Ιωάννα	Μάμαλης Βασίλειος	Καρκαζής Παναγιώτης
Επίκουρη Καθηγήτρια	Καθηγητής	Αναπληρωτής Καθηγητής
Ioanna Kantzavelou Digitally signed by Ioanna Kantzavelou Date: 2022.10.26 11:30:32 +03'00'	Vasileios Mamalis Digitally signed by Vasileios Mamalis Date: 2022.10.26 20:15:35 +03'00'	Panagiotis Karkazis Digitally signed by Panagiotis Karkazis Date: 2022.10.26 12:31:24 +03'00'
[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]	[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]	[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

~~Ο~~/η υπογράφ~~ων~~/ούσα Ιωαννίδου Βασιλική του Δημητρίου με αριθμό μητρώου 71347682 φοιτητ~~ής~~/τρια του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου».

Ημερομηνία

~~Ο~~/Η Δηλ~~ών~~/ούσα

25/10/2022

Περίληψη

Με την επέκταση της διαδικτυακής μετα-πραγματικότητας οι χρήστες είναι πλέον εξοικειωμένοι με την χρήση εφαρμογών και υπηρεσιών που παρέχονται μέσω του διαδικτύου. Η παρουσία των χρηστών εντός του διαδικτύου και η διατήρηση σημαντικών και ευαίσθητων δεδομένων πάνω σε αυτό δημιουργεί ζητήματα ασφαλείας που πρέπει να αντιμετωπίζονται με σοβαρότητα και προσοχή. Στην παρούσα διπλωματική εργασία αποσκοπούμε σε δύο βασικούς στόχους. Πρώτος στόχος είναι να εισάγουμε τον αναγνώστη στην παρούσα κατάσταση του διαδικτύου, τις μεθόδους ασφαλείας και προστασίας των χρηστών και τους κινδύνους που καραδοκούν εντός του κυβερνοχώρου. Στην συνέχεια ως δεύτερο στόχο έχουμε να αναλύσουμε τις ευπάθειες των παρόντων μεθόδων επαλήθευσης χρήστη και κατ' επέκταση, να προτείνουμε μια δική μας λύση που εν δυνάμει να βελτιώνει το πρόβλημα. Με το πέρας της μελέτης μας θα παρουσιάσουμε τα συλλογικά συμπεράσματα και τους σχολιασμούς μας.

Λέξεις Κλειδιά: Συνθηματικά Πρόσβασης, Ασφάλεια Συστημάτων, Κατανεμημένα Συστήματα, Διαδικτυακές Επιθέσεις.

Abstract

With the expansion of online meta-reality users are now familiar with the use of applications and services provided over the internet. The presence of users on the Internet and the maintenance of important and sensitive data on it creates security issues that must be treated with seriousness and caution. In this thesis we aim at two main goals. The first goal is to introduce the reader to the current state of the internet, security and user protection methods, and the dangers lurking within cyberspace. Then, as a second goal, we intent to analyze the vulnerabilities of the current user verification methods and, by extension, to propose our own solution that could potentially improve the problem. At the end of our study, we will present our collective conclusions and comments.

Keywords: Passwords, System Security, Distributed Systems, Internet Attacks

Ευχαριστίες:

Ευχαριστώ πολύ όσους με στήριξαν σε αυτή μου την προσπάθεια, έτσι ώστε να πετύχω το καλύτερο δυνατό αποτέλεσμα. Ευχαριστώ τους γονείς μου, τους φίλους μου και κυρίως την Επιβλέπουσα Καθηγήτριά μου Ιωάννα Καντζάβελου που με στήριξε και με υποστήριξε καθ' όλη την διάρκεια αυτής μου της προσπάθειας.

Περιεχόμενα

Πίνακας Εικόνων, Γραφημάτων και Πινάκων

Κεφάλαιο 1: Η Πρόσβαση στα Υπολογιστικά Συστήματα

1.1. Ο Έλεγχος Πρόσβασης

Ο **έλεγχος πρόσβασης (Access Control)**, στον κλάδο της ασφάλειας υπολογιστών, θεωρείται βασικό στοιχείο, καθώς λειτουργεί και ως **υπηρεσία (security service)**, αλλά και ως **μηχανισμός (security mechanism)**. Ένα σύστημα πρέπει να προστατεύει τους πόρους του από μη εξουσιοδοτημένους χρήστες, ενώ οι εξουσιοδοτημένοι χρήστες δεν πρέπει να προσπελάζουν τους πόρους του συστήματος με μη εξουσιοδοτημένους τρόπους και αυτός είναι ακριβώς ο ρόλος τού ελέγχου πρόσβασης· να αποφασίζει-καθορίζει που και σε ποιους πρέπει να παρέχεται πρόσβαση στα αντικείμενα ενός συστήματος.

Σαν υπηρεσία ασφάλειας, εφαρμόζει μια **πολιτική (policy)** που καθορίζει ποιος (π.χ ανθρώπινο δυναμικό) ή τι (π.χ μια διεργασία) μπορεί να προσπελάσει και να έχει πρόσβαση σε κάποιους από τους πόρους του συστήματος. Ως μηχανισμός υλοποιεί αυτή την πολιτική, έτσι ώστε να παρέχει εξουσιοδότηση μόνο στις οντότητες που πρέπει.

Ο έλεγχος πρόσβασης δεν περιορίζεται μόνο στην πρόσβαση σε ένα σύστημα ή ένα αρχείο ή μια βάση δεδομένων. Αντιθέτως, μπορεί να αφορά:

- Την φυσική πρόσβαση σε εγκαταστάσεις που στεγάζουν ευαίσθητες πληροφορίες ή εξοπλισμό.
- Την πρόσβαση στο δίκτυο.
- Την πρόσβαση στον κεντρικό υπολογιστή (host) για την λειτουργικότητα του συστήματος.

Επιπλέον, πόροι δεν θεωρούνται μόνο τα αρχεία, αλλά και οι εφαρμογές, η μνήμη, τα μέσα αποθήκευσης/μετάδοσης και γενικά ό,τι μπορεί να επεξεργαστεί, να αποθηκευτεί ή να μεταδοθεί [3]. Γενικά στον έλεγχο πρόσβασης έχουμε κάποια βασικά στοιχεία τα οποία αναλύονται παρακάτω [1]:

- **Υποκείμενο (Subject)**: Οποιαδήποτε οντότητα, όπως χρήστης, διαχειριστής, διαδικασία, πρόγραμμα κ.α. χρησιμοποιεί ένα αντικείμενο.

- **Αντικείμενο (Object)**: Ο πόρος που χρησιμοποιείται από τον υποκείμενο. Ένας πόρος περιέχει πληροφορίες, όπως μια εγγραφή, ένα αρχείο, ένας φάκελος κ.α.
- **Δικαιώματα Πρόσβασης (Access Right)**: Ο τρόπος με τον οποίο ένα υποκείμενο μπορεί να προσπελάσει ένα αντικείμενο. Οι τρόποι αυτοί είναι ανάγνωση, εγγραφή, εκτέλεση, δημιουργία και διαγραφή.

1.2. Πολιτικές Ελέγχου Πρόσβασης

Οι **πολιτικές ελέγχου πρόσβασης (access control policies)** εφαρμόζονται σε ομάδες πόρων και μπορούν να διαφέρουν ανάλογα με τις δυνατότητες του κάθε πόρου. Για παράδειγμα οι κοινές δυνατότητες που μπορεί να έχει μια οντότητα πάνω σε ένα αρχείο είναι: **Ανάγνωση (Read), Εγγραφή (Record), Εκτέλεση (Execute), Δημιουργία (Create)** και **Διαγραφή (Delete)**. Οι πολιτικές αυτές διακρίνονται σε 4 κατηγορίες και παρουσιάζονται συνοπτικά παρακάτω [3]:

- **Διακριτικός έλεγχος πρόσβασης (discretionary access control, DAC)**: Βασίζεται στην ταυτότητα και στις εξουσιοδοτήσεις του χρήστη που θέλει να αποκτήσει πρόσβαση. Είναι διακριτικός κατά την έννοια ότι, μπορεί να επιτρέψει και σε μια άλλη οντότητα να προσπελάσει κάποιον πόρο.
- **Υποχρεωτικός έλεγχος πρόσβασης (mandatory access control, MAC)**: Βασίζεται στην ευαισθησία των πληροφοριών που περιέχονται στους πόρους. Είναι υποχρεωτική κατά την έννοια ότι, δεν μπορεί να επιτρέψει σε μια άλλη οντότητα να προσπελάσει κάποιον πόρο.
- **Έλεγχος πρόσβασης βασισμένος σε ρόλους (role-based access control, RBAC)**: Βασίζεται στους ρόλους (προνόμια) που έχουν οι χρήστες του συστήματος, καθώς και στους κανόνες που ορίζουν τον τύπο την επιτρεπόμενης πρόσβασης για χρήστες με δεδομένους ρόλους.
- **Έλεγχος πρόσβασης βασισμένος σε ιδιότητες (attribure-based access control, ABAC)**: Βασίζεται στις ιδιότητες που έχουν οι χρήστες του συστήματος, τον πόρο που θα προσπελαστεί ή την απαιτούμενη ενέργεια.

1.3. Αρχή Ελάχιστων Προνομίων (Least Privileges)

Στο επίπεδο ασφάλειας του ελέγχου πρόσβασης, υπάρχει μια αρχή που ονομάζεται **αρχή ελάχιστων προνομίων (least privileges)** και στοχεύει στο γεγονός ότι ένας μηχανισμός-αρχιτεκτονική ασφάλειας, όπως για παράδειγμα μια **λίστα ελέγχου πρόσβασης** (**access control list, ACL** η οποία θα αναλυθεί παρακάτω μαζί με άλλους μηχανισμούς), πρέπει να σχεδιαστεί έτσι ώστε κάθε οντότητα του συστήματος να έχει μόνο τις απαραίτητες εξουσιοδοτήσεις και να τις παρέχονται μόνο οι απαραίτητοι πόροι που χρειάζονται για να εκτελέσει τα καθήκοντά της. Επιπρόσθετα, η αρχή αυτή έχει στόχο να περιορίσει οποιαδήποτε ζημιά μπορεί να προκληθεί από κάποιο σφάλμα ή μια μη εξουσιοδοτημένη οντότητα.

Οι οργανισμοί εφαρμόζουν τον κανόνα ελάχιστων προνομίων σε συστήματα πληροφοριών και στις διαδικασίες αυτών, διασφαλίζοντας έτσι ότι οι διαδικασίες λειτουργούν με τα προνομία που τους έχουν ανατεθεί από τον διαχειριστή του συστήματος και όχι με υψηλότερα. Επιπλέον, οι οργανισμοί θεωρούν ότι είναι απαραίτητη η δημιουργία πρόσθετων διαδικασιών, ρόλων και λογαριασμών συστήματος πληροφοριών, προκειμένου να επιτευχθεί το λιγότερο προνόμιο [3].

1.4. Μηχανισμοί Ελέγχου Πρόσβασης

Όπως αναφέρθηκε και παραπάνω, οι υπηρεσίες ασφάλειας εφαρμόζουν μια πολιτική που υλοποιείται από μηχανισμούς ασφαλείας· ένας μηχανισμός ασφαλείας είναι μια διαδικασία/μέθοδος, που υλοποιείται από το σύστημα με σκοπό να επιτευχθεί μια υπηρεσία ασφάλειας.

Οι μηχανισμοί ελέγχου πρόσβασης μπορεί να είναι σε επίπεδο υλικού, λογισμικού, φυσικού ελέγχου, διαδικασιών λειτουργίας ή και συνδυασμού αυτών. Παρακάτω αναφέρονται οι μηχανισμοί αυτοί:

- Μήτρα ελέγχου πρόσβασης (access control matrix, ACM)
- Λίστα ελέγχου πρόσβασης (access control list, ACL)
- Λίστα δυνατοτήτων (Capability list)
- Κωδικοί Πρόσβασης (Passwords)

1.4.1. Μήτρα Ελέγχου Πρόσβασης (ACM)

Μια **μήτρα ελέγχου πρόσβασης** είναι ένας πίνακας όπου οι στήλες αντιπροσωπεύουν τα αντικείμενα, οι γραμμές τα υποκείμενα και κάθε κελί υποδηλώνει τα δικαιώματα πρόσβασης που έχει το υποκείμενο στο αντικείμενο. Επειδή, η μήτρα είναι συνήθως αραιή, δηλαδή τα περισσότερα υποκείμενα δεν έχουν δικαιώματα πρόσβασης στα αντικείμενα οι στήλες αυτού του πίνακα υποδιαιρούνται σε ACLs ενώ οι γραμμές σε λίστες δυνατοτήτων [7].

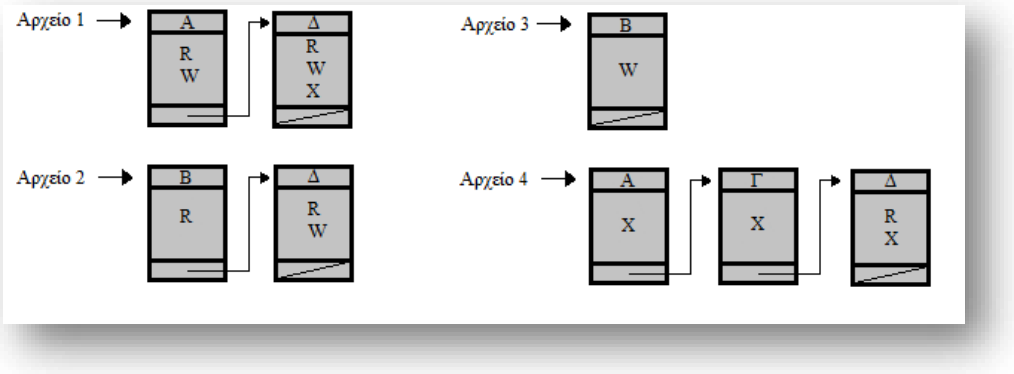
Υποκείμενα Subjects \ Αντικείμενα Objects	Αρχείο 1	Αρχείο 2	Αρχείο 3	Αρχείο 4
Χρήστης Α	RW			X
Χρήστης Β		R	W	
Χρήστης Γ				X
Χρήστης Δ	RWX	RW		RX

Εικόνα 1: Μήτρα Ελέγχου Πρόσβασης [1]

1.4.2. Λίστα Ελέγχους Πρόσβασης (ACL)

Μια **λίστα ελέγχου πρόσβασης** είναι μια λίστα δικαιωμάτων πρόσβασης που σχετίζεται με ένα αντικείμενο που καθορίζει όλα τα υποκείμενα που έχουν δικαιώματα πρόσβασης σε αυτό· μια στήλη από την ACM. Η λίστα μπορεί να περιλαμβάνει μια προεπιλεγμένη καταχώρηση, η οποία επιτρέπει σε χρήστες που δεν αναφέρεται ότι έχουν ειδικά δικαιώματα

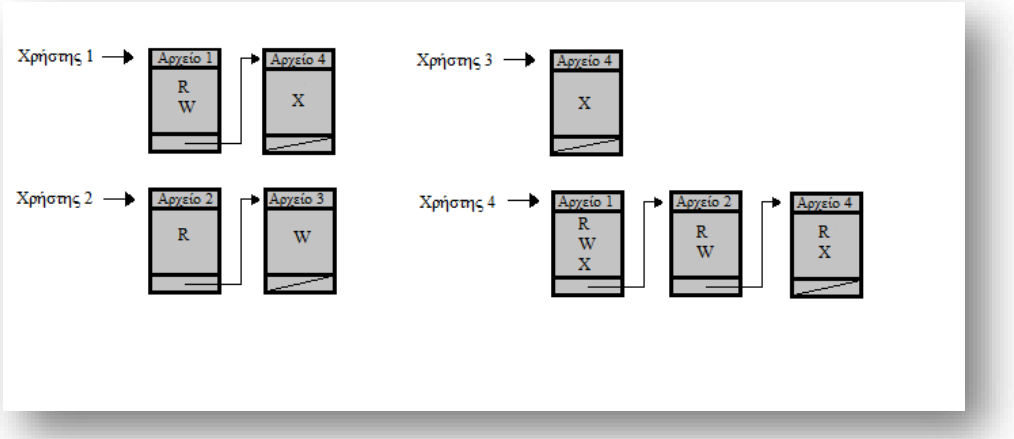
να έχουν ένα προεπιλεγμένο σύνολο δικαιωμάτων που ακολουθεί τον κανόνα ελάχιστων προνομίων [7].



Εικόνα 2: Λίστα Ελέγχου Πρόσβασης [1]

1.4.3. Λίστα Δυνατοτήτων (Capability List)

Μία **λίστα δυνατοτήτων** παρουσιάζει το υποκείμενο και καθορίζει μέσω ποιων δικαιωμάτων πρόσβασης μπορεί να προσπελαστεί το αντικείμενο· το ανάποδο από την λίστα ελέγχου πρόσβασης, είναι μια γραμμή του ACM. Μια δυνατότητα είναι ένα προστατευόμενο αποδεικτικό που αναγνωρίζει και καθορίζει τις λειτουργίες και τις ενέργειες που μπορεί να προβεί το υποκείμενο στον συγκεκριμένο αντικείμενο. [7]



Εικόνα 3: Λίστα Δυνατοτήτων [1]

1.4.4. Κωδικοί Πρόσβασης (Passwords)

Ο πιο διαδεδομένος μηχανισμός ελέγχου πρόσβασης, είναι οι **κωδικοί πρόσβασης**, τους οποίους και θα αναλύσουμε βαθιά στα επόμενα κεφάλαια. Οι κωδικοί πρόσβασης διαφέρουν από τους προηγούμενους μηχανισμούς ασφαλείας καθώς αυτοί δεν είναι ούτε λίστες ούτε μήτρες, αλλά κωδικοποιημένες συμβολοσειρές χαρακτήρων (character strings) και είναι γνωστές μόνο στο σύστημα και τον χρήστη. Συνήθως τις διαλέγει ο χρήστης, όμως υπάρχει και η επιλογή να εκχωρούνται από το σύστημα, καθώς υπάρχουν αλγόριθμοι που δημιουργούν τυχαίους κωδικούς πρόσβασης που δεν συσχετίζονται με τον χρήστη **(automated password generator)** [8].

Η διαδικασία ελέγχου πρόσβασης μέσω κωδικών πρόσβασης είναι πολύ απλή στον χρήστη. Ο χρήστης συνήθως είναι υποχρεωμένος να εισάγει ένα μοναδικό αναγνωριστικό **(username)** και έναν κωδικό πρόσβασης. Στην συνέχεια το σύστημα αναζητά στα αρχεία κωδικών πρόσβασης τον συγκεκριμένο κωδικό και αν ταυτοποιηθούν μεταξύ τους, δηλαδή ο κωδικός που εισήγαγε ο χρήστης υπάρχει στο αρχείο, τότε ο χρήστης αποκτά πρόσβαση στο σύστημα, αλλιώς το σύστημα αρνείται την πρόσβαση του.

Κατά την διαδικασία της ταυτοποίησης του χρήστη, είναι προτιμότερο το σύστημα να μην ενημερώνει τον χρήστη σε περίπτωση εσφαλμένου αναγνωριστικού ή κωδικού πρόσβασης, πιο από τα 2 πεδία είναι λανθασμένα. Με αυτό τον τρόπο παρέχουμε μια επιπλέον ασφάλεια στο σύστημα, εφόσον ο χρήστης που μπορεί να είναι μη εξουσιοδοτημένος (θα μπορούσε να είναι ένας επιτιθέμενος) δεν γνωρίζει που έχει γίνει το λάθος [2].

Υπάρχουν κάποιες πολιτικές που ενημερώνουν τον κόσμο του πως πρέπει να είναι ένα σωστός κωδικός πρόσβασης. Βάσει αυτών των πολιτικών, αλλά και βάσει της λογικής ένας κωδικός πρόσβασης πρέπει:

- Να είναι γνωστός μόνο στον χρήστη.
- Να είναι μεγάλου μήκους έτσι ώστε να είναι πιο δύσκολο να παραβιαστεί.

- Να περιέχει αριθμούς, κεφαλαία, πεζά, σύμβολα και ό,τι άλλο μπορεί να είναι αποδεκτό για να είναι επίσης πιο δύσκολος στην παραβίαση.
- Να αλλάζετε συχνά.
- Να μην καταγράφετε.
- Αποφυγή ημερομηνιών, ονόματος και άλλων αναγνωριστικών που συσχετίζονται άμεσα με τον χρήστη.

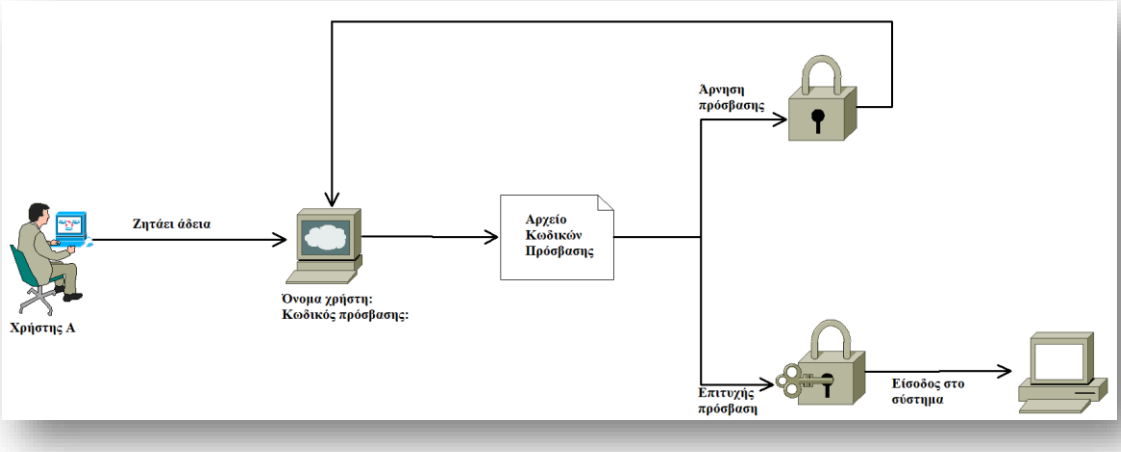
Κεφάλαιο 2: Κωδικοί Πρόσβασης

Σε αυτό το κεφάλαιο θα εξετάσουμε βαθιά τα συνθηματικά, τι είναι, που αποθηκεύονται ποια είναι η μορφή τους, ποια είναι η χρήση τους και άλλα πολλά που αφορούν τον συγκεκριμένο μηχανισμό ασφάλειας.

2.1. Δομή και λειτουργία

Ένας ευρέως διαδεδομένος μηχανισμός ελέγχου πρόσβασης είναι οι **κωδικοί πρόσβασης**, οι οποίοι έχουν μορφή κωδικοποιημένων συμβολοσειρών (character strings) και είναι γνωστές μόνο στο σύστημα και τον χρήστη. Πλέον, οι περισσότερες πλατφόρμες που βρίσκονται στο Διαδίκτυο (π.χ. κοινωνικά μέσα δικτύωσης), συστήματα με πολλούς χρήστες και άλλες τέτοιες υπηρεσίες ζητούν από τον χρήστη όχι μόνο ένα **αναγνωριστικό όνομα (username)**, αλλά και κωδικό πρόσβασης, για να εισέλθουν εκεί που επιθυμούν. Τα αναγνωριστικά χρησιμοποιούνται από την πολιτική **Διακριτικού Ελέγχου Πρόσβασης (DAC)** και σε συνδυασμό με τους κωδικούς πρόσβασης προστατεύουν ένα υπολογιστικό σύστημα από μη εξουσιοδοτημένες οντότητες που είτε αυτοί είναι **εισβολείς (hackers)** οι οποίοι προσπαθούν να εκμεταλλευτούν τυχόν ευπάθειες είτε απλοί χρήστες που δεν έχουν δικαιώματα πρόσβασης στα συγκεκριμένα αγαθά. [1]

Η ταυτοποίηση των κωδικών πρόσβασης γίνεται με ένα ήδη υπάρχον αρχείο στο σύστημα που φυλάει όλους τους κωδικούς πρόσβασης σε συνδυασμό με τα αναγνωριστικά ονόματος. Αν η ταυτοποίηση είναι επιτυχής τότε ο χρήστης αποκτά πρόσβαση, αλλιώς το σύστημα τον ενημερώνει ότι δεν μπορεί να εισέλθει σε αυτό.



Εικόνα 4: Οπτική Αναπαράσταση της ταυτοποίησης χρήστη και της εισόδου του στο σύστημα

Επιπρόσθετα, οι κωδικοί πρόσβασης χρησιμοποιούνται για την προστασία αρχείων και άλλων αποθηκευμένων πληροφοριών, όπως προστασία με κωδικό πρόσβασης για ένα μόνο συμπιεσμένο αρχείο, κρυπτογραφικό κλειδί ή κρυπτογραφημένο σκληρό δίσκο. Επιπλέον, χρησιμοποιούνται συχνά με λιγότερο ορατούς τρόπους. Για παράδειγμα, μια βιομετρική συσκευή μπορεί να δημιουργήσει έναν κωδικό πρόσβασης βάσει σάρωσης δακτυλικών αποτυπωμάτων και αυτός ο κωδικός χρησιμοποιείται στη συνέχεια για έλεγχο ταυτότητας.

Υπάρχουν διαφορετικές μορφές κωδικών πρόσβασης όπως είναι:

- Ο προσωπικός αριθμός ταυτοποίησης (Personal Identification Number, PIN)
- Η φράση πρόσβασης (passphrase)
- Η διαχείριση κωδικών πρόσβασης (Password Management).

Ο **προσωπικός αριθμός ταυτοποίησης (PIN)**, είναι μικρό σε μέγεθος (αποτελείται από 4 ή 6 χαρακτήρες) και αποτελείται μόνο από ψηφία. Συνήθως χρησιμοποιείται στις κινητές συσκευές, σε αυτόματες μηχανές ταμείου (ATM), σε συστήματα συναγερμού και άλλες συσκευές που έχουν μικρά πληκτρολόγια, για να αποφύγουν το εκτεταμένος μήκος χαρακτήρων και να είναι πιο εύκολο στον χρήστη να θυμηθεί το PIN. Θεωρείται ότι υπάρχει και φυσικός έλεγχος στις προηγούμενες περιπτώσεις που επιτρέπουν να χρησιμοποιείται μικρό μήκος χαρακτήρων. Τα PIN χρησιμοποιούνται σπάνια ως η μόνη μορφή ελέγχου. Για

παράδειγμα στις κινητές συσκευές εκτός από το PIN υπάρχουν και βιομετρικά όπως δαχτυλικό αποτύπωμα ή αναγνώριση προσώπου, κωδικός μοτίβο κ.α.

Η **φράση πρόσβασης (passphrase)**, είναι παρόμοια με τον κωδικό πρόσβασης και στην λειτουργία και στην σύνταξη, μόνο που είναι πιο μεγάλη σε μήκος και πιο συγκεκριμένα είναι μια σειρά λέξεων όπως μια φράση ή μια ολόκληρη πρόταση. Για παράδειγμα μια φράση πρόσβασης θα μπορούσε να είναι #Iminl0vEwithD0g$!. Ωστόσο, μια απλή πρόταση memyselfandi είναι πιο εύκολα προβλέψιμη, επειδή δεν υπάρχουν ειδική χαρακτήρες που θα την κάνουν "άσπαστη". Συνεπώς, το μεγάλο μήκος δεν είναι από μόνο του ισχυρό.

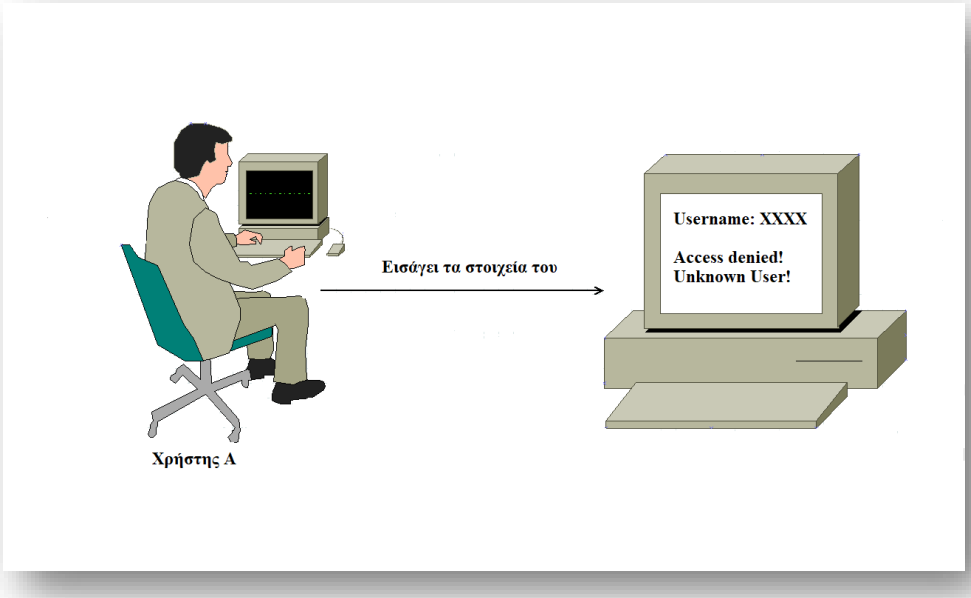
Η **διαχείριση κωδικών πρόσβασης (Password management)** είναι η διαδικασία καθορισμού, εφαρμογής και διατήρησης πολιτικών κωδικού πρόσβασης σε μια επιχείρηση. Η αποτελεσματική διαχείριση κωδικών πρόσβασης μειώνει τον κίνδυνο παραβίασης συστημάτων ελέγχου ταυτότητας βάσει κωδικού πρόσβασης στο βαθμό που είναι δυνατόν. Οι οργανισμοί πρέπει να προστατεύσουν την **εμπιστευτικότητα (confidentiality)**, τη **διαθεσιμότητα (availability)** και την **ακεραιότητα (integrity)**[1] των κωδικών πρόσβασης, έτσι ώστε όλοι οι εξουσιοδοτημένοι χρήστες - και κανένας μη εξουσιοδοτημένος χρήστης - να μπορούν να χρησιμοποιούν επιτυχώς τους κωδικούς πρόσβασης, όπως απαιτείται. Η ακεραιότητα και η διαθεσιμότητα πρέπει να διασφαλίζονται με τυπικά στοιχεία ελέγχου ασφάλειας δεδομένων, όπως η χρήση **λιστών ελέγχου πρόσβασης (ACL)**, οι οποίες εφαρμόζονται από την πολιτική ελέγχου πρόσβασης DAC, για την αποφυγή της αντικατάστασης των κωδικών πρόσβασης από τους εισβολείς και την ασφαλή δημιουργία αντιγράφων ασφαλείας των αρχείων κωδικού πρόσβασης. Η διασφάλιση της εμπιστευτικότητας των κωδικών πρόσβασης είναι πολύ πιο δύσκολη και περιλαμβάνει έναν αριθμό ελέγχων ασφαλείας μαζί με αποφάσεις που περιλαμβάνουν τα ίδια τα χαρακτηριστικά των κωδικών πρόσβασης. Για παράδειγμα, η απαίτηση να είναι μεγάλοι και περίπλοκοι οι κωδικοί πρόσβασης καθιστά λιγότερο πιθανό ότι οι εισβολείς θα τους μαντέψουν ή θα τους σπάσουν, αλλά καθιστά επίσης τους κωδικούς πρόσβασης πιο δύσκολο για τους χρήστες να το θυμούνται και, ως εκ τούτου, είναι πιο πιθανό να αποθηκευτούν ανασφαλή. Αυτό αυξάνει την πιθανότητα οι χρήστες να αποθηκεύουν τους κωδικούς πρόσβασής τους με ασφάλεια και να τους εκθέτουν σε εισβολείς. [8]

[1] Τριάδα CIA: μοντέλο ασφάλειας για την ανάπτυξη των πολιτικών ασφαλείας

Οι οργανισμοί επίσης πρέπει να παρέχουν στον χρήστη και την δυνατότητα να εκχωρούν ένα αναγνωριστικό χρήστη (username) σε συνδυασμό με τον κωδικό πρόσβασης, έτσι ώστε το έργο του εισβολέα - και οποιουδήποτε μη εξουσιοδοτημένου χρήστη - να γίνεται πιο δύσκολο, καθώς θα πρέπει να βρει και τα 2 πεδία. Παράλληλα, το σύστημα δεν θα πρέπει να ενημερώνει καμία οντότητα, εξουσιοδοτημένη ή μη, για το που βρίσκεται το λάθος σε περίπτωση λάθος στοιχείων. Παρακάτω παρουσιάζονται 2 περιπτώσεις εισαγωγής λάθος στοιχείων.

- **Περίπτωση 1: Ενημέρωση του χρήστη για εισαγωγή λανθασμένου ονόματος**

Αυτή η πολιτική είναι λάθος, καθώς ο εισβολέας - μη εξουσιοδοτημένη οντότητα, ενημερώνεται ότι έχει εισάγει λάθος όνομα χρήστη, με αποτέλεσμα να γνωρίζει πλέον ότι το συγκεκριμένο όνομα δεν είναι έγκυρο. Αυτή η περίπτωση θα μπορούσε να θεωρηθεί πολύ επικίνδυνη για το σύστημα, καθώς η μη εξουσιοδοτημένη οντότητα θα μπορούσε να είχε φτιάξει μια λίστα με ονόματα χρηστών και να τα δοκιμάζει. Κάθε φορά που θα εισάγει ένα όνομα και θα βλέπει ότι δεν είναι αποδεκτό η λίστα αυτή θα μικραίνει επικίνδυνα.



Εικόνα 5: Το σύστημα ενημερώνει τον χρήστη ότι έχει εισάγει λάθος όνομα χρήστη

- **Περίπτωση 2: Ενημέρωση χρήστη για την αδυναμία πρόσβασης στο σύστημα**

Αυτή η πολιτική θεωρείται πολύ πιο ασφαλής, καθώς ο χρήστης ενημερώνεται απλά ότι δεν έχει πρόσβαση στο σύστημα χωρίς να γνωρίζει πιο πεδίο έχει κάνει λάθος. Αυτομάτως οι πιθανότητες του εισβολέα - μη εξουσιοδοτημένης οντότητας να μην μπορέσει να εισέλθει στο σύστημα, αυξάνονται πολύ σημαντικά, αφού οι πιθανότητες να βρει τον σωστό συνδυασμό όνομα χρήστη - κωδικός πρόσβασης είναι πολύ λιγότερες από ότι πριν.

Σύμφωνα με τις "οδηγίες" (guidelines) κωδικών πρόσβασης που δημοσίευσε ο οργανισμός **NIST**[2] για το πως πρέπει να είναι ένας σωστός, δυνατός κωδικός πρόσβασης όλοι οι χρήστες - οργανισμοί πρέπει να έχουν υπόψιν τους: [9]

- Οι κωδικοί πρόσβασης που δημιουργούνται από τον χρήστη πρέπει να έχουν τουλάχιστον 8 χαρακτήρες.
- Οι κωδικοί πρόσβασης που δημιουργούνται από μηχανή πρέπει να έχουν τουλάχιστον 6 χαρακτήρες.
- Οι χρήστες θα πρέπει να μπορούν να δημιουργούν κωδικούς πρόσβασης έως και 64 χαρακτήρες.
- Πρέπει να επιτρέπονται όλοι οι χαρακτήρες ASCII / Unicode, συμπεριλαμβανομένων των emoji και των κενών.
- Οι αποθηκευμένοι κωδικοί πρόσβασης πρέπει να κατακερματιστούν (hashed) και να αλατιστούν (salted) και να μην περικοπούν (truncated) ποτέ.
- Οι υποψήφιοι κωδικοί πρόσβασης πρέπει να συγκρίνονται με τις βάσεις δεδομένων παραβίασης κωδικών πρόσβασης και να απορρίπτονται εάν υπάρχει αντιστοιχία.
- Οι κωδικοί πρόσβασης δεν πρέπει να λήγουν. Οι χρήστες πρέπει να εμποδίζονται να χρησιμοποιούν διαδοχικούς (π.χ. "1234") ή επαναλαμβανόμενους (π.χ. "aaaa") χαρακτήρες.
- Ο έλεγχος ταυτότητας δύο παραγόντων (2FA) δεν πρέπει να χρησιμοποιεί SMS για κωδικούς.

[2] NIST: National Institute of Standards and Technology, Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (NIST, μια ομοσπονδιακή υπηρεσία των ΗΠΑ)

- Ο έλεγχος ταυτότητας βάσει γνώσεων (Knowledge-based authentication, KBA), όπως "Ποιο ήταν το όνομα του πρώτου κατοικίδιου ζώου σας;", δεν πρέπει να χρησιμοποιείται.
- Πρέπει να επιτραπεί στους χρήστες 10 αποτυχημένες προσπάθειες κωδικού πρόσβασης πριν κλειδωθούν από ένα σύστημα ή μια υπηρεσία.
- Οι κωδικοί πρόσβασης δεν πρέπει να έχουν υποδείξεις.
- Δεν πρέπει να χρησιμοποιούνται απαιτήσεις πολυπλοκότητας, π.χ. απαιτούν ειδικούς χαρακτήρες, αριθμούς, κεφαλαία κλπ.
- Δεν πρέπει να επιτρέπονται συγκεκριμένες λέξεις, όπως το όνομα της υπηρεσίας, το όνομα χρήστη του χρήστη κλπ.

2.1. Έλεγχος Ταυτότητας 2 Παραγόντων (2FA)

Καθώς ο αριθμός των παραβιάσεων ασφαλείας αυξάνεται όλο και περισσότερο, ο **έλεγχος ταυτότητας 2 παραγόντων (Two - Factor Authentication)** έχει γίνει ένα ουσιαστικό εργαλείο ασφάλειας ιστού, επειδή μετριάζει τον κίνδυνο παραβίασης που σχετίζεται με τα **διαπιστευτήρια σύνδεσης (credentials)** [4], [5]. Διαπιστευτήρια σύνδεσης, υπάρχουν στα μέσα κοινωνικής δικτύωσης (social media) αλλά και σε εφαρμογές ιστού. Επιπροσθέτως, σύμφωνα με το **Hacker Noon**[3] , ένας από τους κυριότερους λόγους είναι οι συνεχόμενες παραβιάσεις δεδομένων (data breaches), οι οποίες έχουν διαθέσει πολλά ζεύγη διευθύνσεων ηλεκτρονικού ταχυδρομείου (emails) - κωδικών πρόσβασης προς πώληση στο Σκοτεινό Ιστό (Dark Web)[4] , έχουν κάνει τους κωδικούς πρόσβασης λιγότερο ασφαλείς. Οι περισσότεροι χρήστες χρησιμοποιούν τον ίδιο κωδικό πρόσβασης σε πολλούς ιστοτόπους και λογαριασμούς, με αποτέλεσμα οι εισβολείς να μπορούν να συνδέσουν γνωστά ζεύγη διευθύνσεων ηλεκτρονικού ταχυδρομείου - κωδικών πρόσβασης σε πολλούς ιστοτόπους και να δουν ποιος από αυτούς τους παρέχει πρόσβαση[12].

[3] Hacker Noon: Ιστότοπος τεχνολογικών μεσών, που απαρτίζεται από 7000 (και περισσότερους) συνεισφέροντες συγγραφείς, οι οποίοι δημοσιεύουν τεχνολογικές ιστορίες σχετικά με τον προγραμματισμό, blockchains κ.α.

[4] Σκοτεινός Ιστός (Dark Web): είναι ένας ανώνυμος διαδικτυακός ιστός, στον οποίο σκιώδεις χρήστες έχουν πρόσβαση σε κρυφές υπηρεσίες.

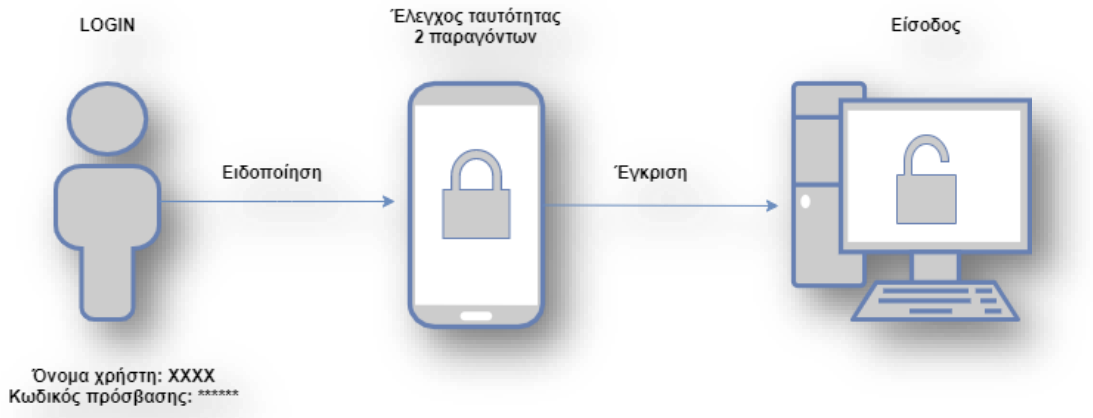
Ο έλεγχος ταυτότητας 2 παραγόντων είναι ο απλούστερος τρόπος επιβεβαίωσης ότι οι χρήστες είναι αυτοί που πραγματικά ισχυρίζονται ότι είναι. Ενισχύει την ασφάλεια, καθώς απαιτεί ένα επιπλέον επίπεδο ελέγχου εκτός από το όνομα χρήστη και τον κωδικό πρόσβασης. Ο επιπλέον αυτός έλεγχος μπορεί να ανήκει στους παρακάτω τύπους[10]:

- **Εφαρμογές Ελέγχου Ταυτότητας (Authenticator Apps)**: Εφαρμογές smartphone που χειρίζονται την διαδικασία ταυτοποίησης ελέγχου ταυτότητας του 2ου παράγοντα ως ειδοποιήσεις push (push notifications).
- **Διακριτά ασφαλείας U2F (U2F Security Tokens)**: Πρότυπο ελέγχου ταυτότητας που χρησιμοποιεί ένα USB και έναν διακομιστή (server). Ο χρήστης κάνει έλεγχο ταυτότητας πατώντας το U2F πλήκτρο που έχει εισαχθεί στην θύρα USB του υπολογιστή του.
- **Διακριτά Υλικού (Hardware Tokens)**: Συσκευή η οποία έχει προγραμματιστεί για την δημιουργία ενός κωδικού πρόσβασης, που πρέπει να πληκτρολογηθεί στον prompt 2 παραγόντων. Επαλήθευση ταυτότητας με το πάτημα ενός κουμπιού.
- **Κωδικοί Πρόσβασης SMS (SMS Passcodes)**: Κωδικός πρόσβασης που στέλνεται μέσω SMS στο κινητό του χρήστη και πρέπει να πληκτρολογηθεί στον prompt 2 παραγόντων. ΔΕΝ ΣΥΝΙΣΤΑΤΑΙ!
- **Τηλεφωνικά Callbacks (Phone Callbacks)**: Καλεί το τηλέφωνό του χρήστη, περιμένει να το σηκώσει και να πατήσει ένα οποιοδήποτε πλήκτρο για έλεγχο ταυτότητας προτού παραχωρήσει πρόσβαση στον λογαριασμό.
- **Κωδικοί Πρόσβασης για Κινητές Συσκευές (Mobile Passcodes)**: Παρόμοια με τα SMS, μόνο που μπορεί να παράγει νέο μοναδικό κωδικό πρόσβασης που πρέπει να πληκτρολογηθεί στον prompt 2 παραγόντων. Είναι γνωστοί ως και one-time passwords (OTP).
- **Βιομετρικά (Biometrics)**: Δαχτυλικό αποτύπωμα ή αναγνώριση προσώπου ως 2 παράγοντας για να αποκτήσει ο χρήστης πρόσβαση.

Εάν ένας κωδικός πρόσβασης έχει παραβιαστεί (hacked), “μαντευτεί” (guessed), πλαστογραφηθεί (phished) ο έλεγχος πρόσβασης 2 παραγόντων εμποδίζει τον εισβολέα να αποκτήσει πρόσβαση χωρίς να ταυτοποιηθεί από τον 2ο παράγοντα.

Η διαδικασία που ακολουθεί ο χρήστης για τον έλεγχο 2 παραγόντων διαφέρει αναλόγως την μέθοδο που πραγματοποιείται, αλλά είναι σχεδόν η ίδια σε όλες και είναι η εξής:

1. Ο χρήστης συνδέεται στον λογαριασμό του με το όνομα χρήστη και τον κωδικό πρόσβασης.
2. Ο κωδικός πρόσβασης επικυρώνεται από τον διακομιστή ελέγχου ταυτότητας και αν είναι ο ίδιος, τότε ο χρήστης προχωράει στον 2ο παράγοντα του ελέγχου ταυτότητας.
3. Ο διακομιστής ελέγχου ταυτότητας στέλνει ένα μοναδικό κωδικό στην μέθοδο του δεύτερου παράγοντα του χρήστη.
4. Ο χρήστης επιβεβαιώνει την ταυτότητά παρέχοντας έτσι τον πρόσθετο έλεγχο για τον δεύτερο παράγοντα.



Εικόνα 6: Έλεγχος Ταυτότητας 2 Παραγόντων

Ο έλεγχος ταυτότητας 2 παραγόντων είναι ουσιαστικά ένα υποσύνολο του **ελέγχου ταυτότητας πολλαπλών παραγόντων (MFA)**. Ο έλεγχος ταυτότητας πολλαπλών παραγόντων και συνεπώς ο έλεγχος 2 παραγόντων παρέχει κάποια σημαντικά οφέλη στους οργανισμούς αλλά και στους χρήστες. Ένα από αυτά είναι η βελτιωμένη εμπιστοσύνη, καθώς ο παραπάνω έλεγχος μπορεί να προστατεύσει τον οργανισμό ή τον χρήστη από μη εξουσιοδοτημένες οντότητες, οι οποίες μπορούν να προκαλέσουν ζημιές. Επιπλέον, ο MFA μπορεί να κάνει έναν οργανισμό πιο έμπιστο, αφού το προσωπικό ή οι χρήστες - και οι πελάτες κυρίως - μπορούν να εκτιμήσουν την επιπλέον ασφάλεια που προσφέρει ένας τέτοιος έλεγχος. Παράλληλα, ένα άλλος όφελος είναι το μειωμένο κόστος, διότι η έλλειψη ασφάλειας μπορεί να οδηγήσει σε μια επίθεση από μια μη εξουσιοδοτημένη οντότητα, η οποία μπορεί να προκαλέσει ζημιές αυξημένου κόστους στο σύστημα, συνεπώς και στον ίδιο

τον οργανισμό. Τέλος, παρέχει ευκολότερες και πιο φιλικές συνδέσεις ως προς τους χρήστες, καθώς η τεχνολογία εξελίσσεται και πλέον ο έλεγχος μπορεί να γίνει παθητικά, όπως για παράδειγμα βιομετρικά (δαχτυλικό αποτύπωμα ή αναγνώριση προσώπου) [11].

2.2. Αποθήκευση Κωδικών Πρόσβασης (Password Storage)

Για την χρήση του ελέγχου ταυτότητας, οι κωδικοί πρόσβασης εφαρμογής αποθηκεύονται σε κεντρικούς υπολογιστές (hosts) μαζί με το λειτουργικό σύστημα. Εάν οι κωδικοί πρόσβασης δεν έχουν αποθηκευτεί σωστά, η πιθανότητα υποκλοπής αυτών από εισβολείς είτε με φυσικό είτε με λογικό τρόπο είναι αρκετά αυξημένη. Οι κωδικοί πρόσβασης δεν πρέπει να αποθηκεύονται χωρίς πρόσθετα στοιχεία ελέγχου ασφάλειας. Στην συνέχεια θα εξετάσουμε όλα τα δυνατά στοιχεία ελέγχου ασφάλειας που μπορούν να κάνουν τους κωδικούς πρόσβασης ασφαλής [8].

- Περίπτωση 1: Αποθήκευση κωδικών πρόσβασης ως απλό κείμενο (plaintext)

Η αποθήκευση των κωδικών πρόσβασης ως ένα απλό κείμενο παρέχει μηδενική ασφάλεια στο σύστημα, καθώς οποιαδήποτε οντότητα εξουσιοδοτημένη (ο διαχειριστής) ή μη (ένας εισβολέας), μπορεί να δει ποιος κωδικός πρόσβασης αντιστοιχεί σε ποιον χρήστη. Ακόμα και ο διαχειριστής δεν πρέπει να γνωρίζει ποιος κωδικός πρόσβασης αντιστοιχεί σε ποιον, πόσο μάλλον ένας εισβολέας ο οποίος μπορεί να είναι αρκετά ικανός ώστε να συνδυάσει το μοτίβο του κωδικού πρόσβασης του χρήστη και με άλλους πιθανούς συνδυασμούς κωδικών που μπορεί να οδηγούν στην πρόσβαση και σε άλλους λογαριασμούς του ίδιου χρήστη. Η αποθήκευση των κωδικών πρόσβασης ως απλό κείμενο δεν συνίσταται! [13].

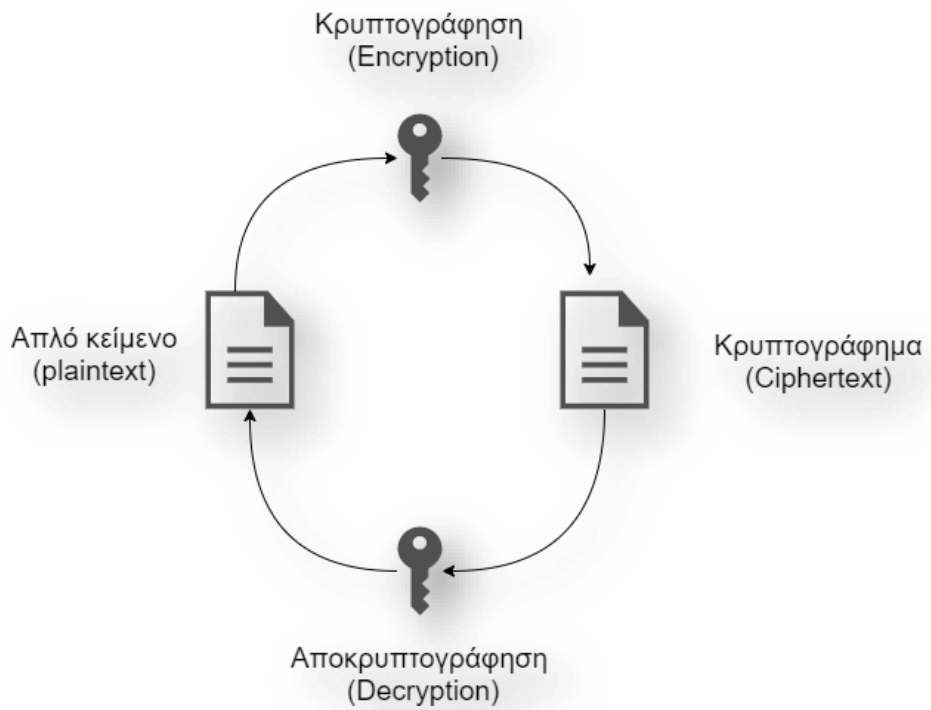
Παράδειγμα 1: Αποθήκευση κωδικού πρόσβασης ως απλό κείμενο

vasiliki: !mthe#1fAN!

- Περίπτωση 2: Αποθήκευση κρυπτογραφημένων κωδικών πρόσβασης

Η κρυπτογράφηση των κωδικών πρόσβασης είναι πιο ασφαλής μέθοδος, καθώς η μη εξουσιοδοτημένη οντότητα δεν γνωρίζει με ποιον αλγόριθμο κρυπτογραφήθηκε ο κωδικός

πρόσβασης και με ποιο κλειδί, με αποτέλεσμα να χρειαστεί αρκετό χρονικό διάστημα για να αποκρυπτογραφήσει και να αντλήσει την πληροφορία που θέλει. Η κρυπτογράφηση λειτουργεί ως εξής:



Εικόνα 7: Βασικό Σχήμα Κρυπτογράφησης

Στον τομέα της κρυπτογράφησης υπάρχουν πολλοί αλγόριθμοι. Αυτό δεν σημαίνει ότι και όλοι ασφαλείς, αφού κάποιοι από αυτούς σπάνε πολύ εύκολα και γρήγορα με την εξέλιξη της τεχνολογίας πλέον.

Έχοντας κρυπτογραφήσει έναν κωδικό πρόσβασης, για παράδειγμα με τον αλγόριθμο AES-256, ο οποίος έχει μήκος κλειδιού 256 bits (32 bytes), η μη εξουσιοδοτημένη οντότητα όταν αποκτήσει πρόσβαση στο shadow file του συστήματος, θα αντικρίσει κάτι πολύ πιο δυσνόητο από πριν στο πεδίο του κωδικού πρόσβασης.

Παράδειγμα 2: Αποθήκευση κρυπτογραφημένου κωδικού πρόσβασης

```
vasiliki: TgoOEy7olWo2TU3K0MRFrEyNu6bI5xnXxR27gL3hu8A=
```

Ο παραπάνω κωδικός πρόσβασης κρυπτογραφήθηκε με τον συμμετρικό αλγόριθμο AES-256 και με κλειδί '0123456789abcdef". Η διαφορά με το απλό κείμενο είναι φανερή. Για να μπορέσει η μη εξουσιοδοτημένη οντότητα να ανακτήσει τον κωδικό πρόσβασης θα πρέπει να γνωρίζει το κλειδί. Το κλειδί αποκρυπτογράφησης μπορεί να είναι αποθηκευμένο σε έναν άλλο διακομιστή (server) και να ανακτάται μόνο όταν είναι απαραίτητο μέσω ενός διακομιστή επαλήθευσης κωδικού. Με αυτόν τον τρόπο το κλειδί είναι θεωρητικά ασφαλές και δεν χρειάζεται οι κωδικοί πρόσβασης να αποθηκεύονται ως απλό κείμενο. Το κλειδί δεν πρέπει να το γνωρίζει ούτε ο διαχειριστής ούτε κανένας χρήστης του συστήματος.

Γενικά, όλη η ισχύς τής κρυπτογραφίας βρίσκεται στο κλειδί. Όσο πιο μεγάλο είναι σε μήκος το κλειδί, τόσο πιο δύσκολο είναι να σπάσει ένας αλγόριθμος. Ο παραπάνω πίνακας μας παρουσίαζε κάποιους συμμετρικούς αλγόριθμους, που σημαίνει ότι εφόσον υπάρχει ένα κλειδί για κρυπτογράφηση, το ίδιο κλειδί θα αποκρυπτογραφεί τον κωδικό πρόσβασης. Αυτό δεν είναι θεμιτό, καθώς αν η μη εξουσιοδοτημένη οντότητα πάρει το κλειδί θα μπορέσει να ανακτήσει και όλους τους κωδικούς πρόσβασης, αφού οι παραπάνω αλγόριθμοι είναι αντιστρέψιμοι. Συμπερασματικά, η κρυπτογράφηση με αναστρέψιμο κρυπτογραφικό αλγόριθμο δεν είναι ούτε αυτή ασφαλής και δεν συνίσταται!

- Περίπτωση 3: Αποθήκευση κατακερματισμένων κωδικών πρόσβασης

Την λύση στο προηγούμενο πρόβλημα δίνουν οι μονόδρομοι (one-way) αλγόριθμοι κρυπτογράφησης οι οποίοι είναι γνωστοί ως αλγόριθμοι κρυπτογράφησης κατακερματισμού ή συναρτήσεις κατακερματισμού (hash functions) και είναι αυτοί που πλέον έχουν τον πρωταγωνιστικό ρόλο στην κρυπτογράφηση των συνθηματικών. Θεωρούνται μονόδρομοι, διότι ο κρυπτογραφημένος κωδικός που παράγεται έχοντας εφαρμόσει πάνω του μια συνάρτηση κατακερματισμού είναι μη αναστρέψιμος. Πιο αναλυτικά, ένας αλγόριθμος κατακερματισμού παίρνει μια αυθαίρετη ποσότητα δεδομένων, εφαρμόζει έναν μαθηματικό τύπο που είναι γνωστός ως συνάρτηση κατακερματισμού και τέλος παράγεται μια τιμή σταθερού μήκους η οποία είναι η τιμή κατακερματισμού ή αλλιώς message digest [11]. Κάποιοι γνωστοί αλγόριθμοι κατακερματισμού είναι ο MD5, SHA-1, SHA-2, NTLM και LANMAN.

Το Hashing βασίζεται σε κάποιες ''αρχές'':

1. Οι συναρτήσεις κατακερματισμού είναι μονόδρομες.

2. Μια καλή συνάρτηση κατακερματισμού δεν πρέπει να δημιουργεί συγκρούσεις (collisions), δηλαδή δύο διαφορετικά σύνολα δεδομένων δεν πρέπει να παράγουν την ίδια έξοδο (τιμή κατακερματισμού).
3. Δεν θα πρέπει να μπορούμε να αλλάξουμε τίποτα που να αφορά την είσοδο αν δεν αλλάξουμε επίσης και το hash.

Εικόνα 8: Συνάρτηση Κατακερματισμού

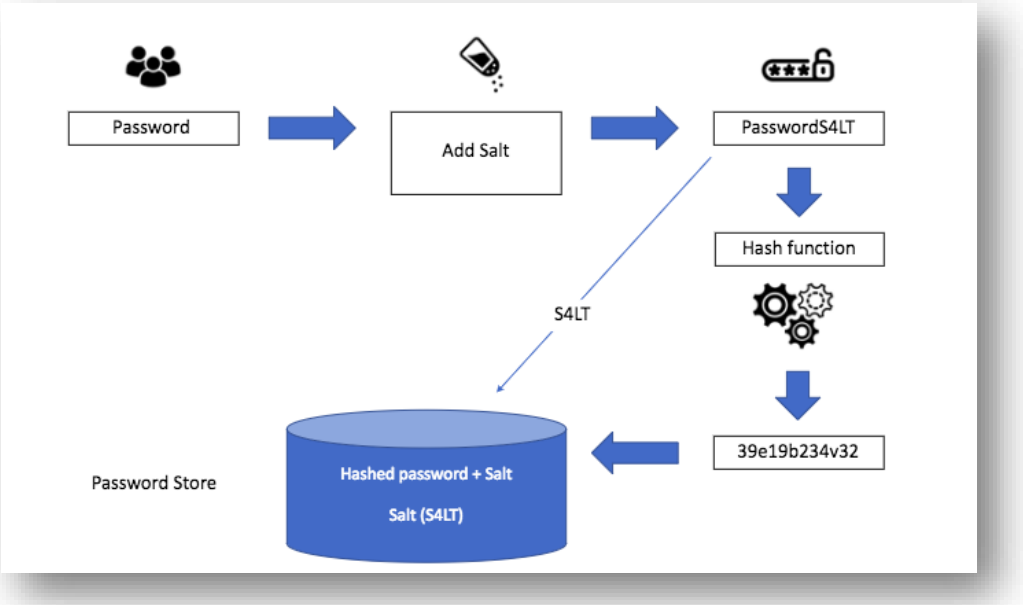
Παράδειγμα 3: Αποθήκευση κατακερματισμένου κωδικού πρόσβασης

vasiliki: 6f0b0941c9f619d2fc974f5755ff6b29=

Ο παραπάνω κωδικός πρόσβασης κρυπτογραφήθηκε με τον αλγόριθμο κατακερματισμού MD5.

Ένα μειονέκτημα που έχουν όμως οι αλγόριθμοι αυτοί είναι ότι παράγουν την ίδια τιμή κατακερματισμού για το ίδιο κείμενο εισόδου, συνεπώς παράγουν ίδια έξοδο σε περίπτωση που 2 χρήστες χρησιμοποιήσουν τον ίδιο κωδικό πρόσβασης. Αυτό σημαίνει ότι ένας επιτιθέμενος μπορεί να αποκτήσει πρόσβαση σε περισσότερους από έναν λογαριασμούς. Επιπρόσθετα, οι "απλοί" αλγόριθμοι κατακερματισμού είναι ευαίσθητοι στις επιθέσεις **Λεξικού (Dictionary attack)** και σε **Πίνακες Ουράνιου Τόξου (Rainbow tables)** , καθώς ο εισβολέας μπορεί να είναι αρκετά ικανός έτσι ώστε να πραγματοποιεί επίθεση Λεξικού και να ανακτά κωδικούς πρόσβασης από ήδη γνωστά συνθηματικά ή από προ-υπολογισμένες τιμές.

Το **salt** δίνει την λύση στα παραπάνω προβλήματα καθώς, ενισχύει τον αλγόριθμο κατακερματισμού και τον καθιστά πιο πολύπλοκο. Το salt είναι μια μοναδική συμβολοσειρά που προστίθεται στον κωδικό πρόσβασης κατά την διάρκεια του κατακερματισμού. Ο επιτιθέμενος χρειάζεται πολύ χρόνο και χρήματα, για να σπάσει τα κατακερματισμένα συνθηματικά, εφόσον το salt είναι μοναδικό για κάθε χρήστη και συνεπώς χρειάζεται να σπάσει το κάθε salt χωριστά και όχι μόνο έναν κατακερματισμό που αποκρυπτογραφεί όλους τους κωδικούς πρόσβασης. Επιπρόσθετα, το salt δημιουργεί διαφορετικούς κατακερματισμούς ακόμα και αν οι κωδικοί πρόσβασης δύο χρηστών είναι ίδιοι κι αυτό οφείλεται στα διαφορετικά salts που παράγονται, οπότε αυτό έχει ως αποτέλεσμα την επιπλέον ασφάλεια των συνθηματικών και κάνοντας έτσι πιο δύσκολο το έργο του επιτιθέμενου[15] .



Εικόνα 9: Συνάρτηση Κατακερματισμού με αλάτι [13]

Κεφάλαιο 3: Επιθέσεις στους Κωδικούς πρόσβασης

Οι κωδικοί πρόσβασης (passwords) αποτελούν τον πιο συνηθισμένο τρόπο επαλήθευσης δικαιώματος πρόσβασης σε ευαίσθητες ή προσωπικές πληροφορίες. Μερικές μορφές κωδικών είναι οι κωδικοί PIN χρεωστικών/πιστωτικών καρτών, κωδικοί PIN καρτών SIM και περίπλοκοι κωδικοί αλφαριθμητικών στοιχείων που χρησιμοποιούνται εκτενώς στα προσωπικά υπολογιστικά συστήματα (προσωπικοί ηλεκτρονικοί υπολογιστές, φορητές συσκευές και λογαριασμοί ιστοσελίδων). Μερικές από τις θετικές πτυχές των κωδικών πρόσβασης είναι η απλότητα, η απουσία χρέωσης για την χρήση τους και οι διάφοροι μηχανισμοί που διευκολύνουν την δημιουργία και την διαχείριση τους (εργαλεία αποθήκευσης κωδικών πρόσβασης π.χ. LastPass, Mozilla Passwords, κτλ.). Βέβαια με την εξέλιξη της επιστήμης των υπολογιστών, οι κωδικοί πρόσβασης πλέον καθίστανται μια από τις πιο αδύναμες μορφές περιορισμού πρόσβασης, καθώς από μελέτες έχει διαπιστωθεί ότι το 80% των κενών ασφαλείας έχει προκληθεί από κωδικούς πρόσβασης κακής ποιότητας [14]. Οι ευπάθειες των κωδικών πρόσβασης είναι πολύ δύσκολο να περιοριστούν καθώς αρκεί ένας κακός κωδικός για να εκθέσει σε κίνδυνο μια ολόκληρη βάση δεδομένων κωδικών.

3.1. Ευπάθειες κωδικών πρόσβασης

Όταν μιλάμε για ευπάθειες κωδικών αναφερόμαστε στα τρωτά σημεία τα οποία μπορεί ένας επίδοξος εισβολέας να εκμεταλλευτεί για να διεισδύσει στα ευαίσθητα δεδομένα του συστήματος. Μπορούμε να πούμε ότι διακρίνουμε πως οι ευπάθειες βασίζονται σε τρεις παράγοντες:

- Ανθρώπινος παράγοντας (χρήστες)
- Διαχειριστής / Σχεδιαστής συστήματος
- Κυβερνοχώρος (Διαδίκτυο)

Ο **ανθρώπινος παράγοντας** μπορεί να χαρακτηριστεί ως η βασικότερη ευπάθεια που παρατηρείται στην χρήση κωδικών πρόσβασης[1][2]. Είναι λογικό όπως γνωρίζουμε από την τεχνολογία λογισμικού, πως ο άνθρωπος αποτελεί την πηγή σφαλμάτων και αδυναμιών κατά

την ανάπτυξη ενός λογισμικού, έτσι και στην αλυσίδα ασφάλειας μεταξύ συστήματος και δικτύου, ο χρήστης αποτελεί τον πιο αδύναμο κρίκο. Κατά την δημιουργία ενός κωδικού, ο χρήστης είναι αυτός που ευθύνεται πλήρως για την ποιότητα και την ισχύ του, επομένως η πρώτη ευπάθεια μπορεί να ονομαστεί ”αδύναμος χρήστης”. Οι περισσότεροι χρήστες αδιαφορούν για την σημαντικότητα της διαδικασίας επιλογής κωδικού, είτε επειδή την βρίσκουν χρονοβόρα και κουραστική, είτε λόγω έλλειψης γνώσης βασικών αρχών ασφαλείας. Βέβαια η τρωτότητα των κωδικών βασίζεται και στο εκάστοτε σενάριο στο οποίο βρίσκεται ο χρήστης, εισάγοντας έτσι την επόμενη ευπάθεια. Για παράδειγμα, η χρήση ενός δημόσιου υπολογιστή για την σύνδεση σε έναν λογαριασμό αυξάνει τον κίνδυνο απώλειας του κωδικού αρχικά, με φυσικό τρόπο (να τον δει κάποιος διπλανός), δεύτερον με τεχνολογικό τρόπο (να υποκλαπεί από κάποιο κακόβουλο λογισμικό) και τέλος λόγω αμέλειας, με τον χρήστη να ξεχάσει να κάνει αποσύνδεση από το σύστημα και να το εγκαταλείψει [1].

Όμως, ακόμα και αν ο χρήστης κατέχει κάποια απαιτούμενη γνώση και προσοχή μπορεί να βρεθεί εκτεθειμένος αν ο **σχεδιαστής** ή ο **διαχειριστής** του συστήματος αδιαφορήσει για την ασφάλεια αυτού, ορίζοντας έτσι την δεύτερη ευπάθεια που είναι ο “κακός σχεδιασμός”. Συναντιούνται συχνά συστήματα, είτε τοπικά (λογισμικά) είτε διαδικτυακά (ιστοσελίδες), τα οποία παρουσιάζουν τρωτότητες και ελλείψεις λόγω κακού σχεδιασμού και υλοποίησης του πρωτοκόλλου ασφαλείας. Η προηγούμενη περίπτωση οδηγεί το σύστημα να είναι ευάλωτο σε επιθέσεις, με χειρότερη πτυχή την ανικανότητα των χρηστών να αντιδράσουν. Ακόμα και η φύση των κωδικών μπορεί να κουράσει τους χρήστες και τους σχεδιαστές [1], με κύρια αιτία την απαίτηση για πλήρη προσοχή και συντήρηση τους ως προς τις αλλαγές ασφαλείας. Συχνά έχουμε συστήματα τα οποία απαιτούν από τους χρήστες την αλλαγή του κωδικού ανά κάποιο χρονικό διάστημα, το οποίο σε βάθος χρόνου μπορεί να κουράσει τόσο τους χρήστες, όσο και τους διαχειριστές, οδηγώντας τους σε μειωμένη ενασχόληση και κατά συνέπεια την δημιουργία κενού ασφαλείας.

Τέλος, μερίδιο ευθύνης αποδίδεται και στον **κυβερνοχώρο**, δηλαδή το ίδιο το περιβάλλον συμβίωσης χρηστών και συστημάτων. Από τον σχεδιασμό του, ο κυβερνοχώρος παρέχει στους πιθανούς εισβολείς τα μέσα και τα εργαλεία για να εκτελέσουν τις επιθέσεις τους, αφήνοντας τους χρήστες και διαχειριστές των συστημάτων σε μια συνεχή κατάσταση αμύνης για την ακεραιότητα των δεδομένων τους.

3.2. Επίθεση εξαντλητικής αναζήτησης (Brute Force Attack)

Η επίθεση εξαντλητικής αναζήτησης (brute force), έχει ως σκοπό την “αποκάλυψη” των κρυπτογραφημένων κωδικών πρόσβασης που υπάρχουν σε ένα υπολογιστικό σύστημα μέσω της μεθόδου δοκιμή – και – σφάλμα (trial - and – error). Η παραπάνω μέθοδος δοκιμάζει εξαντλητικούς συνδυασμούς πιθανών συνθηματικών έχοντας ως σκοπό να βρει το σωστό που θα φανερώσει την κωδικοποιημένη τιμή. Λέγεται δοκιμή – και – σφάλμα, διότι δοκιμάζει αν ταιριάζει ο συνδυασμός που υπάρχει στο αρχείο του επιτιθέμενου με τα πιθανά συνθηματικά, με τον κωδικό πρόσβασης του θύματος. Αν ταιριάζει αποκαλύπτει την τιμή, εάν όχι το απορρίπτει και συγκρίνει το συνθηματικό με τον επόμενο συνδυασμό που έχει.

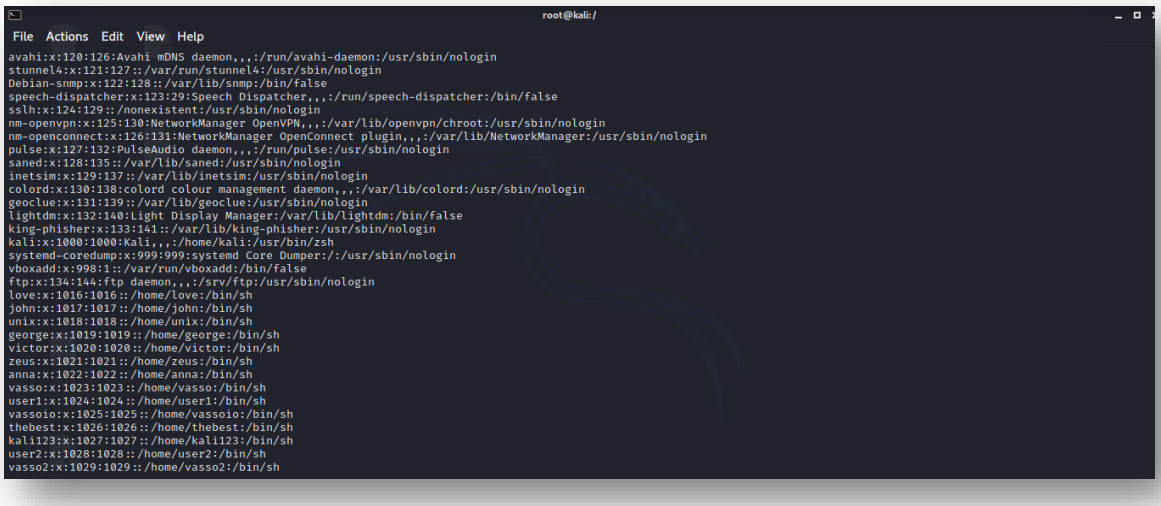
Σκοπός αυτής της επίθεσης είναι η γρήγορη αποκάλυψη του συνθηματικού και ταυτοχρόνως η πλήρης πρόσβαση στο υπολογιστικό σύστημα. Το πόσο γρήγορα θα πραγματοποιηθεί η επίθεση εξαρτάται από την πολυπλοκότητα του κωδικού και από την υπολογιστική ισχύς που διαθέτει ο εισβολέας. Αν ο εισβολέας διαθέτει χαμηλή ή ακόμα και μέτρια υπολογιστική ισχύς και πόρους και η πολυπλοκότητα του κωδικού είναι μεγάλη, τότε είναι πολύ δύσκολο και χρονικά αλλά και από θέμα κόστους μια τέτοια επίθεση να είναι επιτυχής. Η πολυπλοκότητα του συνθηματικού εξαρτάται τόσο από τον διαχειριστή του συστήματος όσο και από τον ίδιο τον χρήστη[21].

3.2.1. Θεωρητικό μέρος Επίθεσης Εξαντλητικής Αναζήτησης

Η επίθεση ωμής βίας λειτουργεί έχοντας προορίσει ένα συγκεκριμένο εύρος πιθανών λέξεων που μπορεί να αποκαλύψει τον κωδικό πρόσβασης του θύματος. Το έργο του επιτιθέμενου συνήθως σε τέτοιες περιπτώσεις δεν είναι εύκολο καθώς πλέον οι κωδικοί πρόσβασης δεν είναι τόσο αδύναμοι. Όμως αυτό εξαρτάται τόσο από τον σχεδιασμό του λειτουργικού συστήματος όσο και από τον ίδιον τον χρήστη. Πολλοί χρήστες χρησιμοποιούν εύκολα συνθηματικά για να μπορούν να τα θυμούνται, όπως για παράδειγμα το όνομα τους σε συνδυασμό με την ημερομηνία γέννησης τους, τα οποία κάνουν το έργο του εισβολέα πιο εύκολο, αφού πλέον με την εξέλιξη της τεχνολογίας μπορεί πολύ εύκολα να αντλήσει πληροφορίες για τον “στόχο” του.

Επιπρόσθετα, μπορεί να υπάρξουν συστήματα (όχι τόσο συχνά) που δεν χρησιμοποιούν κάποια πολιτική κωδικών πρόσβασης ή να ενημερώνουν τον χρήστη για την ισχύς του συνθηματικού του ή ακόμα να τον αναγκάζουν να ανανεώνει τον κωδικό του ανά κάποιο χρονικό διάστημα. Γνωρίζοντας την πολιτική ενός συστήματος, ο εισβολέας θα ξέρει τι μορφή συνθηματικών επιτρέπει το σύστημα και έτσι θα μπορεί να δημιουργήσει μικρότερες λίστες με τους πιθανούς συνδυασμούς. Όσο πιο απλό είναι ένα συνθηματικό τόσο πιο εύκολο το έργο του επιτιθέμενου[22].

Τα σύγχρονα συστήματα χρησιμοποιούν κατακερματισμένες συναρτήσεις (SHA512, bcrypt, MD5 κ.α.) για την κρυπτογράφηση των κωδικών πρόσβασης. Στο λειτουργικό σύστημα υπάρχει ένα **αρχείο κωδικών πρόσβασης (password file)** που περιέχει τα ονόματα των χρηστών μαζί με τα συνθηματικά τους όπως εμφανίζεται παρακάτω:



Εικόνα 10: Αρχείο Κωδικών Πρόσβασης του Συστήματος

Το αρχείο κωδικών πρόσβασης ακολουθεί την παρακάτω μορφή:

1ο πεδίο: Όνομα σύνδεσης (login name)..

2ο πεδίο: Προαιρετικός κρυπτογραφημένος κωδικός πρόσβασης.

3ο πεδίο: Αριθμητικό αναγνωριστικό χρήστη (User Identifier (UserID)).

4ο πεδίο: Αριθμητικό αναγνωριστικό ομάδας (Group Identifier (GroupID)).

5ο πεδίο: Όνομα χρήστη ή πεδίο σχολίων.

6ο πεδίο: Αρχικός κατάλογος χρήστη (Home Directory).

7ο πεδίο: Προαιρετικός διερμηνέας εντολών χρήστη/Κέλυφος (Shell).

Αν στο πεδίο του κωδικού πρόσβασης (2ο πεδίο) υπάρχει το "x", σημαίνει ότι στην πραγματικότητα ο κρυπτογραφημένος κωδικός πρόσβασης αποθηκεύεται σε ένα αρχείο που λέγεται **αρχείο κωδικών πρόσβασης shadow file**. Αν το πεδίο αυτό είναι κενό, τότε δεν απαιτείται κωδικός πρόσβασης για την ταυτοποίηση του χρήστη. Αν το πεδίο ξεκινά με ένα θαυμαστικό, σημαίνει ότι ο κωδικός πρόσβασης είναι κλειδωμένος κι οι υπόλοιποι χαρακτήρες στην γραμμή αντιπροσωπεύουν το πεδίο κωδικού πρόσβασης πριν κλειδωθεί ο κωδικός πρόσβασης. Εάν το πεδίο κωδικού πρόσβασης περιέχει κάποια συμβολοσειρά που δεν είναι έγκυρο αποτέλεσμα του **crypt(3)**[5], για παράδειγμα ! ή *, ο χρήστης δεν θα μπορεί να χρησιμοποιήσει κωδικό Unix (για παράδειγμα σε linux συστήματα) για να συνδεθεί (αλλά ο χρήστης μπορεί να συνδεθεί στο σύστημα με άλλα μέσα).

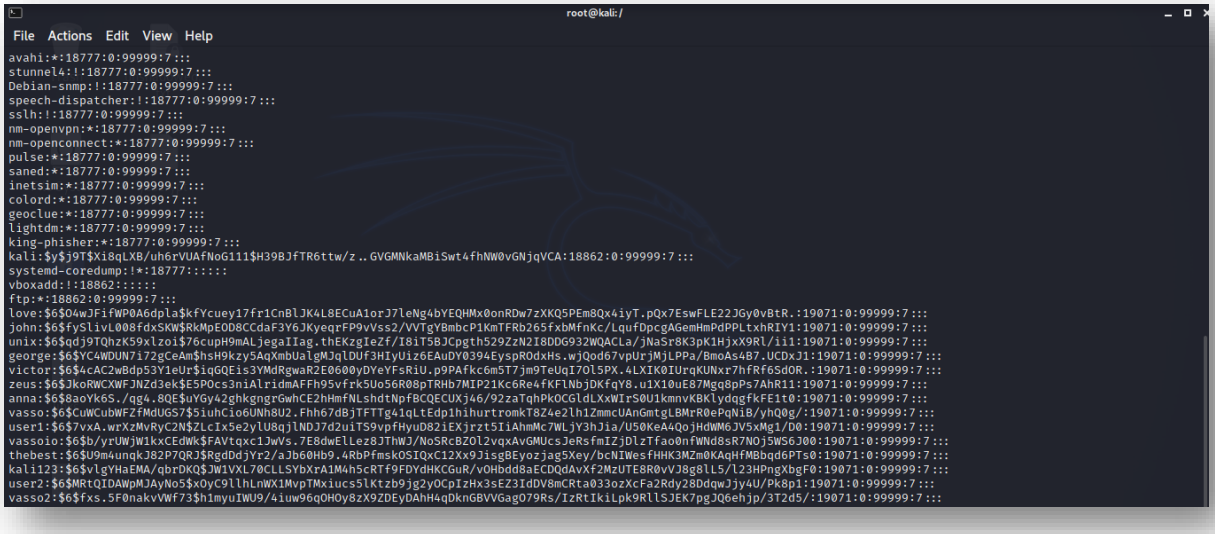
Το πεδίο σχολίων (5ο πεδίο) χρησιμοποιείται από διάφορα βοηθητικά προγράμματα συστήματος, όπως το finger(1)[6].

Το πεδίο αρχικός κατάλογος χρήστη (6ο πεδίο) παρέχει το όνομα του αρχικού καταλόγου εργασίας. Το πρόγραμμα σύνδεσης χρησιμοποιεί αυτές τις πληροφορίες για να ορίσει την τιμή της περιβαλλοντικής μεταβλητής **$HOME**.

Το πεδίο διερμηνέα εντολών (7ο πεδίο) παρέχει το όνομα του διερμηνέα γλώσσας εντολών του χρήστη ή το όνομα του αρχικού προγράμματος προς εκτέλεση. Το πρόγραμμα σύνδεσης χρησιμοποιεί αυτές τις πληροφορίες για να ορίσει την τιμή της περιβαλλοντικής μεταβλητής **$SHELL**. Εάν αυτό το πεδίο είναι κενό, έχει ως προεπιλογή την τιμή /bin/sh.

Ο εισβολέας αν αποκτήσει πρόσβαση σε αυτό το αρχείο ίσως και να μπορέσει να ανακτήσει τα συνθηματικά των χρηστών όμως, δεν είναι αυτό το αρχείο που φυλάσσονται οι κωδικοί πρόσβασης. Οι κωδικοί πρόσβασης φυλάσσονται σε ένα αρχείο που λέγεται **αρχείο κωδικών πρόσβασης shadow file** και η μορφή των συνθηματικών είναι δυσανάγνωστη καθώς είναι κρυπτογραφημένα. Η μορφή αυτού του αρχείου είναι η εξής:

[5] Τα crypt, crypt_r, crypt_rn και crypt_ra λειτουργούν μη αναστρέψιμα «hash» για αποθήκευση στη βάση δεδομένων κωδικών πρόσβασης του συστήματος, χρησιμοποιώντας τη μέθοδο του κατακερματισμού.
[6] Το finger εμφανίζει πληροφορίες σχετικά με τους χρήστες του συστήματος.



Εικόνα 11: Shadow File

1ο πεδίο: Όνομα σύνδεσης (login name)

2ο πεδίο: Κρυπτογραφημένος κωδικός πρόσβασης (encrypted password)

3ο πεδίο: Ημερομηνία τελευταίας αλλαγής κωδικού πρόσβασης

4ο πεδίο: Ελάχιστη ηλικία κωδικού πρόσβασης

5ο πεδίο: Μέγιστη ηλικία κωδικού πρόσβασης

6ο πεδίο: Περίοδος προειδοποίησης κωδικού πρόσβασης

7ο πεδίο: Περίοδος αδράνειας κωδικού πρόσβασης

8ο πεδίο: Ημερομηνία λήξης του λογαριασμού

9ο πεδίο: Δεσμευμένο πεδίο

Το όνομα σύνδεσης (1ο πεδίο), πρέπει να είναι ένα έγκυρο όνομα λογαριασμού, που υπάρχει στο σύστημα.

Το πεδίο του κρυπτογραφημένου κωδικού πρόσβασης (2ο πεδίο), μπορεί να είναι κενό, οπότε δεν απαιτούνται κωδικοί πρόσβασης για έλεγχο ταυτότητας στο καθορισμένο όνομα σύνδεσης. Ωστόσο, ορισμένες εφαρμογές που διαβάζουν το αρχείο /etc/shadow ενδέχεται να αποφασίσουν να μην επιτρέψουν καμία πρόσβαση εάν το πεδίο κωδικού πρόσβασης

είναι κενό. Εάν το πεδίο κωδικού πρόσβασης ξεκινά με ένα θαυμαστικό ή περιέχει κάποια συμβολοσειρά ισχύουν αυτά που ισχύουν και στο αρχείο κωδικών πρόσβασης.

Αν η ημερομηνία της τελευταίας αλλαγής κωδικού πρόσβασης (3ο πεδίο) έχει την τιμή 0 έχει μια ειδική σημασία, η οποία είναι ότι ο χρήστης πρέπει να αλλάξει τον κωδικό πρόσβασής του την επόμενη φορά που θα συνδεθεί στο σύστημα. Αν το πεδίο είναι κενό, σημαίνει ότι οι δυνατότητες αλλαγής κωδικού πρόσβασης βάση χρόνου είναι απενεργοποιημένες.

Η ελάχιστη ηλικία κωδικού πρόσβασης (4ο πεδίο) είναι ο αριθμός των ημερών που θα πρέπει να περιμένει ο χρήστης προτού του επιτραπεί να αλλάξει ξανά τον κωδικό πρόσβασής του. Ένα κενό πεδίο και η τιμή 0 σημαίνουν ότι δεν υπάρχει ελάχιστη ηλικία κωδικού πρόσβασης.

Η μέγιστη ηλικία κωδικού πρόσβασης (5ο πεδίο) είναι ο αριθμός των ημερών μετά τις οποίες ο χρήστης θα πρέπει να αλλάξει τον κωδικό πρόσβασής του. Μετά την πάροδο αυτού του αριθμού ημερών, ο κωδικός πρόσβασης μπορεί να εξακολουθεί να είναι έγκυρος. Όμως, θα πρέπει να ζητηθεί από τον χρήστη να αλλάξει τον κωδικό πρόσβασής του την επόμενη φορά που θα συνδεθεί. Ένα κενό πεδίο σημαίνει ότι δεν υπάρχει μέγιστη ηλικία κωδικού πρόσβασης, περίοδος προειδοποίησης κωδικού πρόσβασης και περίοδος αδράνειας κωδικού πρόσβασης. Εάν η μέγιστη ηλικία κωδικού πρόσβασης είναι μικρότερη από την ελάχιστη ηλικία κωδικού πρόσβασης, ο χρήστης δεν μπορεί να αλλάξει τον κωδικό πρόσβασής του.

Η περίοδος προειδοποίησης κωδικού πρόσβασης (6ο πεδίο) είναι ο αριθμός των ημερών πριν από τη λήξη ενός κωδικού πρόσβασης κατά τη διάρκεια των οποίων ο χρήστης πρέπει να προειδοποιηθεί. Ένα κενό πεδίο και η τιμή 0 σημαίνουν ότι δεν υπάρχει περίοδος προειδοποίησης κωδικού πρόσβασης.

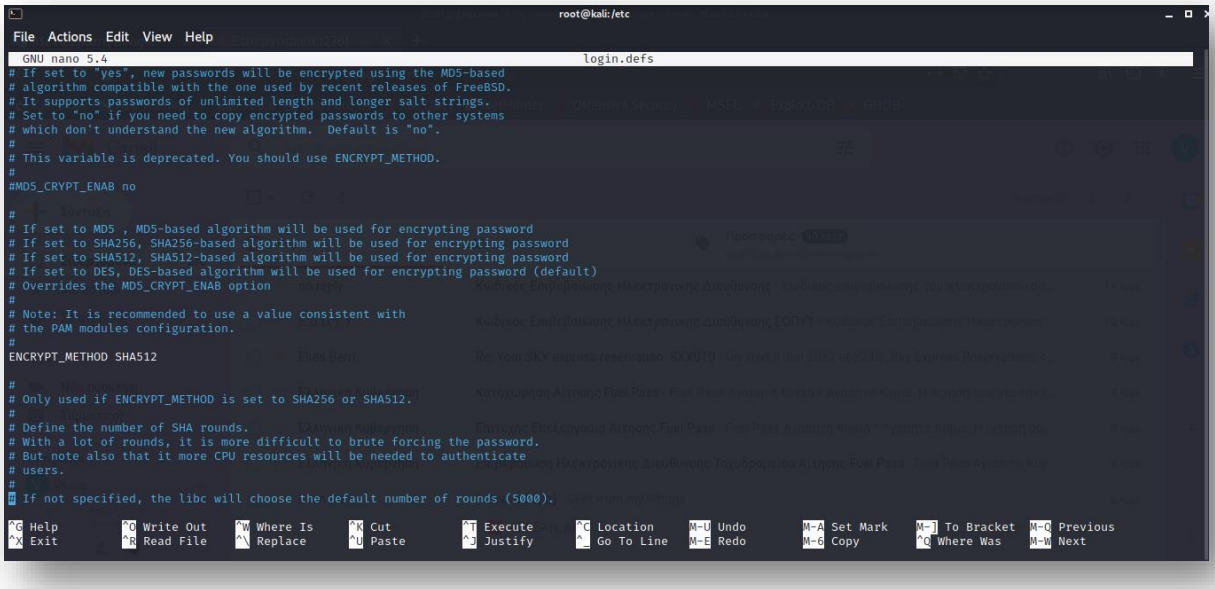
Η περίοδος αδράνειας κωδικού πρόσβασης (7ο πεδίο) είναι ο αριθμός των ημερών μετά τη λήξη ενός κωδικού πρόσβασης, κατά τη διάρκεια των οποίων ο κωδικός πρόσβασης θα πρέπει να είναι ακόμα αποδεκτός (και ο χρήστης θα πρέπει να ενημερώσει τον κωδικό πρόσβασής του κατά την επόμενη σύνδεση). Μετά τη λήξη του κωδικού πρόσβασης και την πάροδο αυτής της περιόδου λήξης, δεν είναι δυνατή η είσοδος χρησιμοποιώντας τον κωδικό

πρόσβασης του τρέχοντος χρήστη. Ο χρήστης θα πρέπει να επικοινωνήσει με τον διαχειριστή του. Ένα κενό πεδίο σημαίνει ότι δεν υπάρχει επιβολή μιας περιόδου αδράνειας.

Σε περίπτωση λήξης λογαριασμού (8ο πεδίο), ο χρήστης δεν επιτρέπεται να συνδεθεί, ενώ σε περίπτωση λήξης κωδικού πρόσβασης, ο χρήστης δεν επιτρέπεται να συνδεθεί χρησιμοποιώντας τον κωδικό πρόσβασής του. Ένα κενό πεδίο σημαίνει ότι ο λογαριασμός δεν θα λήξει ποτέ. Το δεσμευμένο πεδίο (9ο πεδίο) είναι δεσμευμένο για μελλοντική χρήση.

Η προστασία αυτού του αρχείου είναι πολύ σημαντική καθώς αν ο εισβολέας αποκτήσει πρόσβαση σε αυτό μπορεί να καταφέρει να ανακτήσει τα κρυπτογραφημένα συνθηματικά, διότι αν είναι καλός γνώστης των επιθέσεων και μπορεί να διαβάσει σωστά το παραπάνω αρχείο, μπορεί να καταλάβει ότι το σύστημα κρυπτογραφεί τα συνθηματικά με τον αλγόριθμο κατακερματισμού SHA-512, αφού το δεύτερο πεδίο ξεκινά με το πρόθεμα (prefix) **6** που αυτό υποδηλώνει ότι ο αλγόριθμος κατακερματισμού είναι ο παραπάνω. Ο αλγόριθμος όμως μπορεί να βρεθεί και μέσω του αρχείου login.defs γράφοντας στον τερματικό του Unix συστήματος:

1. Cd /etc
2. Nano login.defs



Εικόνα 12: Εμφάνιση του αρχείου Login.defs

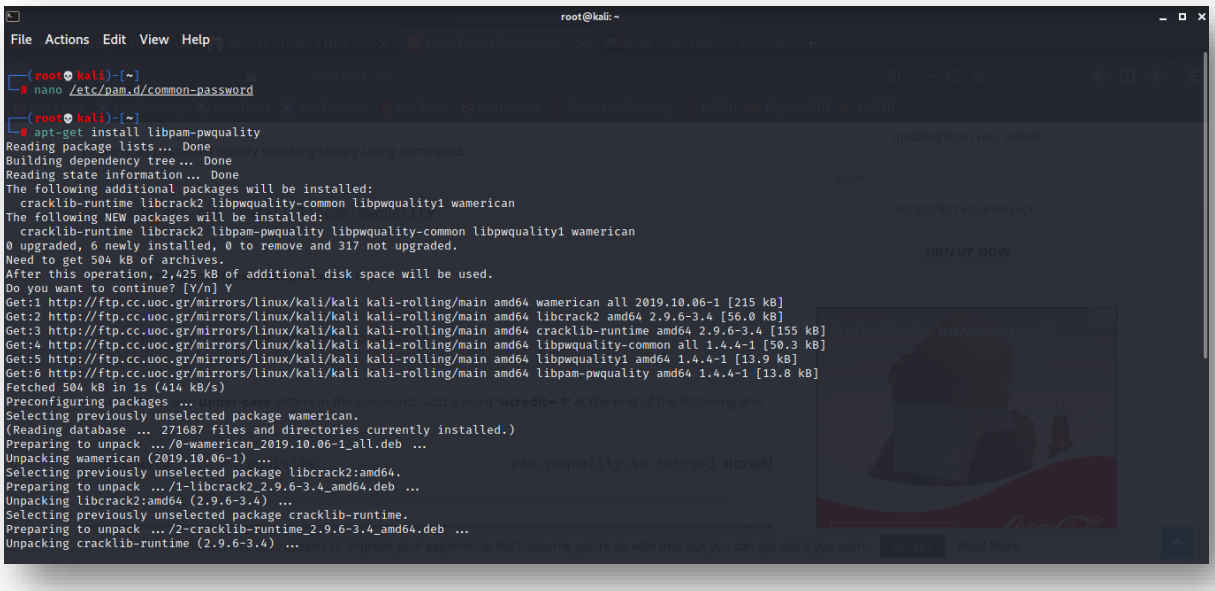
Στόχος της επίθεσης αυτής είναι:

- Η κλοπή προσωπικών πληροφοριών, κωδικών πρόσβασης που μπορούν να δώσουν πρόσβαση σε διαδικτυακούς λογαριασμούς και άλλους πόρους.
- Συλλογή διαπιστευτηρίων (credentials), δηλαδή όνομα χρήστη και συνθηματικά, με σκοπό την πώληση σε τρίτα μη εξουσιοδοτημένα πρόσωπα.
- Παρουσίαση ως ο ίδιος ο χρήστης για αποστολή δεδομένων "ψαρέματος" (phising) ή διάδοση πλαστού περιεχομένου, με σκοπό την άντληση πληροφοριών.
- Ανακατεύθυνση ιστοσελίδων σε ιστότοπους με κακόβουλο λογισμικό[23].

3.2.2. Εργαστηριακό μέρος Επίθεσης Εξαντλητικής Αναζήτησης

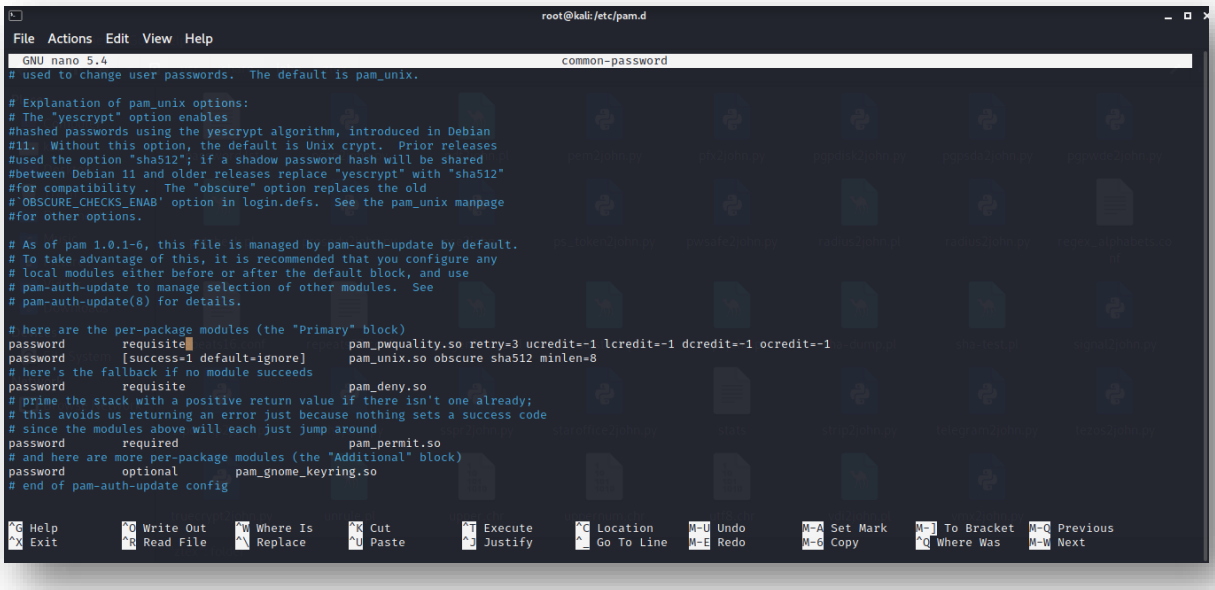
Για το εργαστηριακό μέρος της επίθεσης αυτής, χρειάστηκε να δουλέψουμε σε περιβάλλον Unix και συγκεκριμένα με τα KALI LINUX. Δημιουργήσαμε πολλούς χρήστες με διαφορετικά συνθηματικά τα οποία πήραμε από το αρχείο **password.lst** που βρίσκεται στην τοποθεσία usr/share/john/password.lst και υλοποιήσαμε την επίθεση με την βοήθεια του εργαλείου John Ripper.

Αρχικά, εγκαταστήσαμε την βιβλιοθήκη **libpam-pwquality** με τον παρακάτω τρόπο:



Εικόνα 13: Εγκατάσταση βιβλιοθήκης libpam-pwquality

Έπειτα, επεξεργαστήκαμε το αρχείο **common password**, που βρίσκεται στην τοποθεσία /etc/pam.d [24].

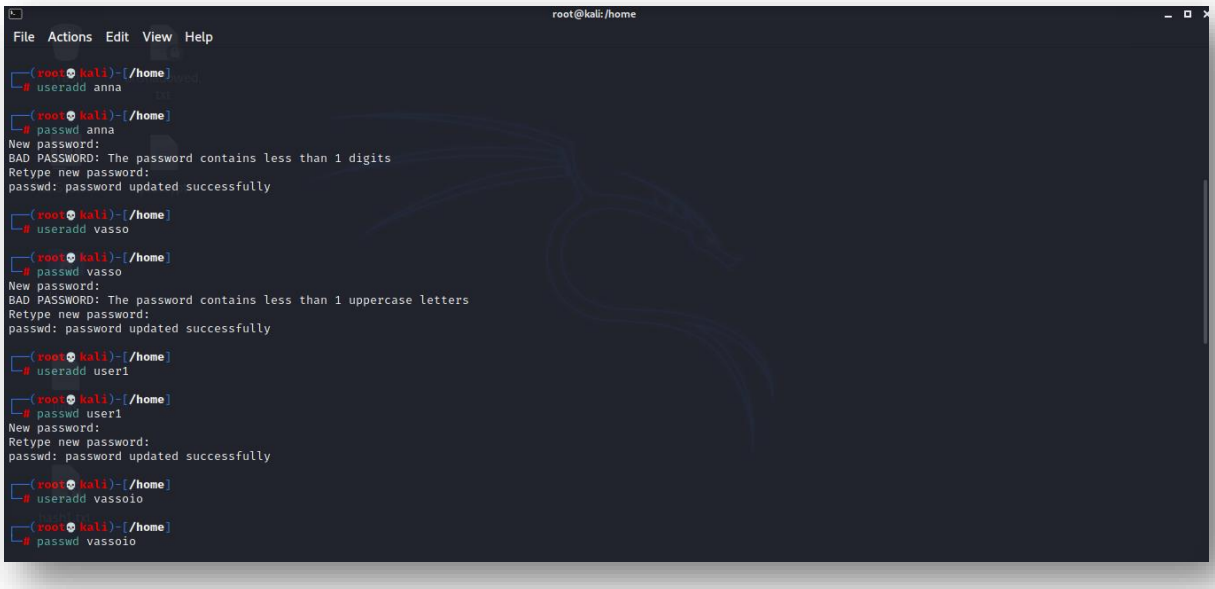


Εικόνα 14: Common Password

Σε αυτό το αρχείο βλέπουμε τον αλγόριθμό κατακερματισμού και την πολιτική κωδικών πρόσβασης που ακολουθεί το σύστημα. Η συγκεκριμένη πολιτική μας λέει ότι:

- **Retry = 3:** Ο χρήστης μπορεί ως 3 φορές να γράψει λάθος τον κωδικό πρόσβασης του.
- **Ucredit = -1:** Πρέπει να υπάρχει οπωσδήποτε ένας κεφαλαίος χαρακτήρας (uppercase).
- **Lcredit = -1:** Πρέπει να υπάρχει οπωσδήποτε ένας πεζός χαρακτήρας (lowercase).
- **Dcredit = -1:** Πρέπει να υπάρχει οπωσδήποτε ένα ψηφίο (digit).
- **Ocredit = -1:** Πρέπει να υπάρχει οπωσδήποτε ένας ειδικός χαρακτήρας.
- **Minlen = 8:** Το ελάχιστο μήκος του συνθηματικού πρέπει να είναι 8 χαρακτήρες.
- **Obscure sha512:** Αλγόριθμος κατακερματισμού SHA-512.

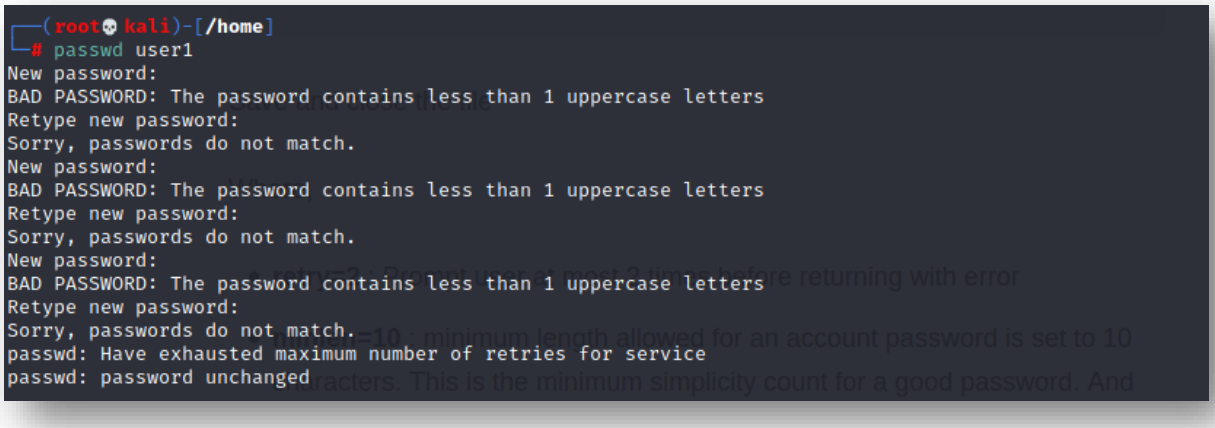
Στην συνέχεια κάναμε κάποιες δοκιμές για να ελέγξουμε αν εφαρμόζεται σωστά παραπάνω πολιτική.



Εικόνα 15: Εφαρμογή πολιτικής κωδικών πρόσβασης (1)

Ο χρήστης anna έχει κωδικό πρόσβασης bluebird και βλέπουμε ότι το σύστημα τον ειδοποιεί πως το συνθηματικό του περιέχει λιγότερο από ένα ψηφίο. Ο χρήστης vasso έχει κωδικό πρόσβασης 1qw23e και πάλι όμως βλέπουμε ότι το σύστημα τον ειδοποιεί πως το συνθηματικό του περιέχει λιγότερο από ένα κεφαλαίο γράμμα. Ο χρήστης user1 έχει κωδικό

πρόσβασης Im#1fan! και βλέπουμε ότι το σύστημα δεν τον ειδοποιεί για κάτι καθώς πληροί όλες τις προδιαγραφές που απαιτεί η πολιτική του συστήματος.

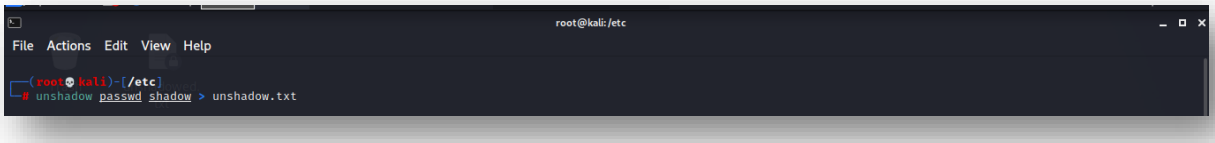


Εικόνα 16: Εφαρμογή πολιτικής κωδικών πρόσβασης (2)

Εδώ βλέπουμε ότι το σύστημα τηρεί την παράμετρο Retry = 3, αφού στην τρίτη προσπάθεια ανεπιτυχούς εισαγωγής κωδικού πρόσβασης ειδοποιεί τον χρήστη ότι έχει εξαντλήσει τις προσπάθειες και δεν του αλλάζει τον κωδικό.

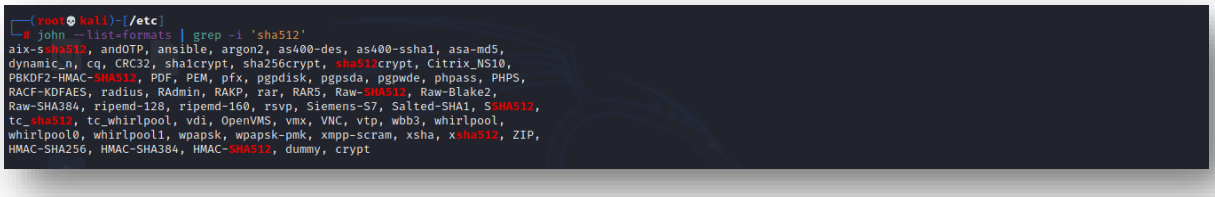
Στην συνέχεια ξεκινήσαμε την επίθεση ακολουθώντας τα παρακάτω βήματα. Αντλήσαμε κάποιους από τους κωδικούς πρόσβασης από το password list του John Ripper, έτσι ώστε να μπορέσουν να σπάσουν και η επίθεση να έχει ένα ποσοστό επιτυχίας [24][25].

Βήμα 1: Λαμβάνουμε ένα αντίγραφο του αρχείου κωδικού πρόσβασης (password file) με την βοήθεια της εντολής **unshadow** που μας παρέχει ο John Ripper.



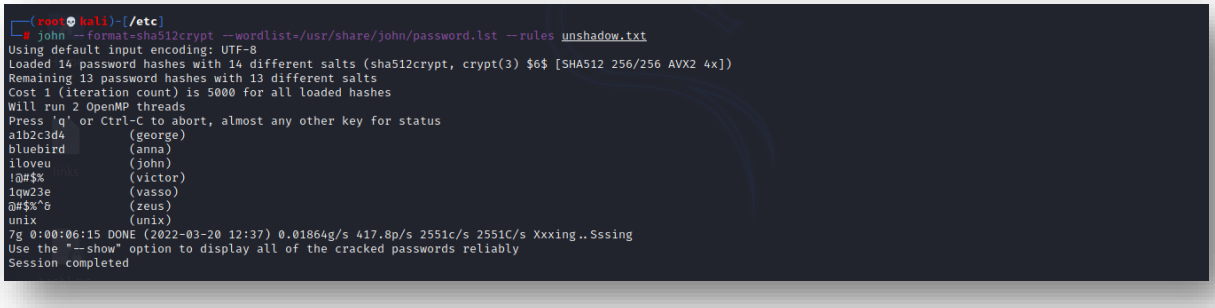
Εικόνα 17: Αντίγραφο αρχείου κωδικού πρόσβασης

Βήμα 2: Από την λίστα δυνατοτήτων ζητάμε να μας δείξει οτιδήποτε αφορά τον αλγόριθμο κατακερματισμού SHA-512.



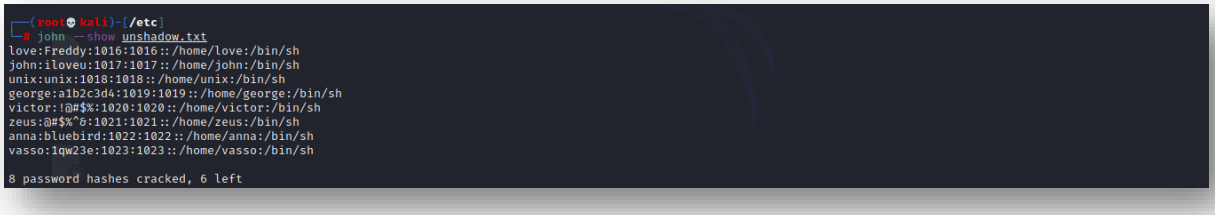
Εικόνα 18: Λίστα δυνατοτήτων και SHA-512

Βήμα 3: Εκτελούμε την επίθεση χρησιμοποιώντας μια λίστα λέξεων (όπως χρειάζεται μια εξαντλητική επίθεση) που βρίσκεται στην τοποθεσία /usr/share/john/password.lst , ενεργοποιούμε τους κανόνες παραποίησης λέξεων και προσπαθούμε να σπάσουμε τους κατακερματισμούς (που είναι της μορφής του SHA-512) των κωδικών πρόσβασης στο αρχείο unshadow.txt.



Εικόνα 19: Εκτέλεση επίθεσης Brute Force

Βήμα 4: Ανοίγουμε το αρχείο unshadow και βλέπουμε τους σπασμένους κωδικούς πρόσβασης. Το αρχείο αυτό μας ενημερώνει ότι σπάσανε οι 8 από τους 14 κωδικούς και αυτό έπρεπε να γινεί, αφού αυτοί οι 8 κωδικοί υπήρχαν στο password.lst και στους παρακάτω χρήστες.



Εικόνα 20: Αποτελέσματα επίθεσης Brute Force

Συμπερασματικά, αν ο εισβολέας μπορέσει και αποκτήσει πρόσβαση στο αρχείο που είναι γραμμένη η πολιτική των συνθηματικών, μπορεί να μειώσει τις λίστες λέξεων που είχε ως "εργαλείο" σε υπερβολικό βαθμό, καθώς θα γνωρίζει τι μορφή συνθηματικών επιτρέπει το σύστημα και σε συνδυασμό με μια δυνατή υπολογιστική ισχύς ή και μέτρια, μπορεί να σπάσει σε μικρό χρονικό διάστημα τους κατακερματισμούς των κωδικών πρόσβασης και να αποκτήσει πρόσβαση στο σύστημα και σε ό,τι άλλο του επιτρέπεται.

3.3. Επίθεση Λεξικού (Dictionary Attack)

Μια από τις γνωστές μεθόδους παραβίασης κωδικού πρόσβασης αποτελεί η επίθεση λεξικού ή dictionary attack. Η μέθοδος αποτελεί είδος εξαντλητικής επίθεσης (brute force) και μπορεί να χρησιμοποιηθεί και για την παραβίαση δικτύων αλλά και για οποιοδήποτε άλλο υπολογιστικό σύστημα που προστατεύεται από κωδικό πρόσβασης. Η διαδικασία είναι αρκετά απλή και αποτελεί την συστηματοποιημένη εισαγωγή όλων των εγγραφών ενός λεξικού ως πιθανό κωδικό πρόσβασης. Επιπλέον, η μέθοδος βρίσκει εφαρμογή στην διαδικασία αναζήτησης του κλειδιού αποκρυπτογράφησης για την αποκρυπτογράφηση κάποιου κρυπτογραφημένου αρχείου ή μηνύματος[20].

Οι επιθέσεις τέτοιου τύπου βρίσκουν εφαρμογή καθώς πολύ χρήστες υπολογιστικών συστημάτων, αλλά και επιχειρήσεις, τείνουν να χρησιμοποιούν απλές λέξεις ως συνθηματικά. Φυσικά, είναι προφανές πως για ένα σύστημα στο οποίο εφαρμόζεται η χρήση συνθηματικών πολλαπλών λέξεων (multi-word passwords) ή η χρήση συνθηματικών που απαρτίζονται από αλφαριθμητικά στοιχεία, τέτοιες επιθέσεις είναι συχνά ανεπιτυχείς. Η φύση των συνθηματικών πολλαπλών λέξεων και των αλφαριθμητικών συνθηματικών, η

οποία τα καθιστά σύνολα από πολύπλοκους συνδυασμούς συμβόλων, οδηγεί τόσο την μέθοδο λεξικού όσο και την μέθοδο της εξαντλητικής αναζήτησης (brute force) σε αδιέξοδο, θέτοντας τις ακίνδυνες για το σύστημα. Πρακτικά, η τυχαία επιλογή συνθηματικού περιορίζεται από το λεξικό προορισμένου περιεχομένου που χρησιμοποιείται, αμέσως εκμηδενίζει την αποτελεσματικότητα της μεθόδου καθώς το συνθηματικό στις παραπάνω περιπτώσεις είναι είτε απρόβλεπτο, είτε πολύ δύσκολα προβλέψιμο[20].

3.3.1. Θεωρητικό Μέρος Επίθεσης Λεξικού

Η επίθεση λεξικού λειτουργεί έχοντας προορίσει ένα λεξικό (dictionary) πιθανών συνθηματικών, ή κλειδιών στην περίπτωση αναζήτησης κλειδιού. Τα λεξικά συχνά προέρχονται από παρελθοντικές παραβιάσεις ασφαλείας, οι οποίες οδήγησαν και στην απόκτηση διαφόρων συνθηματικών ή κλειδιών[**Error! Reference source not found.**]. Βασικός "πυλώνας" της επίθεσης αυτής, είναι η πεποίθηση πως οι διάφοροι χρήστες τείνουν να βασίζονται στις ίδιες λέξεις ή εύρος λέξεων για την δημιουργία ενός συνθηματικού π.χ. "1234", "123abc", "1q2w3e4r5t6y7u8i9o0p".

Τα λεξικά αυτά συνήθως περιέχουν προβλέψιμα μοτίβα, τα οποία βασίζονται συνήθως σε προσωπικές πληροφορίες των χρηστών, όπως η χώρα διαμονής, η χώρα καταγωγής, προσωπικά δεδομένα (φωτογραφικό υλικό, προσωπικοί ιστότοποι/blogs). Το μέγεθος των λεξικών δεν ξεπερνά το μέγεθος των παραγόμενων συνθηματικών που προκύπτουν από την επίθεση εξαντλητικής αναζήτησης (brute force), αλλά μπορούν να γίνουν εξίσου μεγάλα. Η εφαρμογή των μεμονωμένων εγγραφών του λεξικού γίνεται σε συνδυασμό με άλλα προγράμματα, συχνά προγράμματα της προσέγγισης brute force. Η διαδικασία διαφοροποιείται ανάλογα με το αν η υπηρεσία πρόσβασης είναι διαδικτυακή ή τοπική. Αν η υπηρεσία είναι διαδικτυακή, τότε ο εισβολέας πρέπει να μεριμνήσει για τον αριθμό επιτρεπόμενων εισαγωγών και να προσπαθήσει να εισβάλλει όσο το δυνατών συντομότερα. Πολλοί επίδοξοι εισβολείς μπορούν φυσικά να απενεργοποιήσουν τον παραπάνω περιορισμό, επιτρέποντας τους περισσότερο χρόνο ανενόχλητης δράσης. Επιπλέον επιτάχυνση προσδίδει η ταξινόμηση του λεξικού ως προς την προτεραιότητα, η οποία παράγεται από μια διαδικασία μέγιστης πιθανοφάνειας. Οι παραπάνω περιορισμοί περιορίζονται σ' ένα τοπικό σύστημα, όπου ο εισβολέας ανησυχεί κυρίως για τον περιορισμό

εισαγωγών. Φυσικά μια επίθεση σε τοπικό σύστημα απαιτεί την πρόσβαση στο αρχείο αποθήκευσης συνθηματικών του συστήματος, αν αυτή η προϋπόθεση δεν ικανοποιηθεί, τότε δεν μπορεί να υπάρξει επίθεση.

Οι επιθέσεις λεξικών βασίζονται στην παραγωγή όλων των δυνατών αντιμεταθέσεων μεταξύ των εγγραφών του. Η επίθεση αυτή, παραλληλίζεται συχνά με την διαδικασία διαδικτυακού ψαρέματος (phishing), καθώς οι εισβολείς «ποντάρουν» στην ευάλωτη φύση των απλών χρηστών, οι οποίοι είναι πολύ πιθανόν να έχουν χρησιμοποιήσει κάποιο προβλέψιμο συνθηματικό. Τα βασικά βήματα της ροής εκτέλεσης της άνωθεν διαδικασίας είναι τα εξής[19]:

1) Ορισμός της πολιτικής συνθηματικών της εκάστοτε εφαρμογής/υπηρεσίας/συστήματος.
    a) Ορισμός του μέγιστου και του ελάχιστου μήκους συνθηματικού
    b) Ορισμός της μορφοποίησης (format) των επιτρεπόμενων συνθηματικών , δηλαδή η υποχρεωτική ύπαρξη συγκεκριμένων χαρακτήρων. Κυρίως ενδιαφέρον έχει η γνώση για το αν περιέχονται εγγραφές του λεξικού εντός της μορφοποίησης.
    c) Ορισμός της πολιτικής κλειδώματος σε περίπτωση αναγνώρισης προσπάθειας μη-εξουσιοδοτημένης πρόσβασης.
2) Επιλογή των αντίστοιχων λεξικών
    a) Δίνεται βάση στις προτιμώμενες γλώσσες εισαγωγής του χρήστη.
    b) Επιλογή των λεξικών βάση των υποστηριζόμενων γλωσσών.
3) Επιλογή των ονομάτων χρηστών προς επίθεση.
    a) Απόκτηση ονόματος χρήστη μέσω επίθεσης στο δίκτυο, στο αρχείο συστήματος ή με την χρήση ερωτήσεων - απαντήσεων στο σύστημα (querying).
4) Η εκμετάλλευση των δεδομένων για την διεξαγωγή της επίθεσης.
    a) Χρήση όλων των εγγραφών του λεξικού, συμπεριλαμβανομένων των πιθανών ορθογραφικά λανθασμένων.
    b) Χρήση συνήθων αντιμεταθέσεων σε συνδυασμό και με τις ορθογραφικά λανθασμένες εγγραφές.

Επιπλέον προαπαιτήσεις για να διεξαχθεί επιτυχώς η παραπάνω ροή εκτέλεσης είναι[19]:

- Το σύστημα να χρησιμοποιεί μέθοδο ταυτοποίησης ενός παράγοντα
- Το σύστημα να μην έχει επιβολή χρήσης ασφαλούς συνθηματικού, δηλαδή την αναγκαία υψηλή ποιότητα συνθηματικού (συνήθως απεικονίζονται με τα χρώματα κόκκινο, πορτοκαλί και πράσινο οι αντίστοιχες βαθμίδες αδύναμο, ισχυρό, πολύ ισχυρό).
- Το σύστημα να μην περιέχει υλοποίηση περιορισμού εισαγωγών καθώς και αναγνώρισης ύποπτων εισαγωγών (robot checker).

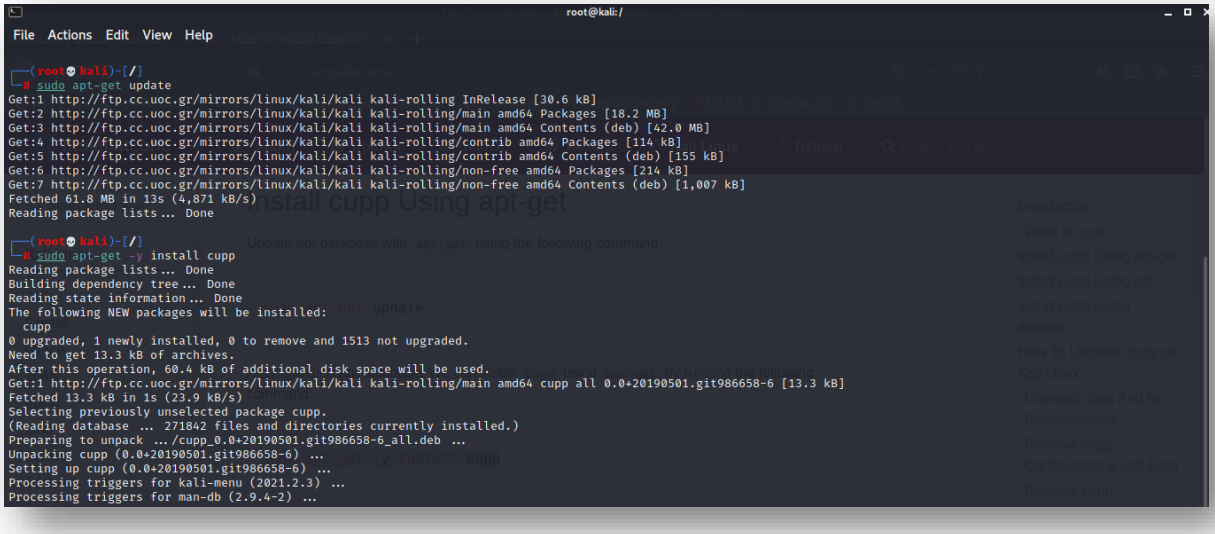
Για την εκτέλεση μιας τέτοιας επίθεσης, ο χρήστης απαιτείται να κατέχει υπολογιστικό σύστημα με επαρκείς πόρους (π.χ. CPU, RAM και HDD storage). Πρέπει να μπορεί να ορίσει την δομή του λεξικού, και να υπάρχει εγκατεστημένο κάποιο εργαλείο επίθεσης κωδικών, ή κάποιο custom script, το οποίο αξιοποιεί αποδοτικά το λεξικό[19].

Συμπερασματικά, η βασική διαφορά ανάμεσα στην επίθεση λεξικού και της ωμής βίας είναι το μέγεθος των πιθανών συνδυασμών που θα χρησιμοποιηθούν, έτσι ώστε να είναι επιτυχής η πρόσβαση στο υπολογιστικό σύστημα του θύματος.

3.3.2. Εργαστηριακό Μέρος Επίθεσης Λεξικού

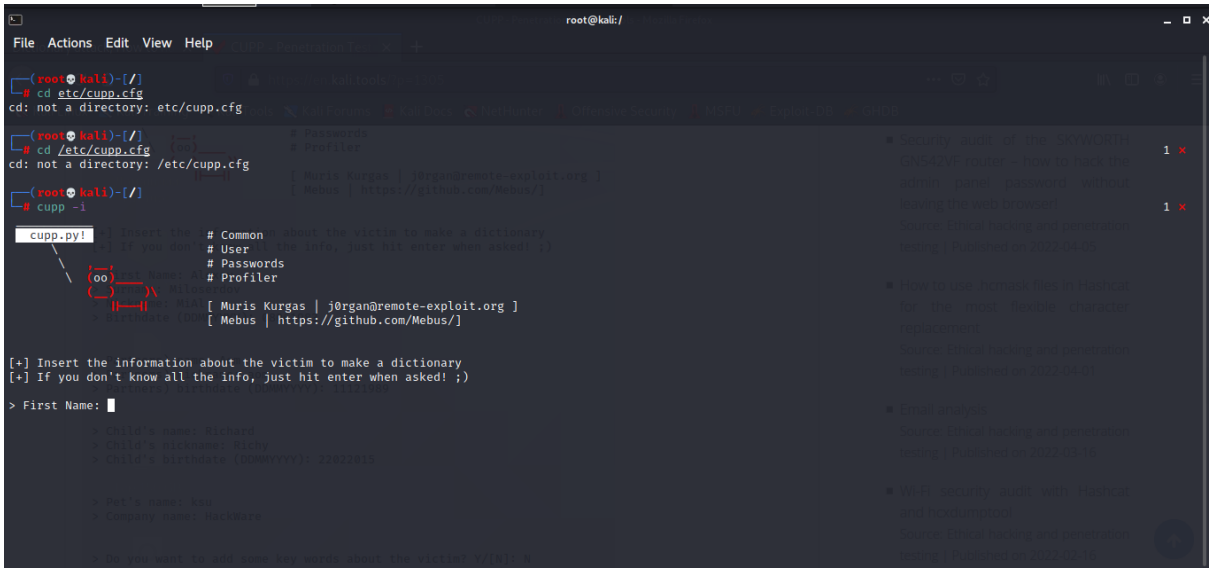
Για το εργαστηριακό μέρος της επίθεσης αυτής, χρειάστηκε να δουλέψουμε σε περιβάλλον Unix και συγκεκριμένα με τα KALI LINUX. Από τους ήδη υπάρχοντες χρήστες αλλάξαμε κάποια συνθηματικά και βάλαμε αυτά που δημιουργήθηκαν από το εργαλείο Common User Password Profiler (CUPP) , με το οποίο δημιουργήσαμε το λεξικό που χρησιμοποιήσαμε στην επίθεση. Η επίθεση πραγματοποιήθηκε πάλι με τον John Ripper.

Αρχικά, εγκαταστήσαμε το εργαλείο CUPP έτσι ώστε να δημιουργήσουμε ένα λεξικό, που έχει δημιουργήσει λέξεις βάσει των πληροφοριών που έχουμε δώσει[28].



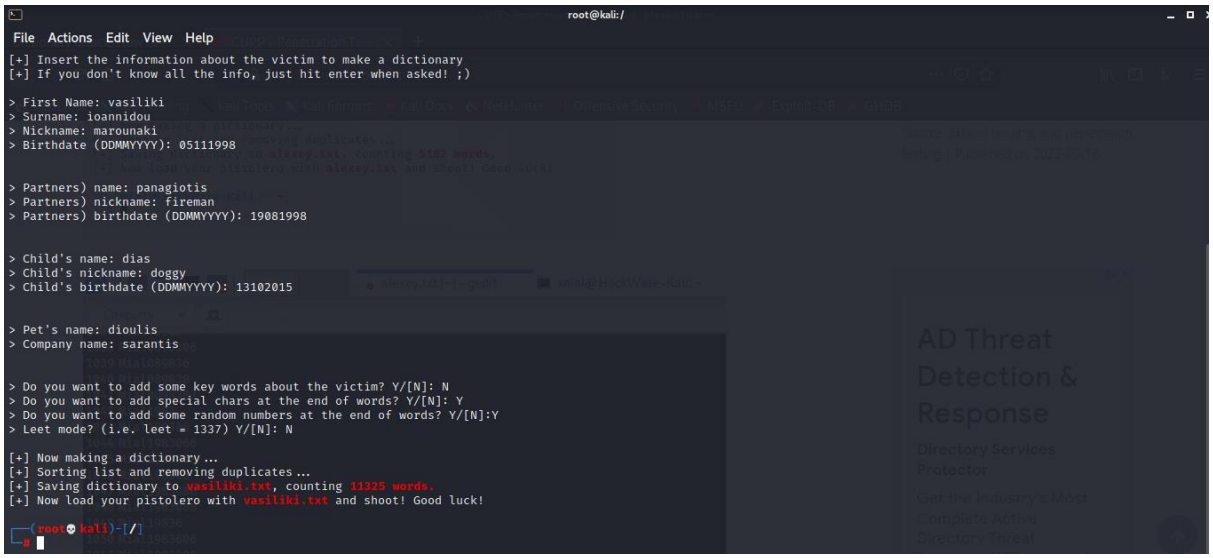
Εικόνα 21: Εγκατάσταση πακέτου CUPP

Μετά, ξεκινήσαμε το CUPP, έτσι ώστε να μπορέσουμε να δημιουργήσουμε το προφίλ του θύματος και να δημιουργηθεί το λεξικό.



Εικόνα 22: Έναρξη CUPP

Στην συνέχεια, δημιουργήσαμε το προφίλ του θύματος.

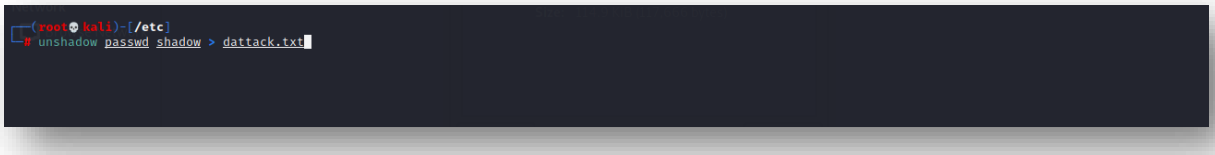


Εικόνα 23: Δημιουργία Προφίλ Θύματος

Παρατηρούμε ότι μόλις ολοκληρώσουμε το προφίλ του θύματος δημιουργείται το λεξικό αναζήτησης που θα χρησιμοποιήσουμε στην επίθεση με όνομα vasiliki.txt.

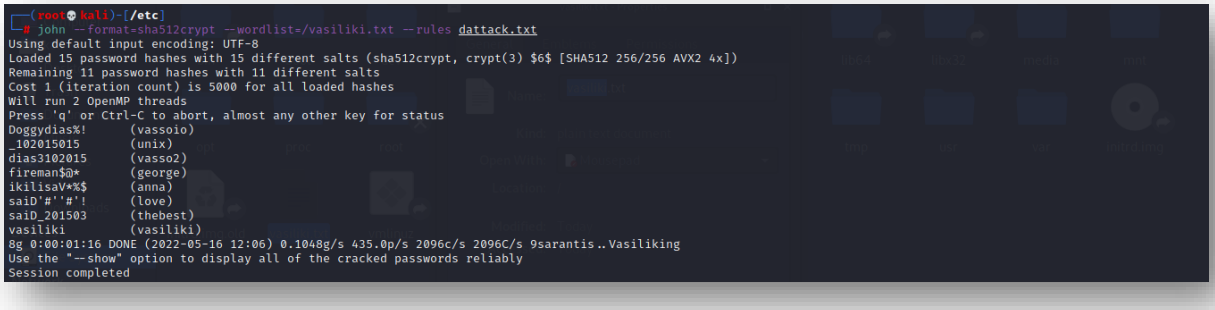
Ξεκινάμε την επίθεση όπως και την εξαντλητική επίθεση μόνο που αλλάζουν κάποιοι παράμετροι όταν εκτελούμε την επίθεση. Κι επιπλέον έχουμε αλλάξει κάποιους από τους ήδη υπάρχοντες κωδικούς πρόσβασης και έχουμε βάλει συνθηματικά που δημιουργήθηκαν και υπάρχουν στο vasiliki.txt.

Βήμα 1: Λαμβάνουμε ένα αντίγραφο του αρχείου κωδικού πρόσβασης (password file) με την βοήθεια της εντολής **unshadow** που μας παρέχει ο John Ripper.



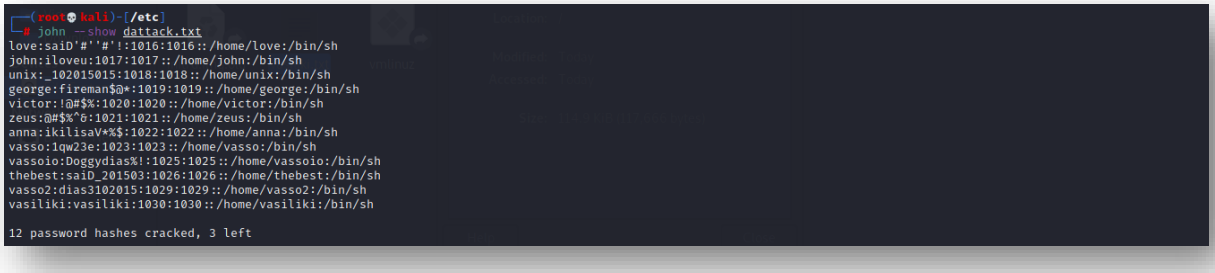
Εικόνα 24: Αντίγραφο αρχείου κωδικού πρόσβασης

Βήμα 2: Εκτελούμε την επίθεση χρησιμοποιώντας το λεξικό vasiliki.txt (όπως απαιτείται για αυτήν την επίθεση) που παράχθηκε με την βοήθεια του εργαλείου CUPP, ενεργοποιούμε τους κανόνες παραποίησης λέξεων και προσπαθούμε να σπάσουμε τους κατακερματισμούς (που είναι της μορφής του SHA-512) των κωδικών πρόσβασης στο αρχείο dattack.txt.



Εικόνα 25: Εκτέλεση Επίθεσης Dictionary Attack

Βήμα 3: Ανοίγουμε το αρχείο dattack και βλέπουμε τους σπασμένους κωδικούς πρόσβασης. Βλέπουμε ότι το αρχείο μάς ενημερώνει ότι έχουν μείνει 3 άσπαστοι κωδικοί κι αυτό γιατί δεν υπήρχαν στο λεξικό που χρησιμοποιήσαμε για την επίθεση με αποτέλεσμα να μην μπορέσει να τους σπάσει.



Εικόνα 26: Αποτελέσματα Επίθεσης Dictionary Attack

Συμπερασματικά, όπως και με την εξαντλητική επίθεση αν ο εισβολέας διαθέτει κάποιες πληροφορίες για το θύμα και σε συνδυασμό με μια μέτρια ως υψηλή υπολογιστική ισχύ μπορεί να κάνει μια τέτοια επίθεση να είναι επιτυχής αν και χρειάζεται αρκετό χρόνο για να σπάσει τους κατακερματισμένους κωδικούς , καθώς τα λεξικά είναι μεγάλου μεγέθους και μπορεί να περιέχουν χιλιάδες λέξεις.

3.4. Επίθεση Καταγραφής

Σε αυτό το υποκεφάλαιο θα εξετάσουμε την επίθεση καταγραφής η οποία βασίζεται στο τι πληκτρολογεί ο χρήστης μας και πως πραγματοποιείται μια τέτοια επίθεση όταν το θύμα συμπληρώνει το πεδίο κωδικού πρόσβασης μιας web εφαρμογής ή κάποιου συστήματος.

3.4.1. Θεωρητικό Μέρος Επίθεσης Καταγραφής

Ως Επιθέσεις Καταγραφής ή Keyloggers αναφερόμαστε σε διεργασίες που κύριος στόχος τους είναι να καταγράφουν την δραστηριότητα του πληκτρολογίου ενός χρήστη. Οι διεργασίες αυτές χρησιμοποιούνται για την μελέτη της αλληλεπίδρασης ανθρώπου-υπολογιστή, αλλά πιο συχνά χρησιμοποιούνται κακόβουλα με στόχο την απόσπαση ευαίσθητων πληροφοριών. Στην περίπτωση κακόβουλης χρήσης, η καταγραφή εκτελείται κρυφά από τον χρήστη είτε μέσω υλικού είτε μέσω λογισμικού, ενώ σπάνια βασίζεται και στην ακουστική ανάλυση.

Παρά την κατηγοριοποίηση τους ως κακόβουλα λογισμικά, τα Keyloggers δεν είναι κατά κανόνα παράνομα. Όπως αναφέραμε χρησιμοποιούνται για την εξαγωγή πληροφοριών, τόσο για την αλληλεπίδραση ανθρώπου-υπολογιστή όσο και για την διαδικασία του debug-ging προγραμμάτων. Σε κάθε περίπτωση όταν η χρήση των keyloggers γίνεται στην συσκευή του χρήστη, τότε αυτή καθίσταται απολύτως νόμιμη. Ωστόσο η ανησυχία για τα Keylogger είναι ότι από πίσω μπορούν να κρύβονται κακόβουλοι χρήστες και πως κατά την εμφύτευση ενός Keylogger σε ένα υπολογιστή ο χρήστης δεν γνωρίζει ότι έχουν παραβιάσει τον υπολογιστή του. Ανάλογα με το είδος του Keylogger, ο κακόβουλος χρήστης μπορεί:

- Να κλέψει κωδικούς πρόσβασης
- Να τραβήξει περιοδικά στιγμιότυπα οθόνης
- Να καταγράψει τις ιστοσελίδες που επισκέπτεται ο ιδιοκτήτης της συσκευής
- Να συγκεντρώσει μηνύματα ηλεκτρονικού ταχυδρομείου και προσωπικές συνομιλίες

- Ευαίσθητες οικονομικές πληροφορίες (όπως αριθμούς πιστωτικών καρτών, κωδικούς PIN και τραπεζικούς λογαριασμούς)

Στην συνέχεια η διεργασία του Keylogger μπορεί να στείλει όλα αυτά τα δεδομένα μέσω του δικτύου σε έναν απομακρυσμένο υπολογιστή ή σε κάποιον διαδικτυακό εξυπηρετητή. Με την αποστολή των πληροφοριών ο κακόβουλος χρήστης έχει πετύχει τον στόχο του.

Όπως έχουμε ήδη αναφέρει, τα Keyloggers μπορούν να αποτελούνται είτε από υλικό είτε λογισμικό. Οι Keyloggers υλικού αποτελούν συσκευές οι οποίες ενσωματώνονται εσωτερικά σε υπολογιστικά συστήματα και συγκεντρώνουν τα σήματα υλικού που ανταλλάσσονται μεταξύ του συστήματος και των περιφερειακών συσκευών. Από την άλλη, οι Keyloggers λογισμικού χρησιμοποιούν τις ροές δεδομένων (stdout, stdin) για να καταγράψουν τις πληροφορίες και να τις μεταδώσουν. Επιπρόσθετα, αποτελούν τους πιο διαδεδομένους λόγω της ευκολίας κατασκευής τους αλλά και εμφύτευσης τους στις διάφορες συσκευές. Στην επόμενη ενότητα θα μελετήσουμε πρακτικά την υλοποίηση ενός Keylogger.

3.4.2. Εργαστηριακό Μέρος Επίθεσης Καταγραφής

Σε αυτό το εργαστηριακό κομμάτι θα συνεχίσουμε την εργασία πάνω στα KALI LINUX χρησιμοποιώντας την γλώσσα προγραμματισμού Python. Για την υλοποίηση ενός Keylogger με την χρήση της Python θα χρειαστούμε τα modules keyboard, smtplib, threading και datetime. Ειδικότερα από τα modules threading και datetime χρησιμοποιούμε μόνο τις μεθόδους Timer και datetime αντίστοιχα. Η χρησιμότητα των παραπάνω πακέτων φαίνεται παρακάτω:

- **Keyboard**: Το module αυτό επιτρέπει στον χρήστη να ελέγξει πλήρως το πληκτρολόγιο του καθώς μέσω αυτού μπορεί να χρησιμοποιήσει τα global events του πληκτρολογίου, να καταχωρήσει hotkeys, να προσομοιώσει events πληκτρολογίου και πολλά ακόμα.
- **Smtplib**: Μέσω του module smtplib ένας χρήστης μπορεί να συστήσει το πρωτόκολλο ανταλλαγής μηνυμάτων στον κώδικα του. Η χρήση του module δημιουργεί μια συνεδρία πελάτη smtp στον κώδικα η οποία μπορεί να χρησιμοποιηθεί για την αποστολή μηνυμάτων email σε οποιαδήποτε μηχανή του διαδικτύου.

- **Threading.Timer:** Με την χρήση αυτού του module ο χρήστης μπορεί να κατασκευάσει πολυνηματικές διεπαφές. Συγκεκριμένα, η κλάση Timer που βρίσκεται εντός του πακέτου threading αναπαριστά μια ενέργεια που θα εκτελεστεί μόνο μετά από συγκεκριμένο χρονικό πλαίσιο. Εφόσον αποτελεί υποκλάση της κλάσης Thread, τα αντικείμενα τύπου Timer δημιουργούν νήματα.
- **Datetime**: Το πακέτο datetime παρέχει στους χρήστες κλάσεις για την διαχείριση ημερομηνιών και χρόνων.

Επομένως όπως αντιλαμβανόμαστε η υλοποίηση των Keyloggers αποτελείται από ένα Timer αντικείμενο το οποίο ορίζει τα χρονικά διαστήματα κατά τα οποία το πρόγραμμα εξάγει αναφορές καταγραφής. Το χρονικό διάστημα ορίζεται από τον χρήστη ως σταθερά στην αρχή του κώδικα. Κατά την διάρκεια αυτού, το πρόγραμμα καταγράφει τα events του πληκτρολογίου και τα αποθηκεύει σε μια προσωρινή συμβολοσειρά. Με το πέρας του χρονικού διαστήματος, η συμβολοσειρά αποθηκεύεται στο αρχείο καταγραφής ή αποστέλλεται μέσω email σε διεύθυνση που έχει οριστεί ως σταθερά στην αρχή του κώδικα. Παρακάτω βλέπουμε στιγμιότυπα εκτέλεσης του κώδικα.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo python keylogger.py
[sudo] password for kali:
2022-07-18 09:58:36.611531 - Started keylogger
[+] Saved keylog--2022-07-18-095836_2022-07-18-095936.txt
[+] Saved keylog--2022-07-18-095936_2022-07-18-100036.txt
^CTraceback (most recent call last):
  File "/home/kali/Desktop/keylogger.py", line 104, in <module>
    keylogger.start()
  File "/home/kali/Desktop/keylogger.py", line 96, in start
    keyboard.wait()
  File "/usr/local/lib/python3.9/dist-packages/keyboard/__init__.py", line 886, in wait
    _time.sleep(1e6)
KeyboardInterrupt
```

Εικόνα 27: Εκτέλεση Επίθεσης Keyloggers

Για να σταματήσουμε τον κώδικα χρησιμοποιήσαμε το σήμα Ctrl+C για την διακοπή της διεργασίας. Αν αφήσουμε την διεργασία να εκτελείται, τότε θα εξάγει αναφορές ανά χρονικά διαστήματα, ανάλογα του πλαισίου που έχουμε ορίσει στην αρχή του κώδικα. Να σημειωθεί πως στην περίπτωση που ο χρήστης δεν πληκτρολογεί, το πρόγραμμα του Keylogger δεν αποθηκεύει καμία τιμή. Τα αποτελέσματα μπορούν να φανούν με την ολοκλήρωση του κώδικα:

Εικόνα 28: Αποτελέσματα Επίθεσης Keyloggers

Τα πλήκτρα enter και backspace κωδικοποιούνται έτσι ώστε να εμφανίζονται στο αρχείο καταγραφής ως [ENTER] και [BACKSPACE] αντίστοιχα. Επιπλέον ο χρήστης μπορεί να τροποποιήσει την διαδικασία καταγραφής για την αποθήκευση περισσότερης πληροφορίας, όπως η ώρα που ο χρήστης πάτησε το enter.

Παρατηρούμε πως η κατασκευή ενός Keylogger προϋποθέτει καλή γνώση της γλώσσας Python και ορθή αντίληψη για την συνεργία των περιφερειακών συσκευών και του λειτουργικού συστήματος. Μπορούμε να διακρίνουμε πως η υλοποίηση μας αποτελεί μια ασφαλή υλοποίηση Keylogger, καθώς δεν κάνει κρυφή την παρουσία της στο σύστημα. Η μετατροπή της διαδικασίας σε κακόβουλο λογισμικό προϋποθέτει την απόκρυψη των μηνυμάτων, ενώ η περαιτέρω επέκταση είναι η μετατροπή του προγράμματος σε δαίμονα, δηλαδή πρόγραμμα που εκτελείται αυτόματα κατά την εκκίνηση.

3.5. Rainbow Tables

Μια επίθεση με την χρήση Rainbow Tables αποτελεί μια διαδικασία απόσπασης συνθηματικών η οποία βασίζεται σε μια ειδική δομή δεδομένων και την κρυπτανάλυση. Η δομή δεδομένων ονομάζεται Rainbow Table και χρησιμοποιείται για την απόσπαση κατακερματισμένων συνθηματικών σε μια βάση δεδομένων. Η επίθεση εκμεταλλεύεται την

διαδικασία επαλήθευσης χρήστη με την χρήση κατακερματισμού (hashing). Τα Rainbow Tables αποτελούν προϋπολογισμένους πίνακες οι οποίοι περιέχουν τις κατακερματισμένες τιμές που αντιστοιχούν σε plain text χαρακτήρες οι οποίοι χρησιμοποιούνται κατά την επαλήθευση χρήστη. Η πρόσβαση σε λίστες από κατακερματισμούς (hashes) κωδικών πρόσβασης, που μπορεί να προέρχονται από παλιότερες παραβιάσεις, επιτρέπει στους δράστες να υποκλέψουν επιτυχώς κωδικούς πρόσβασης μέσω της μεθόδου του Rainbow Tables.

Σήμερα η ισχύς των επιθέσεων με βάση τα Rainbow Tables έχει ελαττωθεί σημαντικά λόγω της τεχνικής γνωστής ως “Salting”. Η τεχνική αυτή προσθέτει μια τυχαία τιμή σε κάθε κατακερματισμένο συνθηματικό, δημιουργώντας έτσι ένα νέο hash μειώνοντας σημαντικά τον κίνδυνο των επιθέσεων βασισμένων σε Rainbow Tables. Παρά την ασφάλεια που προσφέρει, η τεχνική salting δεν είναι τόσο διαδεδομένη, καθιστώντας τα περισσότερα συστήματα ευπαθή σε επιθέσεις Rainbow Table.

Για να καταλάβουμε καλύτερα τον κίνδυνο που παρουσιάζουν οι επιθέσεις με βάση τα Rainbow Tables θα χρειαστεί να μελετήσουμε τον τρόπο με τον οποίο αυτές υλοποιούνται. Αρχικά οι επιτήδειοι εισβολείς πρέπει να αποκτήσουν πρόσβαση σε τιμές hash που έχουν διαρρεύσει λόγω παλιών παραβιάσεων βάσεων δεδομένων. Επιπλέον τρόποι που μπορούν να αποσπαστούν οι πληροφορίες των hash είναι στις περιπτώσεις που η βάση δεδομένων δεν προστατεύεται επαρκώς, αν ο εισβολέας έχει πρόσβαση στον ενεργό κατάλογο της βάσης ή αν καταφέρει να τα αποσπάσει αποκτώντας πρόσβαση στην βάση μέσω τακτικών Phishing σε λογαριασμούς διαχειριστών. Με την απόκτηση των τιμών hash, οι Rainbow Tables μπορούν να χρησιμοποιηθούν με στόχο την αποκρυπτογράφηση των κλειδιών. Η μέθοδος ακολουθεί την προσέγγιση της κρυπτανάλυσης ώστε η αποκρυπτογράφηση να εκτελεστεί γρήγορα και αποδοτικά. Σε αντίθεση με τις μεθόδους ωμής δύναμης, οι οποίες σε κάθε βήμα δοκιμάζουν νέο συνθηματικό παράγοντας νέο hash με κάθε επανάληψη, η μέθοδος των Rainbow Tables προϋπολογίζει τις απαραίτητες τιμές hash εκ των προτέρων για κάθε διαθέσιμο συνθηματικό. Επομένως η διαδικασία χωρίζεται σε δύο βήματα:

1. **Η δημιουργία του πίνακα:** Η διαδικασία χρησιμοποιεί ένα πιθανό συνθηματικό και το κατακερματίζει επανειλημμένα ώστε να παραγάγει μια αλυσίδα πιθανών τιμών hash.

2. **Σύγκριση με την βάση δεδομένων:** Οι τιμές hash του Rainbow Table συγκρίνονται με τα hash της βάσης δεδομένων. Αν το hash βρεθεί εντός της βάσης τότε κινούμαστε αντιστρόφως στην αλυσίδα πιθανών τιμών ώστε να εντοπίσουμε τον κωδικού που οδήγησε στο συγκεκριμένο hash. Δεν υπάρχει επικοινωνία με τον εξυπηρετητή, επομένως δεν τίθεται κίνδυνος ανίχνευσης, αρκεί πάντα ο εισβολέας να έχει αποσπάσει με κάποιο τρόπο τα hashes της βάσης δεδομένων.

Κεφάλαιο 4: Προτεινόμενες λύσεις για κωδικούς πρόσβασης

Έχοντας μελετήσει τις διάφορες ευπάθειες των συνθηματικών μπορούμε να αποφανθούμε πως αυτές δημιουργούν ένα αρκετά δυσχαιρές πεδίο για τους σύγχρονους και τους μέλλοντες χρήστες, τόσο του διαδικτύου αλλά και των τοπικών συστημάτων. Ως εκ τούτου, στο κεφάλαιο αυτό θα προβούμε σε μελέτη των διάφορων προτεινόμενων αντίμετρων, μέσω των οποίων θωρακίζονται τα συστήματα και οι λογαριασμοί των χρηστών, και θα αξιολογήσουμε τόσο την εφαρμογή τους όσο και την πολυπλοκότητα τους.

Ως αντίμετρα ορίζουμε τις τακτικές αντιμετώπισης διαφόρων ευπαθειών των συνθηματικών από την πλευρά του συστήματος. Η τακτικές αυτές ενσωματώνονται στα συστήματα με στόχο την ενίσχυση της ασφάλειας δίχως απαραίτητα να αντικαθιστούν τα συνθηματικά. Τα αντίμετρα ποικίλλουν, από επιπλέον διαπιστευτήρια, ξεχωριστή συσκευή για την επικύρωση, έως συστήματα λογισμικού και απλές γραπτές κατευθυντήριες γραμμές για την ορθή χρήση των συνθηματικών.

4.1. Πολυπλοκότητα Κωδικών

Αρχικά θα αναφέρουμε το πιο προφανές αντίμετρο, το οποίο συγκαταλέγεται στις κατευθυντήριες γραμμές για την επιλογή συνθηματικού πρόσβασης. Αυτό φυσικά αποτελεί η πολυπλοκότητα του κωδικού πρόσβασης, δηλαδή ο συνδυασμός των χαρακτήρων που απαρτίζουν την λέξη. Η αύξηση της πολυπλοκότητας μειώνει την πιθανότητα ορθής πρόβλεψης του συνθηματικού από αλγόριθμους πρόβλεψης ή παραγωγής αντιμεταθέσεων. Βέβαια, όπως έχουμε αναφέρει επανειλημμένως, αυτή η βαθμίδα προστασίας μπορεί να παρακαμφθεί ευκολότατα με την χρήση επιθέσεων όπως είναι οι επιθέσεις καταγραφής (keyloggers), οι επιθέσεις που βασίζονται σε κάποιο έγκυρο σύνολο δεδομένων για τα συνθηματικά (dictionary attacks) και οι πίνακες ουράνιου τόξου (rainbow tables). Για αυτό τον λόγο ξεκινάμε την απαρίθμηση των αποδοτικών αντίμετρων από το αμέσως επόμενο.

4.2. Προσέγγιση Πολλαπλών Συνθηματικών και Συστήματα Διαχείρισης Συνθηματικών

Αρχικό αντίμετρο που μπορούμε να προτείνουμε είναι η δημιουργία πολλών διαφορετικών συνθηματικών ανά λογαριασμό ή εφαρμογή χρήστη, ώστε να είναι λιγότερο εύκολη η εισβολή σε πολλαπλούς λογαριασμούς από γνώση ενός συνθηματικού από αυτούς. Όπως και πάλι γνωρίζουμε όμως, τα πολλαπλά συνθηματικά μπορούν να αποτελέσουν ταυτόχρονα αντίμετρο αλλά και ευπάθεια καθώς οι χρήστες δεν μπορούν να είναι συνεπείς στην παρακολούθηση και την επίβλεψη της ακεραιότητας για όλα αυτά, και μάλιστα ανά τακτά χρονικά διαστήματα. Τέτοιες περιπτώσεις επιτρέπουν στους επίδοξους εισβολείς να χρησιμοποιήσουν μεθόδους όπως οι επιθέσεις ωμής βίας και λεξικών, ώστε να εντοπίσουν τυχόν ξεχασμένα ή παραμελημένα συνθηματικά. Σε αυτή την ευπάθεια ένα από αντίμετρα ασφαλείας αποτελούν τα συστήματα διαχείρισης συνθηματικών.

Τα συστήματα διαχείρισης συνθηματικών αποσκοπούν στην συλλογική αποθήκευση και προστασία διαφόρων συνθηματικών μέσω ενός μοναδικού συνθηματικού-αφέντη (Master Password). Επιπλέον, στην λειτουργικότητα τέτοιων συστημάτων συγκαταλέγονται υπηρεσίες όπως αξιολόγηση ποιότητας συνθηματικού, γεννήτορες συνθηματικών με βάση δημοφιλή πρότυπα και γενικούς ελέγχους ασφαλείας πάνω από όλα τα συνθηματικά. Επιπλέον παρέχουν υπηρεσίες όπως ανανέωση συνθηματικών που αλλάζει ο χρήστης, αυτόματη συμπλήρωση φορμών ιστοσελίδων αλλά και αυτόματη σύνδεση σε λογαριασμούς. Όπως όμως γίνεται προφανές, ακόμα και σε αυτά τα συστήματα έχουμε συνθηματικά που μπορούν να υποκλαπούν. Η ύπαρξη ενός συνθηματικού-αφέντη δεν αποτρέπει την εισβολή στον συγκεκριμένο λογαριασμό διαχειριστή συνθηματικών, θέτοντας σε κίνδυνο συλλογικά τα συνθηματικού του χρήστη.

Τα παραπάνω συστήματα, αν και επιβοηθητικά, δεν μπορούν να αποτελέσουν επαρκή βαθμίδα προστασίας για κάποιο σύστημα, καθώς συνεχίζεται ο κίνδυνος πιθανής διαρροής πληροφορίας για τον χρήστη, αν και αυτός δεν το γνωρίζει, π.χ. επίθεση στον εξυπηρετητή/βάση δεδομένων τους συστήματος διαχείρισης. Συνεπώς, οι σχεδιαστές συστημάτων πρέπει να μεριμνήσουν για την ασφάλεια των συνθηματικών του χρήστη και κατ' επέκταση του ίδιου, μέσω μεθοδολογιών από την πλευρά τόσο του συστήματος όσο και της σχεδίασης του υλικού. Από την πλευρά του συστήματος, οι σχεδιαστές θα πρέπει να

προβούν σε σχεδιασμό τέτοιο ώστε η επικύρωση χρήστη να γίνεται με τον όσο δυνατότερα πληρέστερο τρόπο.

4.3. Αντίμετρα από την πλευρά του Συστήματος

Πέρα από τα αντίμετρα που βασίζονται στην εκπαίδευση και συμπεριφορά του χρήστη, υπάρχουν και τα αντίμετρα που εφαρμόζονται από την πλευρά του συστήματος. Οι σχεδιαστές ενός συστήματος δεν μπορούν να έχουν την απαίτηση από τους χρήστες του συστήματος να κατέχουν πλήρη γνώση πάνω στα θέματα της ασφάλειας συστημάτων, έτσι ώστε να μεριμνούν οι ίδιοι για την ασφάλεια τόσο της δικής τους, όσο και του συστήματος ως σύνολο. Αντιθέτως, οι σχεδιαστές καλούνται να βρουν οι ίδιοι λύσεις για να απομακρύνουν όσο το δυνατόν περισσότερο τον χρήστη από την τεχνική πλευρά του συστήματος, και να τον περιορίσουν στην εμπορική πλευρά του.

Βασισμένη σε αυτές τις προϋποθέσεις, σε αυτή την υποενότητα θα αναφερθούμε στις τακτικές που μπορούν να εφαρμόσουν οι σχεδιαστές συστημάτων με στόχο την ακεραιότητα του συνόλου του συστήματος.

4.4. Προστασία Αρχείου Κωδικών και Ενιαία Επαλήθευση Ταυτότητας

Όπως είναι γνωστό, για να γίνει είσοδος ενός χρήστη σ’ ένα σύστημα χρειάζεται να γίνει διασταύρωση του δοθέντος συνθηματικού με την αντίστοιχη εγγραφή στο αρχείο συνθηματικών πρόσβασης των χρηστών. Ο κίνδυνος έπεται όταν ο επίδοξος εισβολέας καταφέρει να αποσπάσει ή απλά να αποκτήσει ένα αντίγραφο του αρχείου συνθηματικών (συνήθως από κάποια επίθεση στην βάση δεδομένων). Χρησιμοποιώντας το αρχείο, αργά ή γρήγορα, ο δράστης θα μπορέσει να διεισδύσει στο σύστημα, αφού έχει αποσπάσει τα συνθηματικά όλων των χρηστών του συστήματος.

Λαμβάνοντας υπόψη όλα τα παραπάνω, είναι επόμενο πως το αρχείο κωδικών πρέπει να αποτελεί ένα από τα πιο προστατευμένα αρχεία εντός οποιουδήποτε συστήματος. Τα αρχεία αυτά συνήθως βρίσκονται αποθηκευμένα σε μηχανές του συστήματος στις οποίες υπάρχει η ελάχιστη και πιο αυστηρή πρόσβαση, ενώ πολλές φορές δεν είναι άμεσα ανιχνεύσιμες όταν μιλάμε για κάποιο κατανεμημένο σύστημα. Αν η απόκρυψη της μηχανής που περιέχει

το αρχείο κωδικών δεν είναι εφικτή, τότε πρέπει να επιλεχθεί κάποια πολύ ισχυρή συνάρτηση κατακερματισμού, έτσι ώστε οι κωδικοί να βρίσκονται αποθηκευμένοι ως κλειδιά της συνάρτησης. Επιπλέον, πρέπει να χρησιμοποιείται η πιο ισχυρή και ασφαλής συνάρτηση κατακερματισμού, έτσι ώστε ο χρόνος διείσδυσης να είναι αρκετά μεγάλος.

Επιπλέον τρόπος για την προστασία του συστήματος είναι η επαλήθευση της ταυτότητας να γίνεται από έναν εξυπηρετητή, ο οποίος χρησιμοποιεί ένα συνθηματικό το οποίο επιτρέπει πρόσβαση του χρήστη σε πολλές υπηρεσίες που παρέχονται από το σύνολο του συστήματος. Πρακτικά, μειώνει τον αριθμό των συστημάτων τα οποία κατέχουν ευαίσθητες πληροφορίες, αλλά και τον αριθμό των συνθηματικών που πρέπει να θυμάται κάποιος χρήστης. Βέβαια επιβάλλεται η χρήση επιπλέον μηχανισμών προστασίας και πρωτοκόλλων για την μεταφορά των δεδομένων από κόμβο σε κόμβο μέσα στο σύστημα. Σε αυτή την προσέγγιση, οι χρήστες δεν χρησιμοποιούν κωδικούς αλλά φράσεις-κλειδιά. Οι φράσεις-κλειδιά τείνουν να είναι περίπλοκες και σύνθετες, έτσι ώστε να είναι και πιο ασφαλείς. Φυσικά, η διασύνδεση πολλών υποσυστημάτων με μια συνθηματική φράση-κλειδί αποτελεί αρκετά ριψοκίνδυνη τακτική, η οποία προϋποθέτει πως ο εξυπηρετητής ή η ομάδα κόμβων στην οποία διατηρείται η διαδικασία επαλήθευσης πρέπει να βρίσκεται υπό την ύψιστη ασφάλεια. Τέλος, οι χρήστες του συστήματος οφείλουν να υπακούν στις κατευθυντήριες γραμμές που θα θέσει ο πάροχος της υπηρεσίας/συστήματος η οποίες θα τους προστατέψουν από τις περιπτώσεις Phishing ή keylogging.

4.5. Αποκλεισμός πρόσβασης και «Γνώση του Εχθρού»

Μέχρι τώρα έχουμε αναφερθεί στο σύστημα ως παθητική οντότητα, η οποία λαμβάνει απλά αμυντικές ενέργειες και προληπτικά μέτρα. Αυτό όμως δεν είναι απαραίτητο, καθώς το σύστημα μπορεί και το ίδιο να γίνει επιθετικό ενάντια στους εισβολείς. Συγκεκριμένα, οι σχεδιαστές του συστήματος έχουν τα εφόδια και την ικανότητα να γνωρίζουν για την ύπαρξη των μεθόδων επίθεσης. Αυτή η γνώση επιτρέπει στους σχεδιαστές να «γνωρίζουν τον εχθρό», και κατ’ επέκταση να ελέγχουν την ανθεκτικότητα των συστημάτων τους, καθώς γνωρίζουν που να ψάξουν για ευπάθειες.

Όπως φυσικά γίνεται αντιληπτό, ο τομέας της ασφάλειας συστημάτων, αλλά και οι επαγγελματίες του χώρου αποτελούν από τις πιο δυσεύρετες και καλοπληρωμένες θέσεις εργασίας, οι οποίες όμως απαιτούν πολλές ώρες και αφοσίωση. Το προσωπικό στο οποίο ανατίθενται αυτές οι αρμοδιότητες αποτελεί πρέπει να είναι έμπιστο και εξειδικευμένο, καθώς έρχεται σε επαφή με το αρχείο συνθηματικών. Η αλληλεπίδραση της ομάδας ασφαλείας με το αρχείο συνθηματικών πρέπει να είναι η ελάχιστη δυνατή, έτσι ώστε να πληρούνται οι απαιτήσεις εχεμύθειας, ακολουθώντας τις ηθικές και νομικές δεσμεύσεις του οργανισμού / παρόχου. Οι ομάδα πρέπει να έχει γνώση μόνο για την κατάσταση ενός λογαριασμού και όχι για το συνθηματικό που τον προστατεύει. Όπως είναι αναμενόμενο, σε αυτό τον τομέα, έχουμε ξανά την περίπτωση του ανθρώπινου παράγοντα, όπου πιθανή διαρροή πληροφοριών δεν είναι απίθανη, με στόχο την αμαύρωση του κύρους του οργανισμού.

Το αποτέλεσμα της επιτυχούς λειτουργίας μιας ομάδας ασφαλείας οδηγεί στην υλοποίηση λειτουργιών που μπορούν, όχι μόνο να προστατέψουν έγκαιρα ένα σύστημα, αλλά να αποκλείσουν και να στιγματίσουν κάποιον κακόβουλο χρήστη. Στην περίπτωση ανίχνευσης κάποιου υπόπτου χρήστη ή περίπτωσης χρήσης του συστήματος, το σύστημα μπαίνει σε κατάσταση πανικού και εκτελεί ρουτίνες για την αναγνώριση του πιθανού κινδύνου. Μια από τις πιο γνωστές περιπτώσεις ανίχνευσης και αποκλεισμού κινδύνου είναι ο αποκλεισμός ύποπτου χρήστη. Αυτή η τακτική αποσκοπεί στην άμυνα του συστήματος ενάντια σε επιθέσεις που έχουν βάση την επαναληπτική εισαγωγή πιθανών συνθηματικών, ώστε να διεισδύσουν στο σύστημα. Το σύστημα μετρά τις αποτυχίες εισόδου για κάποιον δεδομένο χρήστη που έχει τεθεί υπό παρακολούθηση, όπου αν ο αριθμός αυτών ξεπεράσει ένα προκαθορισμένο όριο, τότε ο χρήστης ορίζεται ως κακόβουλος ή απλά επικίνδυνος, και το σύστημα τον κλειδώνει εκτός αυτού. Η τακτική αυτή επιτρέπει στους διαχειριστές του συστήματος και στην ομάδα ασφαλείας να αντιδράσουν σε πιθανή επίθεση έγκαιρα, και να αποτρέψουν τον επιτιθέμενο από το να αποκτήσει παραπάνω πρόσβαση στο σύστημα.

4.6. Ανίχνευση και Περιορισμός Επιθέσεων Ασφαλείας

Όταν ένα συνθηματικό έχει υποκλαπεί τότε μπορούμε να παρατηρήσουμε διαφορά στην συμπεριφορά του χρήστη στον οποίο αντιστοιχεί. Τα αρχεία καταγραφής του λογαριασμού χρήστη δηλώνουν την χρήση του λογαριασμού και την τοποθεσία του χρήστη, τα οποία αν διαφέρουν από προηγούμενες εγγραφές, τότε ίσως αποτελούν αποδείξεις για μια μη-πιστοποιημένη χρήση του λογαριασμού του χρήστη. Η ανίχνευση βασίζεται σε τρία χαρακτηριστικά, τα μοτίβα χρήσης, την ύποπτη δραστηριότητα και η γήρανση συνθηματικών πρόσβασης.

Αρκετοί λογαριασμοί χρηστών εμφανίζουν συχνά μοτίβα χρήσης, π.χ. ένας λογαριασμός στο YouTube μπορεί να βλέπει συγκεκριμένο περιεχόμενο σε διαφορετικές χρονικές περιόδους μέσα στην ημέρα, τον μήνα ή τον χρόνο. Αυτά τα μοτίβα συγκεντρώνονται από τους αλγόριθμους του συστήματος και σχηματίζουν το προφίλ του χρήστη. Επιπλέον, σε αυτά τα μοτίβα υπάρχουν πληροφορίες για την τοποθεσία από την οποία συνδέθηκε ο χρήστης ή με τις πιο συχνές τοποθεσίες από τις οποίες συνδέεται. Στις πιο σύγχρονες εφαρμογές, το σύστημα δύναται να αναγνωρίζει ακόμα και τις συσκευές από τις οποίες συνδέεται κάποιος χρήστης. Φυσικά, η συλλογή των παραπάνω δεδομένων υπάγεται στις πολιτικές διαχείρισης ευαίσθητων δεδομένων, οι οποίες πρέπει να γνωστοποιούνται στον χρήστη κατά την δημιουργία του λογαριασμού του. Επομένως, αν παρατηρηθεί κάποια αλλαγή στα παραπάνω μοτίβα χρήσεις, το σύστημα πρέπει να θέσει τον λογαριασμό υπό παρακολούθηση.

Η αλλαγή στα μοτίβα χρήσης ενός λογαριασμού χρήστη μπορεί πολλές φορές να υποδηλώνει την ύπαρξη κάποιου προβλήματος, όπως η εισβολή ενός κακόβουλου χρήστη στο σύστημα, ενώ και άλλες φορές μπορεί να αποτελεί απλή συγκυρία, π.χ. η σύνδεση στους λογαριασμούς του χρήστη μετά την αγορά νέας συσκευής ή την μετακόμιση σε άλλη γεωγραφική τοποθεσία. Πολλές φορές, ακόμα και η ταυτόχρονη πρόσβαση από διαφορετικές τοποθεσίες μπορεί να αποτελέσει σήμα κινδύνου για το σύστημα π.χ. στην περίπτωση που δύο φίλοι χρησιμοποιούν τον ίδιο λογαριασμό στο Netflix. Βέβαια, η προηγούμενη περίπτωση χρήσεις ομοιάζει πολύ με την περίπτωση ο δεύτερος χρήστης να είναι κάποιος κακόβουλος, μη-επικυρωμένος χρήστης, συνεπώς το σύστημα οφείλει να είναι ικανό να εκτελεί ορθό διαχωρισμό των περιπτώσεων.

Ο διαχωρισμός των περιπτώσεων δεν είναι πάντα εφικτός, καθώς υπάρχουν πολλοί αστάθμητοι παράγοντες για να εξάγουμε κάποιο καθοριστικό συμπέρασμα. Πολλές φορές μπορεί ο λόγος για τον οποίο κάποια σύνδεση εκληφθεί ως ύποπτη να οφείλεται σε τεχνικές συγκυρίες, όπως η διεύθυνση IP την σύνδεσης να βρίσκεται σε κάποια λίστα αυτόματης απόρριψης ή ύποπτων διευθύνσεων. Για να καλυφθούν όλες οι πιθανές περιπτώσεις, τα συστήματα λαμβάνουν δρακόντεια μέτρα οφείλουν να είναι αμετάκλητα και καχύποπτα για κάθε χρήστη που δεν είναι αυτός που αρχικά εμφανίζει, και σε συνέχεια να τον υποβάλλουν σε πολλαπλούς ελέγχους διαπιστευτηρίων, έτσι ώστε να διασφαλιστεί η ακεραιότητα όλων των χρηστών του συστήματος.

Το θέμα είναι το πώς ορίζουμε την ύποπτη συμπεριφορά του χρήστη. Εν ολίγης, η ερώτηση αφορά για το ποιες περιπτώσεις χρήσης καθορίζουν τον δρουν χρήστη ως εν δυνάμει κίνδυνο. Όπως μπορούμε να δούμε και από το άμεσο μας περιβάλλον, οι περισσότερες επιθέσεις σε πανεπιστημιακούς λογαριασμούς απευθύνονται σε ανυποψίαστους φοιτητές, οι οποίοι θα πέσουν θύματα της τακτικής που ονομάσαμε ψάρεμα (phishing). Οι επιθέσεις αυτές μπορούν να πάρουν την μορφή μηνυμάτων ηλεκτρονικού ταχυδρομείου, τα οποία φαίνονται ελκυστικά καθώς μπορεί να προσποιούνται πως προέρχονται από κάποια επιτροπή υποτροφιών ή κάποιον καθηγητή. Το σύστημα πρέπει να ορίσει κάποιο κριτήριο για την αναγνώριση και απώθηση τέτοιων κινδύνων. Εκμεταλλευόμενοι την φύση των μηνυμάτων αυτών ως ανεπιθύμητη αλληλογραφία (SPAM), το σύστημα μπορεί να παρακολουθεί τέτοια μηνύματα, ειδικά αυτά που απευθύνονται σε παραλήπτες όπως είναι το allstudents@uniwa.gr ή το all@uniwa.gr, οι οποίες διευθύνσεις συμπεριλαμβάνουν μεγάλο αριθμό παραληπτών.

Μια κλασσική μέθοδος που προτείνεται ώστε να περιοριστούν πιθανές παραβιάσεις είναι η γήρανση των κωδικών πρόσβασης. Κοινή τακτική, ιδιαίτερα σε τραπεζικά συστήματα καταναλωτών, η γήρανση των κωδικών πρόσβασης υποχρεώνει τους χρήστες να αλλάζουν τα συνθηματικά εισόδου τους ανά προκαθορισμένα χρονικά διαστήματα, συνήθως ορίζεται το διάστημα 3 ή 6 μήνες. Το χρονικό διάστημα δεν είναι τυχαίο καθώς επιλέγεται με βάση τον χρόνο ανακάλυψης των συνθηματικών από τους εισβολείς. Βέβαια, η επιλογή αυτή έγινε με βάση την ανακάλυψη μέσω της διαδικασίας brute force. Με την εισαγωγή των Rainbow Tables στο παιχνίδι, το χρονικό διάστημα θα έπρεπε να οριστεί σε ώρες ή ακόμα και λεπτά.

Η τακτική της προστασίας με την χρήση συναρτήσεων κατακερματισμού αποτελεί καλύτερο αντίμετρο ενάντια στις επιθέσεις rainbow table, αλλά αυτό δεν σημαίνει πως η γήρανση θα εγκαταλειφθεί. Ο περιορισμένος χρόνος ζωής των συνθηματικών παραμένει αρκετά βοηθητικό αντίμετρο, το οποίο επιτρέπει την αντιμετώπιση της αρχικής περίπτωσης, όπου χρήστες που χρησιμοποιούν πολλούς λογαριασμούς και συνθηματικά τα εγκαταλείπουν, με αποτέλεσμα αυτά να μείνουν εκτεθειμένα σε επιθέσεις. Συμπεραίνουμε πως η διαδικασία της γήρανσης, αν και πλέον όχι αρκετά ισχυρή, συνεργάζεται άψογα με τα συστήματα διαχείρισης συνθηματικών και την εφαρμογή κατευθυντήριων γραμμών, ώστε να δημιουργήσει ένα ιδανικότερο βασικό περιβάλλον, πάνω στο οποίο θα εφαρμοστούν ακόμα περισσότερα αντίμετρα, με στόχο την προστασία των συνθηματικών.

4.7. Επαλήθευση Ταυτότητας Πολλών Παραγόντων (MFA) και Συνθηματικά μιας Χρήσης (OTPs)

Ένας από τους πιο διαδεδομένους τρόπους με τον οποίο μπορεί να προστεθεί επιπλέον ασφάλεια στο σύστημα για τον χρήστη είναι η χρήση του 2FA και γενικά των MFA. Η διαδικασία ορίζει πως για την σύνδεση ενός χρήστη στο σύστημα απαιτούνται παραπάνω από ένα στοιχείο επικύρωσης, όπου αν χρειάζονται μόνο δύο τότε αναφερόμαστε σε **2FA (Two-Factor Authentication)** ενώ για παραπάνω αναφερόμαστε σε **MFA (Multi-Factor Authentication)**. Η διαδικασία αποτελεί κεντρικό κομμάτι της πολιτικής διαχείρισης ταυτοποίησης και πρόσβασης των συστημάτων, έχοντας πλέον γίνει συχνή επιλογή είτε το σύστημα αποτελεί μια απλή πλατφόρμα κοινωνικής δικτύωσης, εφαρμογή ή ακόμα και κάποιο προσωπικό εικονικό δίκτυο (VPN). Ως δραστηριότητα, το σύστημα στο οποίο επιθυμεί να συνδεθεί ο χρήστης δεν απαιτεί μόνο ένα συνθηματικό, αλλά σε συνέχεια της υποβολής του βασικού συνθηματικού ζητά από τον χρήστη επιπλέον διαπιστευτήρια εισόδου ώστε να εξακριβώσει ότι πρόκειται για τον ίδιο χρήστη. Η διαδικασία της MFA καταφέρνει να μειώσει την πιθανότητα επιτυχούς επίθεσης.

Η πιο γνωστή διαδικασία MFA είναι η χρήση **OTP (One-Time Password)** μέσω της οποίας παράγεται από κάποιον γεννήτορα μια αριθμητική αλληλουχία τεσσάρων έως οκτώ συμβόλων, η οποία γνωστοποιείται στον χρήστη μέσω ηλεκτρονικού ταχυδρομείου,

μηνύματος κινητού τηλεφώνου ή και εφαρμογής κινητού τηλεφώνου. Με την διαδικασία των OTP παράγεται μια νέα αριθμητική αλληλουχία κάθε μερικά κβάντα χρόνου ή κάθε φορά που απαιτείται ταυτοποίηση του χρήστη. Η γεννήτρια βασίζεται σε μια μοναδική ριζική τιμή (seed value) η οποία ανατίθεται στον χρήστη κατά την εγγραφή του στην υπηρεσία και σε κάποια επιπλέον παράμετρο όπως π.χ. θα μπορούσε να είναι ένας μετρητής. Στην περίπτωση που η αλληλουχία αυτή είναι διαθέσιμη για περιορισμένη χρονική περίοδο τότε ονομάζεται TOTP (Time-based One-Time Password).

Η εφήμερη φύση των OTPs επιτρέπει την προσθήκη επιπλέον ασφάλειας στα συστήματα αλλά και στους χρήστες. Στα βασικά πλεονεκτήματα των OTPs συγκαταλέγονται τα παρακάτω:

→ Αντοχή στις επιθέσεις που επαναχρησιμοποιούν κλεμμένα συνθηματικά, καθώς απαιτείται επιπλέον πληροφορία που δεν δύναται να γνωρίζει ο εισβολέας.

→ Δυσκολία στην προσπάθεια ορισμού του OTP. Εφόσον αυτές οι αλληλουχίες παράγονται τελείως τυχαία, οι εισβολείς δεν είναι εύκολα να τα μαντέψουν. Συνήθως οι αλληλουχίες αποτελούνται από σύμβολα του δεκαδικού συστήματος, επομένως όλοι οι πιθανοί τυχαίοι συνδυασμοί με επανάληψη συμβόλου ορίζονται ως

$$n = r^k$$

Όπου ***n*** ο αριθμός των πιθανών αλληλουχιών, ***r*** η βάση του αριθμητικού συστήματος και ***k*** το μήκος της αλληλουχίας. Για παράδειγμα, για τον Google Authenticator, οι πιθανές αναδιατάξεις είναι 10^6 = 1.000.000. Ειδικά αν έχουν διάρκεια μερικών δευτερολέπτων, τότε ο υπολογισμός και η δοκιμή κάθε κωδικού είναι ανέφικτη. Με την υλοποίηση αντίστοιχου κώδικα σε Python καταλήξαμε πως ο χρόνος ολοκλήρωσης της παραγωγής όλων των αντιμεταθέσεων για συνθηματικά έξι ψηφίων είναι μεγαλύτερος των 60 λεπτών, κάτι που κάνει την εκτίμηση του OTP αδύνατη.

→ Μείωση κινδύνου κατά την διαρροή συνθηματικών, καθώς ακόμα και στην περίπτωση που ο χρήστης χρησιμοποιεί το ίδιο συνθηματικό για πολλούς διαφορετικούς λογαριασμούς, οι εισβολείς χρειάζονται και το παραγόμενο OTP.

→ Εύκολη ενσωμάτωση σε εφαρμογές και υπηρεσίες, καθώς πολλές φορές παρέχονται ως πρόσθετα.

→ Η τεχνολογία των OTPs δεν θα ήταν εφικτή δίχως την ύπαρξη των tokens. Τα tokens στην τεχνολογία των OTPs χωρίζονται σε δύο κατηγορίες, τα υλικού και λογισμικού. Ένα token υλικού αποτελεί μια συσκευή η οποία παράγει το OTP. Αυτά τα tokens μπορούν να ανήκουν σε μια από τις παρακάτω κατηγορίες:

→ Συνδεδεμένα tokens: Οι χρήστες συνδέουν τα tokens αυτά στο σύστημα ή την συσκευή στην οποία επιθυμούν την πρόσβαση.

→ Μη-συνδεδεμένα tokens: Η πιο διαδεδομένη μέθοδος για την υλοποίηση του MFA. Οι χρήστες δεν είναι υποχρεωμένοι να συνδέσουν το token σε κάποιο σύστημα, καθώς η συσκευή token τους παράγει OTPs, τα οποία θα εισάγουν οι χρήστες.

→ Ανέπαφα tokens: Σε αυτά τα token γίνεται εκπομπή των δεδομένων επικύρωσης στο σύστημα που επιθυμεί να εισέλθει ο χρήστης. Παράδειγμα αυτών είναι η σύνδεση σε συστήματα μέσω Bluetooth.

Από την άλλη πλευρά τα token λογισμικού αποτελούν εφαρμογές είτε στην ίδια, είτε σε άλλη συσκευή από την οποία προσπαθεί να συνδεθεί στο σύστημα ο χρήστης. Πολλές φορές τα token λογισμικού παίρνουν την μορφή εφαρμογών, μέσω της οποίας ο κωδικός επαλήθευσης στέλνεται με κάποιο SMS ή κάποια ειδοποίηση. Με την εισαγωγή του κωδικού επαλήθευσης, ο χρήστης μπορεί να εισέλθει στο σύστημα. Γενικά, είτε η αποστολή του κωδικού γίνει μέσω SMS ή ειδοποίησης η διαδικασία είναι ίδια, ο χρήστης στέλνει τα δεδομένα επαλήθευσης στο σύστημα, το σύστημα επικυρώνει την ύπαρξη του χρήστη και στην συνέχεια του παρέχει πρόσβαση σε αυτό. Όπως μπορούμε να παρατηρήσουμε, η λογική είναι παρόμοια με αυτή των κωδικών πρόσβασης, όμως η χρήση OTPs αποτελεί ασφαλέστερη μέθοδος, καθώς δεν βρίσκονται εκτεθειμένα ποτέ, εφόσον η πληροφορία δεν χρειάζεται να μεταδοθεί από το σύστημα στον χρήστη.

4.8. Επαλήθευση SMS

Μια από τις πιο απλές και διαδεδομένες μεθόδους επαλήθευσης είναι αυτή με την χρήστη SMS για την αποστολή του κωδικού επαλήθευσης. Παρά όμως την διαδεδομένη φύση της, η μέθοδος αυτή παρουσιάζει αρκετές ευπάθειες.

→ **Εναλλαγή κάρτας SIM:** Ο δράστης μπορεί να ανακατευθύνει τον κωδικό επαλήθευσης σε δικιά του SIM κάρτα, αποκτώντας πρόσβαση στον λογαριασμό του χρήστη.

→ **Υποκλοπή Λογαριασμού Παρόχου:** Πολλοί πάροχοι τηλεπικοινωνιακών υπηρεσιών παρέχουν στους χρήστες τους λογαριασμούς μέσω των οποίων μπορούν να ελέγχουν τα μηνύματα τους από κάποιο υπολογιστικό σύστημα. Εφόσον και αυτοί οι λογαριασμοί προστατεύονται με συνθηματικά, η εισβολή σε αυτούς δεν είναι διόλου απίθανη. Έτσι, οι δράστες μπορούν να εισβάλουν στον λογαριασμό του χρήστη και να δουν τους κωδικούς επαλήθευσης.

→ **Αντικατάσταση Συσκευής εντός Συγχρονισμού:** Με την αλλαγή συσκευής, οι χρήστες πρέπει να είναι προσεκτικοί έτσι ώστε να μην αφήσουν συγχρονισμένες παλιές συσκευές στους λογαριασμούς τους. Πολλές ιστοσελίδες διεξάγουν ελέγχους στοιχείων ανά τακτά χρονικά διαστήματα ώστε να εντοπίσουν τέτοια κενά ασφαλείας. Σε περίπτωση που ένας λογαριασμός έχει μείνει συνδεδεμένος με έναν αριθμό κινητής τηλεφωνίας που πλέον έχει αλλάξει ιδιοκτήτη, ή ακόμα και στην περίπτωση συσκευής, τότε το σύστημα πρέπει με κάποιον τρόπο να ανιχνεύσει το κενό, ειδάλλως τίθεται σε κίνδυνο η ψηφιακή ακεραιότητα των χρηστών του.

→ **Phishing:** Η γνωστή επίθεση που βασίζεται στις ευπάθειες του κοινωνικού μηχανισμού. Ο δράστης υποδύεται κάποιο κυβερνητικό υπάλληλο, φορέα παροχής υπηρεσιών ή ακόμα και συγγενικό πρόσωπο με στόχο την υποκλοπή στοιχεία (περισσότερα έχουν αναφερθεί σε προηγούμενη ενότητα). Στην περίπτωση αυτή, η υποκλοπή όχι μόνο των συνθηματικών, αλλά και των κωδικών επαλήθευσης δεν είναι καθόλου αδύνατη.

4.9. Επαλήθευση Ταυτότητας Πολλών Παραγόντων

Όπως είδαμε και εκτενώς στην προηγούμενη ενότητα, η επαλήθευση ταυτότητας πολλών παραγόντων αποτελεί μια από τις συνηθέστερες και πιο διαδεδομένες τακτικές ασφαλείας (αντίμετρα) για τα συστήματα που επιθυμούν να εισάγουν επιπλέον βαθμίδες ασφαλείας για την προστασία της ακεραιότητας τόσο των χρηστών τους, όσο και της υπόληψης τους ως

οργανισμοί. Η διαδικασία αυτή προσθέτει ένα επιπλέον επίπεδο σιγουριάς για το σύστημα, πως ο χρήστης που αιτείται την σύνδεση στον λογαριασμό είναι και ο ιδιοκτήτης του λογαριασμού. Στα βασικά πλεονεκτήματα, η μέθοδος μειώνει τον κίνδυνο υποκλοπής του λογαριασμού, την εκμετάλλευση ευαίσθητων δεδομένων και την κατ' επέκταση παραβίαση και υποκλοπή περισσότερων δεδομένων από την βάση δεδομένων του συστήματος.

Η επαλήθευση ταυτότητας πολλαπλών παραγόντων χωρίζει τους παράγοντες επαλήθευσης σε τρείς κατηγορίες:

→ Γνώσης: Αφορούν την επαλήθευση μέσο πληροφοριών που μόνο ο ιδιοκτήτης του λογαριασμού είναι σε θέση να γνωρίζει. Στην κατηγορία αυτή ανήκουν παράγοντες όπως PINs, συνθηματικά πρόσβασης και ερωτήσεις ασφαλείας. Οι παράγοντες αυτοί είναι οι λιγότερο ασφαλείς, καθώς είναι πολύ εύκολα να κοινοποιηθούν στους πιθανούς εισβολείς, μέσω επιθέσεων phishing ή επιθέσεων κοινωνικού μηχανισμού.

→ Κτήσης: Απαιτούν την κτήση κάποιου υλικού μέσου ώστε να γίνει η επαλήθευση του ιδιοκτήτη του λογαριασμού. Στην κατηγορία αυτή συγκαταλέγονται τα USB-keys, τα κινητά τηλέφωνα και οι τραπεζικές κάρτες. Τα αντικείμενα αυτά έχουν το κοινό γνώρισμα πως μπορούν να αποθηκεύσουν τοπικά στοιχεία ταυτοποίησης, και αποτελούν ασφαλέστερο παράγοντα σε σχέση με τους γνωσιακούς παράγοντες. Π.χ. όπως προαναφέραμε, η επαλήθευση μέσω SMS είναι ασφαλέστερη από τις ερωτήσεις ασφαλείας, αλλά εδώ παρατηρούμε τον κίνδυνο απώλειας της συσκευής ή ακόμα και κλοπής της. Επιθέσεις όπως οι man-in-the-middle καθιστούν τον παράγοντα αυτό ως μέσης ασφάλειας.

→ Εγγενή Χαρακτηριστικά: Βασίζονται σε χαρακτηριστικά με τα οποία έχει γεννηθεί ο χρήστης και τον προσωποποιούν αναμφίβολα. Σε αυτή την κατηγορία βρίσκονται τα βιομετρικά χαρακτηριστικά, τα οποία αποτελούν έναν από τους ισχυρότερους παράγοντες ταυτοποίησης, καθώς είναι μοναδικά ανά υποκείμενο χρήστη. Στα βιομετρικά χαρακτηριστικά συμπεριλαμβάνονται τα δακτυλικά αποτυπώματα, η αναγνώριση προσώπου, η αναγνώριση φωνής και οι παράγοντες συμπεριφοράς. Ίσως το μεγαλύτερο πλεονέκτημα αυτών των παραγόντων είναι πως δεν απαιτείται κάποιο υλικό ή και γενικότερα κάποια ιδιαίτερη προσπάθεια για την αποθήκευση τους,

καθιστώντας τους ως τους πιο ασφαλείς για την εφαρμογή της επαλήθευσης πολλών παραγόντων.

Η επαλήθευση ταυτότητας πολλών παραγόντων μπορεί να ενισχυθεί μέσω συγκεκριμένων μεθόδων που εξειδικεύουν την λειτουργικότητα του:

→ **Επαλήθευση βασισμένη στην Τοποθεσία:** Κατά αυτή την μέθοδο χρησιμοποιείται η διεύθυνση IP και η γεωγραφική τοποθεσία, για να επιτρέψει ή να απορρίψει της σύνδεσης σε κάποια εφαρμογή ή σύστημα. Οι πληροφορίες αυτές μπορούν να προσπελαστούν μέσω της διαδικασίας επαλήθευσης, επιπλέον της εισαγωγής PIN ή OTP, για τον περαιτέρω έλεγχο της ταυτότητας του χρήστη.

→ **Επαλήθευση βασισμένη στο Ρίσκο:** Επίσης γνωστή ως προσαρμοστική επαλήθευση, ορίζει δυναμικά τον παράγοντα επαλήθευσης ανάλογα με συνοδευτικές πληροφορίες όπως η γεωγραφική τοποθεσία του χρήστη, η συσκευή που χρησιμοποιείται για είσοδο και το δίκτυο μέσου του οποίου έχει πρόσβαση ο χρήστης. Αυτή η προσέγγιση θεωρείται η ιδανικότερη για μια ισορροπία μεταξύ απαιτήσεων ασφαλείας και εμπειρίας χρήστη.

→ **Επαλήθευση Δίχως Συνθηματικό:** Λαμβάνοντας υπόψη πως οι περισσότερες παραβιάσεις συστημάτων οφείλονται σε υποκλοπές συνθηματικών εισόδου, η μέθοδος αυτή συνδυάζει απλά παράγοντες υψηλής ακρίβειας όπως είναι το ανερχόμενο FIDO2.0 / WebAuth, και επιπλέον πληροφορίες όπως η γεωγραφική τοποθεσία, η υποκείμενη κατάσταση του χρήστη (Ρίσκο), η συμπεριφορά του χρήστη και η κατάσταση της συσκευής. Αυτή η μέθοδος δεν συγκαταλέγεται στα αντίμετρα, επομένως θα την επισκεφτούμε ξανά σε επόμενη ενότητα.

Η χρήση της επαλήθευσης ταυτότητας πολλών παραγόντων έχει δώσει μια λύση στον πρόβλημα των ευπαθειών των κωδικών πρόσβασης (συνθηματικών), καθιστώντας την κλοπή πληροφοριών αρκετά δύσκολη διαδικασία για τους δράστες. Ακόμα και στην περίπτωση που οι δράστες καταφέρουν να αποσπάσουν τις πληροφορίες ταυτοποίησης, αυτές τους είναι σχεδόν άχρηστες δίχως τους επιπλέον παράγοντες εισόδου. Βέβαια, ο χώρος των MFA δεν έχει λιμνάσει, καθώς οι μεθοδολογίες και τακτικές εξελίσσονται ακολουθώντας την ανάπτυξη της επιστήμης των υπολογιστών.

Η ανάπτυξη καινοτόμων μεθόδων έχει ως στόχο την μεγιστοποίηση της σχέσης μεταξύ ασφάλειας συστήματος και ευκολίας χρήσης από την πλευρά των χρηστών. Αυτός είναι και ο λόγος που οι μέθοδοι βιομετρικών και εισόδου δίχως συνθηματικά αποτελούν τις ποιο επικρατέστερες επιλογές. Επιπλέον άλματα στον χώρο της επαλήθευσης ταυτότητας προσφέρει ο χώρος της τεχνητής νοημοσύνης και της μηχανικής μάθησης, αλλά και το ανερχόμενο πρότυπο της FIDO. Χρησιμοποιώντας τα πρώτα, τα συστήματα μπορούν να αναγνωρίσουν εγκαίρως κατηγορίες αιτημάτων σύνδεσης ή μοτίβων συμπεριφοράς τα οποία χαρακτηρίζονται ως «φυσιολογικά». Με αυτή την προσέγγιση, η επαλήθευση γίνεται βάση αυτών των αποφάσεων, και ο χρήστης απλά απολαμβάνει την παρεχόμενη υπηρεσία. Στην δεύτερη περίπτωση, το πρότυπο της FIDO και συγκεκριμένα το πρότυπο FIDO2.0 αποσκοπεί στον προσανατολισμό των περισσότερων συστημάτων προς την εγκατάλειψη των συνθηματικών πρόσβασης, και της εγκαθίδρυσης μεθόδων πολλαπλών παραγόντων. Βασικό μέλημα της FIDO είναι η προετοιμασία του τεχνολογικού πεδίου για την πλήρη εγκατάλειψη των συνθηματικών και των γνωσιακών παραγόντων επαλήθευσης, στα οποία το προαναφερθέν πεδίο έχει αναπτύξει τεράστια εξάρτηση.

Κεφάλαιο 5: Λύσεις Ευπαθειών Κωδικών Πρόσβασης

Έχοντας φτάσει ως αυτό το σημείο έχουμε αποδείξει πως η μέθοδος επαλήθευσης χρήστη για την είσοδο σ' ένα σύστημα με την χρήση συνθηματικών αποτελεί πλέον μια πολύ τρωτή μέθοδο. Η βασική χρήση των συνθηματικών είναι η απόρριψη της μη-εξουσιοδοτημένης εισόδου στο σύστημα, κάτι που όμως αποδείξαμε πως δεν ικανοποιεί, καθώς υπάρχει μια πληθώρα τρόπων ώστε αυτά να παρακαμφθούν. Επομένως σε αυτό το κεφάλαιο θα διερευνήσουμε τους πιθανούς διαδόχους των συνθηματικών στην κοινωνία των νέων τεχνολογιών, και θα τις αξιολογήσουμε ως προς την ανθεκτικότητα τους στις προαναφερθείσες επιθέσεις.

5.1. Μέθοδοι Ταυτοποίησης Βασισμένες στα Συνθηματικά

5.1.1. Επαλήθευση μέσω Μοτίβων Πληκτρολόγησης (Keystroke Dynamics) και Ποντικιού (Click Patterns)

Ξεκινώντας την διερεύνηση μας, θα μείνουμε κοντά στην λογική των συνθηματικών και θα μιλήσουμε για την μέθοδο ταυτοποίησης με βάση την δυναμική πληκτρολόγησης ή Keystroke Dynamics. Η μέθοδος αυτή βασίζεται σε δύο πολύ βασικές μετρικές, τα πλήκτρα που πατιούνται και ο χρονισμός με βάση τον οποίο αυτά πατιούνται. Στην ουσία της, αυτή η μέθοδος δεν ασχολείται με το τί πληκτρολογείτε αλλά το πώς πληκτρολογείτε. Για την ανάλυση του χρονισμού, ο αλγόριθμος ανίχνευσης αποθηκεύει πληροφορία για το χρονικό διάστημα μεταξύ της πίεσης του πλήκτρου και της απελευθέρωσης του, το χρόνο μεταξύ της πίεσης δύο πλήκτρων, το όνομα του πλήκτρου που πιέστηκε και τελικώς αποθηκεύεται ο ρυθμός πληκτρολόγησης, ο οποίος αποσκοπεί στο να αποτελέσει βιομετρικό γνώρισμα ώστε να περιγράψει τον χρήστη.

Έμπνευση αυτής της μεθόδου αποτέλεσε ο τηλέγραφος ο οποίος βασίζεται στον ρυθμό με τον οποίο μεταδιδόταν τα σύμβολα της κωδικοποίησης. Η κωδικοποίηση του μηνύματος

βασιζόταν στην κωδικοποίηση εντροπίας. Τα μηνύματα διαφοροποιόταν από τον αριθμό των πιέσεων και απελευθερώσεων του μοναδικού πλήκτρου, το οποίο έστελνε ηλεκτρικά σήματα κατά μήκος ηλεκτρικών καλωδίων. Τα πλεονεκτήματα που προσέδωσε αυτή η μοντελοποίηση της μεθόδου είναι πως για την υλοποίηση της δεν απαιτείται επιπλέον υλικό, παρά μόνο καλές προγραμματιστικές ικανότητες.

Συμπερασματικά, η μέθοδος παρουσιάζει υψηλή αντίσταση στις επιθέσεις τύπου shoulder surfing και keyloggers καθώς όπως προαναφέραμε, το περιεχόμενο της πληκτρολόγησης δεν αποτελεί την πληροφορία που θα επιτρέψει την είσοδο στο σύστημα. Στα αρνητικά της μεθόδου βρίσκεται ο υψηλός ρυθμός εσφαλμένης απόρριψης πρόσβασης λόγω των ανομοιόμορφων ταχυτήτων πληκτρολόγησης που μπορεί να χαρακτηρίζουν τον χρήστη. Όπως είναι λογικό, λίγοι είναι οι χρήστες που παρουσιάζουν σταθερό ρυθμό πληκτρολόγησης, και πόσο μάλλον λίγοι είναι αυτοί που θα θεωρήσουν πως ο ρυθμός πληκτρολόγησης αποτελεί ένα άξιο διάδοχο των συνθηματικών, και όχι απλά μια περίπλοκη και κουραστική μέθοδο επαλήθευσης χρήστη. Συγκεκριμένα, η μέθοδος επηρεάζεται πολύ από την ψυχολογική και σωματική κατάσταση του χρήστη. Π.χ. ένας αγχωμένος χρήστης μπορεί να έχει αυξημένο ρυθμό αστοχιών στην ορθή πίεση ενός πλήκτρου σε σχέση με κάποιον αντίστοιχο ήρεμο χρήστη.

Αντίστοιχα, έχει προταθεί η μέθοδος με την χρήση του ποντικιού (Click Patterns) αντί του πληκτρολογίου. Σε αυτή την περίπτωση επαλήθευσης ο χρήστης καλείται να χρησιμοποιήσει τον κέρσορα του ποντικιού και να επιλέξει περιοχές από μια γραφική διεπαφή χρήστη. Η γραφική διεπαφή μπορεί να αποτελείται από απλά σχήματα ή χρώματα μέχρι και ως περίπλοκα σύμβολα. Τα αντίστοιχα μετρικά που συλλέγονται αφορούν το σχήμα που χάραξε ο χρήστης με το ποντίκι αλλά και τον ρυθμό των κλικ που εκτελούνται. Η μέθοδος αντιστέκεται στις ίδιες επιθέσεις με την περίπτωση του πληκτρολογίου, και αντίστοιχα, εμφανίζει την ίδια αδυναμία, καθώς βασίζεται στην ακρίβεια των χεριών του χρήστη, η οποία μπορεί να επηρεαστεί από ψυχολογικούς και σωματικούς παράγοντες.

Οι παραπάνω μέθοδοι επαλήθευσης, αν και κάπως περίπλοκοι μας παρέχουν μια εικόνα για το πως μπορεί να εγκαταλειφθεί η χρήση των συνθηματικών, και να αντικατασταθεί από στοιχεία που είναι κομμάτια του υπολογιστικού συστήματος. Φυσικά, η παραπάνω προτάσεις δεν λαμβάνουν υπόψη την είσοδο σε συστήματα που δεν αποτελούνται από την

κλασσική αρχιτεκτονική του προσωπικού ηλεκτρονικού υπολογιστή. Η αντίστοιχη διαδικασία για κάποια φορητή συσκευή θα απαιτούσε ακόμα περισσότερη προγραμματιστική εμπειρία και ικανότητα, καθώς η μέτρηση θα γίνει πάνω από οθόνη αφής, της οποίας οι αποκρίσεις στα ερεθίσματα θα πρέπει να καταγράφουν με ακρίβεια[3].

5.1.2. Γραφικά Συνθηματικά (Graphical Passwords)

Με την είσοδο των φορητών συσκευών στην αγορά, βλέπουμε πως η σκέψη και η φαντασία μας πρέπει να λάβει υπόψη νέες παραμέτρους, όπως αυτή της οθόνης αφής. Μια οθόνη αφής δεν μπορεί εύκολα να αντιμετωπιστεί ως ένα φυσικό πληκτρολόγιο, επομένως χρειαζόμαστε μια διαφορετική προσέγγιση. Αν θυμηθούμε από την θεωρία γραφημάτων πως από ένα σύνολο σημείων μπορεί να σχεδιαστούν πολλαπλοί γράφοι, τότε μπορούμε να δημιουργήσουμε μια μέθοδο επαλήθευσης που θα βασίζεται σε συγκεκριμένη σύνδεση των αναφερθέντων σημείων.

Τα γραφικά συνθηματικά μπορούν να λάβουν πολλές μορφές. Από μια απλή χάραξη μιας μονοκονδυλιάς, μέχρι και τον σχεδιασμό ενός συγκεκριμένου σχήματος ή ακόμα ολόκληρου σχεδίου. Μια απλή περίπτωση χρήσης είναι πως σε συνέχεια της εισαγωγής του ονόματος του, ο χρήστης δύναται να επιλέξει κάποια συγκεκριμένα αντικείμενα. Ακολούθως, ο χρήστης καλείται να χαράξει τα προεπιλεχθέντα αντικείμενα με την χρήση του ποντικιού, της οθόνης αφής, της γραφίδας ή της πινακίδας αφής. Το σύστημα στην συνέχεια βαθμολογεί το σχήμα που χάραξε ο χρήστης, με αυτό που είχε προεπιλεγεί. Στην περίπτωση υψηλής βαθμολογίας, ο χρήστης ταυτοποιείται και εισέρχεται στο σύστημα, ενώ στην αντίθετη περίπτωση απορρίπτεται.

Η μέθοδος του γραφικού σχεδιασμού μπορεί οριακά να χαρακτηριστεί ως επαλήθευση μέσω παιγνίου, κατά την οποία ο χρήστης συμμετέχει σε ένα παιχνίδι σχεδιασμού, όπου αν βγει νικητής, του επιτρέπεται η πρόσβαση. Στα πλεονεκτήματα της μεθόδου συγκαταλέγεται η ανθεκτικότητα σε επιθέσεις shoulder-surfing, καθώς ο δράστης δεν μπορεί να λάβει κανένα συμπέρασμα για την πλήρως ορθή αναπαράσταση του αντικειμένου. Στα μειονεκτήματα παρατηρείται πως η ταυτοποίηση γίνεται βασισμένη στην άμεση αλληλεπίδραση του χρήστη με την μηχανή, επομένως απαιτείται επιπλέον υλικό για αυτήν, όπως γραφίδες, επιφάνειες αφής κτλ. Η γραφική διεπαφή, πέρα του ότι πρέπει να είναι ακριβείας, παρουσιάζει

καθυστερήσεις υπολογισμού της ορθής αναπαράστασης, καθώς απαιτούνται αλγόριθμοι αναγνώρισης εικόνας. Επιπλέον, η μέθοδος βασίζεται τόσο στην καλλιτεχνική δεξιότητα του χρήστη, η οποία για κανένα λόγο δεν μπορεί να θεωρηθεί δεδομένη, αλλά και πάλι μπορεί να επηρεαστεί από την ψυχολογική και σωματική κατάσταση του χρήστη, αλλά και από παράγοντες όπως είναι η βιασύνη.

Αν και δεν προτείνονται ως διάδοχοι των συνθηματικών, τα γραφικά συνθηματικά χρησιμοποιούνται ευρέως στις κινητές συσκευές για την απλή λειτουργία του ξεκλειδώματος της οθόνης. Φυσικά, ένα τέτοιου είδους μοτίβο μπορεί εύκολα να καταγραφεί και να παραβιαστεί, επομένως δεν αποτελεί κατά κανένα τρόπο λύση του προβλήματος.

5.1.3. Βιομετρικά (Biometrics)

Όπως έχει ήδη προαναφερθεί συνοπτικά σε προηγούμενη ενότητα, η επαλήθευση ταυτότητας με την χρήση βιομετρικών αποτελεί μια από τις ισχυρότερες μεθόδους. Η διαδικασία αξιοποιεί τον τομέα της ψηφιακής επεξεργασίας εικόνας, καθώς κατά την ανάγνωση του χαρακτηριστικού ακολουθείται εξαγωγή των απαραίτητων πληροφοριών από την εικόνα και τέλος οι πληροφορίες συγκρίνονται με αυτές της βάσης δεδομένων για την επικύρωση του χρήστη.

Υπάρχουν πολλά βιομετρικά χαρακτηριστικά που μπορούν να χρησιμοποιηθούν και ήδη χρησιμοποιούνται:

→ **Ανάγνωση Δακτυλικού Αποτυπώματος (Finger print Authentication):** Αποτελούν τις πιο διαδεδομένες περιπτώσεις χρήσης των βιομετρικών, και τις πιο επιτυχημένες. Υπάρχουν πολλαπλές ιστορικές αναφορές που δηλώνουν πως η χρήση των δακτυλικών αποτυπωμάτων χρησιμοποιούταν στο εμπόριο από το 500 π.Χ. στην Βαβυλωνία, και από Κινέζους αξιωματούχους τον 3^{ο} αιώνα π.Χ. Τα δακτυλικά αποτυπώματα αποτελούνται από ένα μοτίβο που αποτελείται από χαρακώματα και υψώματα. Τα χαρακτηριστικά αυτά δημιουργούν αυτά που ονομάζονται μικρολεπτομέρειες του αποτυπώματος, οι οποίες αποτελούνται από απολήξεις και διακλαδώσεις των χαρακωμάτων. Η χωρική κατανομή των χαρακτηριστικών αυτών

είναι μοναδική για κάθε δάκτυλο και για κάθε άνθρωπο. Όσο μεγαλύτερη ανάλυση εικόνας χρησιμοποιηθεί, τόσο περισσότερα επίπεδα λεπτομέρειας μπορούν να εντοπιστούν, τα οποία διακρίνουν περαιτέρω τα αποτυπώματα μεταξύ τους. Η εξέλιξη στον τομέα των σαρωτές δακτυλικών αποτυπωμάτων επιτρέπουν την ανάλυση τους ακόμα και σε επίπεδο πόρων, κάτι το οποίο χρησιμοποιείται από τον χώρο της Εγκληματολογίας. Με το πέρασμα των χρόνων, οι σαρωτές δακτυλικών αποτυπωμάτων έχουν πέσει σε κόστος και σε μέγεθος, επιτρέποντας την χρήση τους σε όλο και περισσότερα συστήματα.

→ **Αναγνώριση Προσώπου (Face Recognition):** Οι άνθρωποι έχουν την ικανότητα να αναγνωρίζουν τους συνανθρώπους τους με βάση τα μορφολογικά χαρακτηριστικά του προσώπου τους. Επομένως, η αναγνώριση προσώπου είναι λογικό να αποτελεί μια από τις πιο συχνές εφαρμογές χρήσης βιομετρικών. Οι αλγόριθμοι αναγνώρισης εκμεταλλεύονται τις χωρικές σχέσεις μεταξύ σημείων αναφοράς (μάτια, μύτη, χείλια, πιγούνι και το σχήμα του προσώπου), για να εξάγουν την σύγκριση με την πληροφορία που βρίσκεται αποθηκευμένη στην βάση δεδομένων. Τα βασικά προβλήματα που αντιμετωπίζει η αναγνώριση προσώπου έχουν να κάνουν με την φωτεινότητα, τους μορφασμούς, την ύπαρξη καλλυντικών προσώπου, την συνοφρύωση και την προοπτική. Η αναγνώριση προσώπου αποτελεί ακόμα τομέα της υπολογιστικής όρασης που εξερευνάται και ανανεώνεται τακτικά.

→ **Επικύρωση Υπογραφής (Signature Verification):** Η υπογραφή μας αποτελεί ένα από τα πιο χαρακτηριστικά γνωρίσματα επικύρωσης. Ο γραφικός μας χαρακτήρας αποτελεί αντιπροσωπευτικό γνώρισμα, το οποίο υπάρχει ξεχωριστός κλάδος που τον μελετά ονόματι Γραφολογία. Οι εφαρμογές αναγνώρισης υπογραφής ακόμα υστερούν, αλλά οι ψηφιακές υπογραφές αποτελούν συχνά εμφανιζόμενη μέθοδο κατά την σύναψη συμβολαίων και συμβάσεων. Το πρόβλημα της αναγνώρισης υπογραφής αποτελείται από πολλαπλές συνιστώσες, συγκεκριμένα την αναγνώριση πληροφοριών πίεσης γραφίδας (αν αυτές είναι διαθέσιμες), την εξαγωγή του σχήματος, της ταχύτητας, της επιτάχυνσης, της σειράς και της ταχύτητας των γραμμών κατά την σχεδίαση της υπογραφής. Ακόμα λίγα είναι τα συστήματα που υλοποιούν αυτοματοποιημένες εφαρμογές αναγνώρισης υπογραφών.

→ **Αναγνώριση Ομιλίας (Speech Recognition):** Η ανάλυση του φάσματος συχνοτήτων της φωνής χρησιμοποιείται για την εξαγωγή πληροφοριών που αφορούν την ένταση, της διάρκεια, την ποιότητα και την οξύτητα, οι οποίες με την σειρά τους συγκροτούν ένα μοντέλο αναγνώρισης, συνήθως ένα HMM – Hidden Markovian Model. Η αναγνώριση φωνής χρησιμοποιείται εκτενώς σε εφαρμογές τηλεφωνικής τραπεζικής, είναι όμως ευαίσθητη στον θόρυβο και στο φαινόμενο της επιστροφής ήχου, προκαλώντας την ανάγκη για εκτενή ψηφιακή επεξεργασία σήματος.

→ **Ανάγνωση και Αναγνώριση Ίριδας Ματιού (Iris Recognition):** Οι εικόνες της ίριδος αποκτώνται μέσω φωτισμού με υπέρυθρο φως, και η πληροφορία που αποσπάται περιέχει περίπλοκα μοτίβα υφών με πολλαπλά επιπλέον χαρακτηριστικά όπως είναι ρίγες, λακκούβες και αυλάκια. Όλες αυτές οι πληροφορίες επιτρέπουν την υψηλής ποιότητα αναγνώρισης, καθώς για κάθε άνθρωπο, η κατανομή αυτών των πληροφοριών είναι διαφορετική.

Η μέθοδος παρουσιάζει τα πιο ισχυρά πλεονεκτήματα καθώς συνδυάζει μοναδικές και πραγματικές υπογραφές που δεν μπορούν να υποκλαπούν με κανέναν τρόπο. Στα μειονεκτήματα της βρίσκεται το υψηλό κόστος υλοποίησης λόγω της ανάγκης αισθητήρων υψηλής ακρίβειας, ο χρόνος ολοκλήρωσης της ανάλυσης και επαλήθευσης είναι ακόμα κάπως μεγάλος, και τέλος πως πολλές φορές μπορεί να υπάρξουν αστοχίες λόγω παρεμβολών μεταξύ του μέσου ταυτοποίησης και του βιομετρικού χαρακτηριστικού. Π.χ. Αν τα χέρια ενός χρήστη ιδρώνουν αρκετά, τότε υπάρχει η περίπτωση ο αισθητήρας δακτυλικού αποτυπώματος να μην αναγνωρίσει ορθά το αποτύπωμα. Επίσης, στην περίπτωση λερωμένης κάμερας ή σε καταστάσεις χαμηλού φωτισμού, η αναγνώριση προσώπου μπορεί να μην λειτουργήσει σωστά.

Με την ανάπτυξη της τεχνολογίας ο χώρος, τόσο των συστημάτων αισθητήρων, όσο και της τεχνητής νοημοσύνης και της μηχανικής μάθησης, έχει αναπτυχθεί έτσι ώστε οι προηγούμενοι περιορισμοί να τείνουν να εκμηδενιστούν. Η απόκριση των αισθητήρων ιδιαίτερα έχει σημάνει μεγάλη πρόοδο, έχοντας πολύ υψηλές ταχύτητες ανάγνωσης, ενώ στο κομμάτι της η υπολογιστική νοημοσύνη μπορεί να εξάγει το αποτέλεσμα της επαλήθευσης σε μερικά δευτερόλεπτα.

Συγκριτικά με τις υπόλοιπες μεθόδους που μελετήσαμε σε αυτό το κεφάλαιο, τα βιομετρικά ίσως αποτελούν τον πιθανότερο διάδοχο των συνθηματικών. Η επαλήθευση μέσο βιομετρικών χαρακτηριστικών επιτυγχάνει την ισορροπία μεταξύ ευκολίας χρήσης και επιπέδου ασφαλείας, κάνοντας την αποδεκτή από την πλειοψηφία του πληθυσμού. Στο κομμάτι της ευκολίας και άνεσης, η διαδικασία αποδεσμεύει τον χρήστη από το να θυμάται περίπλοκες και μεγάλου μήκους ακολουθίες χαρακτήρων, αριθμών και συμβόλων, καθώς το διαπιστευτήριο ταυτότητας του βρίσκεται συνεχώς πάνω του. Από την πλευρά της ασφάλειας, ένα βιομετρικό χαρακτηριστικό είναι απίθανο να κλαπεί, καθώς δεν βρισκόμαστε σε ταινία κατασκόπων. Ένα βιομετρικό χαρακτηριστικό αντιπροσωπεύει τον χρήστη μοναδικά και δια βίου, επομένως δεν τίθεται ζήτημα σύγχυσης ή απαρχαίωσης της πληροφορίας.

5.1.4. Fast Identity Online (FIDO) Alliance

Η συμμαχία FIDO ή Fast Identity Online πραγματεύεται την ανάπτυξη και προώθηση προτύπων επαλήθευσης χρήστη, η οποία έχει ως απώτερο σκοπό να ελαττώσει την εξάρτηση του κόσμου από τα συνθηματικά. Συγκεκριμένα, τονίζεται η έλλειψη διασύνδεσης μεταξύ συσκευών που χρησιμοποιούν αυστηρή επαλήθευση και αποσκοπείται η μείωση των συνθηματικών που πρέπει να θυμούνται οι χρήστες. Στην ευρεία γκάμα των τεχνολογιών που υποστηρίζονται από την συμμαχία συγκαταλέγονται τα βιομετρικά, τα USB-keys, οι έξυπνες κάρτες και η τεχνολογία NFC. Όλες οι προδιαγραφές βασίζονται στον σχεδιασμό με κέντρο τις συσκευές. Η επαλήθευση πάνω από το δίκτυο γίνεται μέσω κρυπτογράφηση δημόσιου κλειδιού. Η συσκευή του χρήστη καταχωρείτε σε εξυπηρετητή με την χρήση ενός δημόσιου κλειδιού. Για την επαλήθευση του χρήστη, η συσκευή υπογράφει μια αίτηση από τον εξυπηρετητή με την χρήση του ιδιωτικού κλειδιού που κατέχει. Τα κλειδιά στο FIDO παρέχουν δύο τύπους εμπειρίας χρήστη βασισμένες πάντα στο χρησιμοποιούμενο πρωτόκολλο. Και τα δύο πρωτόκολλα ορίζουν μια κοινή διεπαφή στον πελάτη, για οποιαδήποτε τοπική επαλήθευση χρειαστεί ο χρήστης.

5.2. Συγκριτική Αξιολόγηση

Στην παρακάτω ενότητα θα αποσκοπήσουμε στην ανάλυση και αξιολόγηση των παραπάνω λύσεων με βάση τον παρακάτω συλλογικό πίνακα συμπερασμάτων αλλά και την μεταφορά του σε μια κλίμακα δικιάς μας κατασκευής.

Μέθοδος	Ανθεκτικότητα	Ειδικό Υλικό	Κόστος	Ψυχολογική Κατάσταση	Επίπεδο Προστασίας	Χρόνος
Συνθηματικά	Όχι	Όχι	Κανονικό	Όχι	Χαμηλό	Γρήγορο
Keystroke Dynamics	Shoulder Surfing, phishing, keyloggers	Όχι	Κανονικό	Ναι	Μέτριο	Μέτριος
Click Patterns	Shoulder Surfing, phishing, keyloggers	Όχι	Κανονικό	Ναι	Μέτριο	Μέτριος
Γραφικά	Shoulder Surfing	Ναι	Υψηλό	Ναι	Μέτριο	Αργός
Βιομετρικά	Shoulder Surfing, phishing, keyloggers, etc.	Ναι	Υψηλό	Όχι	Υψηλό	Αργός

Πίνακας 1: Πίνακας Χαρακτηριστικών Μεθόδων Επαλήθευσης Χρήστη

Στην συνέχεια θα παρουσιάσουμε την κλίμακα αξιολόγηση που θα αποτελέσει το μέτρο ποιότητας των μεθόδων αντιμετώπισης των ευπαθειών.

- **Ανθεκτικότητα:** Η ανθεκτικότητα θα οριστεί ως μια τιμή ανάλογη του αριθμού των επιθέσεων, κατά των οποίων η μέθοδος θωρακίζει το σύστημα. Συνολικά παρουσιάσαμε τέσσερεις επιθέσεις που συνεχίζουν να καραδοκούν παρά την απαλοιφή των συνθηματικών. Συγκεκριμένα με την εγκατάλειψη των συνθηματικών αντιμετωπίζονται άμεσα επιθέσεις όπως οι εξαντλητικής αναζήτησης, λεξικών και Rainbow Tables. Επομένως η αντίστοιχη ποσοτική μεταβλητή θα κυμαίνεται εντός του εύρους 0…3.
- **Ειδικό Υλικό:** Η μετρική αυτή αποτελεί δυαδική μεταβλητή με τις τιμές 0 ή 1.

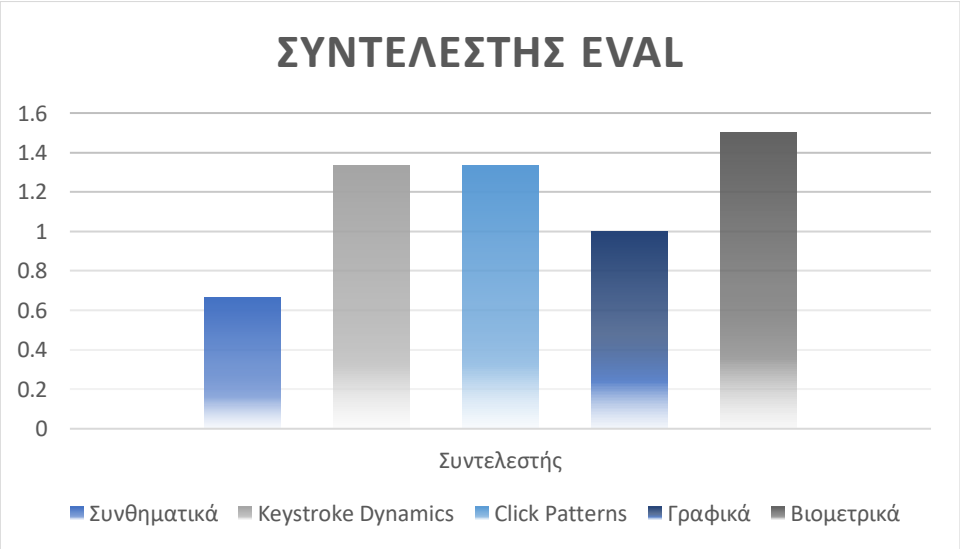
- **Κόστος:** Η μετρική αυτή ορίζεται στο εύρος 0…2 εφόσον υπάρχουν τρείς διαφορετικές εναλλακτικές: χαμηλό, μέτριο, υψηλό.
- **Ψυχολογική Κατάσταση Χρήστη:** Ομοίως με την μεταβλητή «Ειδικό Υλικό».
- **Επίπεδο Προστασίας:** Ομοίως με την μεταβλητή κόστος.
- **Χρόνος:** Θα μετατραπεί στην κλίμακα 0…3 με τις αντίστοιχες τιμές να ορίζονται στις περιπτώσεις Αργός, Αυξανόμενος, Μέτριος και Γρήγορος αντίστοιχα.

Επομένως για την κάθε μέθοδο έχουμε ένα μέτρο αξιολόγησης, το οποίο θα ονομάσουμε συντελεστή Eval. Ο συντελεστής Eval ορίζεται ως το άθροισμα τον επιμέρους βαθμολογιών και την διαίρεση του αθροίσματος με το πλήθος των χαρακτηριστικών. Οι τιμές του eval ξεκινάνε με ελάχιστο το 0 και μέγιστο το 2.167.

Μέθοδος	Συντελεστής Eval
Συνθηματικά	0.67
Keystroke Dynamics	1.33
Click Patterns	1.33
Γραφικά	1
Βιομετρικά	1.5

Πίνακας 2: Πίνακας Συντελεστών Eval

Όπως θα δούμε από την γραφική απεικόνιση, το οποίο είναι και προφανές, είναι πως η κάθε μέθοδος αποτελεί καλύτερη από την επαλήθευση με την χρήση συνθηματικών.



Γράφημα 1: Γράφημα Συντελεστή Eval ως προς τις διαφορετικές μεθόδους επαλήθευσης χρήστη

Με την γραφική απεικόνιση μπορούμε να αντιληφθούμε και οπτικά πως τα συνθηματικά αποτελούν πλέον μια από τις χειρότερες τεχνικές επαλήθευσης χρήστη. Από την άλλη τα βιομετρικά κατέχουν την πρώτη θέση, εφόσον όπως είπαμε αποτελούν εγγενώς μια πιο ασφαλή και σίγουρη μέθοδο επαλήθευσης. Στην συνέχεια θα μελετήσουμε μια προτεινόμενη εναλλακτική προσέγγιση για την επαλήθευση χρήστη.

Κεφάλαιο 6: Κατανεμημένη Προσέγγιση disCode

Όπως είδαμε και κατά την άνωθεν μελέτη, η παραδοσιακή προσέγγιση επαλήθευσης χρήστη με την χρήση συνθηματικών αποτελεί μια από τις πιο τρωτές και ευπαθείς μεθόδους. Η χρήση των συνθηματικών ενθαρρύνει τους επίδοξους εισβολής καθώς τους αφήνει πολλαπλές πτυχές που μπορούν να εκμεταλλευτούν. Επομένως, με την γνώση που λάβαμε από τις παραπάνω ενότητες, στην παρούσα θα αποσκοπήσουμε να προτείνουμε εμείς μια μέθοδο επαλήθευσης χρήστη.

6.1. Κατανεμημένα Συστήματα

Ένα κατανεμημένο σύστημα είναι ένα υπολογιστικό περιβάλλον στο οποίο διάφορα στοιχεία είναι κατανεμημένα σε πολλούς υπολογιστές (ή σε άλλες υπολογιστικές μονάδες/συσκευές). Η επικοινωνία των υπολογιστικών συστημάτων υποστηρίζεται από μια υποδομή δικτύου. Σε αυτά τα συστήματα οι εργασίες χωρίζονται σε υποδεέστερα τμήματα τα οποία αποστέλλονται στους ξεχωριστούς υπολογιστικούς κόμβους. Η λειτουργία των υπολογιστικών κόμβων συντονίζεται με στόχο την ολοκλήρωση της αρχικής εργασίας.

6.2. Μια πολυϋπολογιστική προσέγγιση στην αυθεντικοποίηση χρήστη (disCode)

6.2.1. Στόχος

Η εφαρμογή μας η οποία ονομάζεται **disCode**, έχει στόχο να παρουσιάσει μια πολυϋπολογιστική προσέγγιση στο πρόβλημα αυθεντικοποίησης χρήστη. Η ύπαρξη μας στο διαδίκτυο ακολουθείται από πολλούς κινδύνους, με τον βασικότερο την υποκλοπή των προσωπικών μας δεδομένων. Για την ασφάλεια των χρηστών, ο πιο διαδεδομένος τρόπος αποτελεί η χρήση συνθηματικών πρόσβασης. Όπως αναλύσαμε στις παραπάνω ενότητες, τα συνθηματικά αποτελούν μια από τις ποιο ευπαθείς μεθόδους ασφάλειας. Συνεπώς, στόχος μας σε αυτή την ενότητα αποτελεί ο συνδυασμός των συμπερασμάτων μας με αποτέλεσμα τον σχεδιασμό ενός συστήματος που δεν εξαρτάται από τα συνθηματικά ασφαλείας.

6.2.2. Σχεδιασμός

Ως εκ τούτου, το σύστημα μας θα αξιοποιεί τα OTP και μια επιπλέον διαδικασία ελέγχου με βάση την σύγκριση των στοιχείων του μηχανήματος χρήστη. Αν και στις περιπτώσεις κλοπής των στοιχείων εισόδου τα συστήματα δεν μπορούν να γνωρίζουν πως ο ανθρώπινος χρήστης είναι διαφορετικός, μπορούν όμως να γνωρίζουν τα στοιχεία του μηχανήματος που χρησιμοποιείται για την εισαγωγή τους. Όπως γνωρίζουμε, πολύ συχνά ιστοσελίδες όπως το Facebook και το Google μας ενημερώνουν για συνδέσεις από άγνωστες IP. Επεκτείνοντας σε αυτό δημιουργούμε ένα σύστημα που πέρα από το OTP θα χρησιμοποιεί και τα στοιχεία του φυσικού μηχανήματος για να επικυρώσει τον χρήστη.

Η εφαρμογή αποτελείται από έναν πελάτη και μια ομάδα από εξυπηρετητές. Στην πλευρά του πελάτη έχουμε την διαδικασία εγγραφής (sign-up) και σύνδεσης (sign-in). Ο πελάτης επιλέγει την διαδικασία που επιθυμεί να εκτελέσει. Στην περίπτωση της εγγραφής ο χρήστης μόλις επιλέξει την εγγραφή λαμβάνει ένα OTP και με βάση αυτό εκτελεί την εγγραφή του. Με την ορθή εισαγωγή του OTP ο χρήστης στέλνει το όνομα χρήστη και τα στοιχεία της μηχανής και του συστήματος που χρησιμοποιεί. Δημιουργείται μια εγγραφή στο παρόν εξυπηρετητή εγγραφής και αυτός στην συνέχεια εκτελεί broadcast για να μεταδώσει την πληροφορία στους υπόλοιπους εξυπηρετητές της ομάδας.

Σε αυτό το σημείο καλό θα ήταν να αναφερθούμε στο κατανεμημένο μας σύστημα. Το σύστημα μπορεί να αποτελείται από N υπολογιστικούς σταθμούς. Ο σταθμός με rank ίσο με 0 αποτελεί τον υπολογιστή χρήστη ή πελάτης. Ο σταθμός με rank ίσο με 1 αποτελεί τον OTP Server που αναλαμβάνει την παραγωγή και αποστολή των OTPs. Οι σταθμοί με rank 2 έως N-1 αποτελούν την ομάδα επαλήθευσης. Σε μια κανονική εφαρμογή, πριν την αποδοχή αιτημάτων πελατών, η ομάδα επαλήθευσης εκλέγει έναν αρχηγό που θα αποτελέσει τον εκπρόσωπο και κεντρικό ελεγκτή. Ο κεντρικός ελεγκτής συλλέγει τα δεδομένα και τα εκπέμπει στους υπόλοιπους σταθμούς της ομάδας ώστε να τον βοηθήσουν να λάβει την τελική απόφαση.

6.2.3. Κώδικας

Για την υλοποίηση του κώδικα χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python, και συγκεκριμένα γράφηκε κώδικας με την χρήση των πακέτων **mpi4py**, **platform** και

το δικό μας **discodelib**. Το mpi4py αποτελεί πακέτο που εισάγει την διεπαφή MPI στον κώδικα Python, ώστε να προσομοιώσουμε την ύπαρξη πολλών υπολογιστικών σταθμών. Το πακέτο platform χρησιμοποιείται για την ανάκτηση πληροφοριών του συστήματος όπως ο τύπος του επεξεργαστή ή το λειτουργικό σύστημα. Από το πακέτο discodelib που δημιουργήσαμε χρησιμοποιούμε τις συναρτήσεις **generateOTP()** για την παραγωγή του OTP και την getInfo για την εκτέλεση της συλλογής πληροφοριών του συστήματος.

```
def generateOTP():
    digits = "0123456789"
    OTP = ""

    for i in range(4):
        OTP += digits[math.floor(random.random()*10)]

    return OTP

def verifyOTP(OTP):
    a = input("Enter your OTP >>:")
    if a == OTP:
        print("Verified")
    else:
        print("Check your OTP again")

def getInfo():
    keys = ['machine', 'platform', 'uname', 'system', 'processor']
    values = [plt.machine(), plt.version(), plt.platform(), plt.uname(), plt.system(), plt.processor()]
    data = dict(zip(keys, values))
    return data

if __name__ == "__main__":
    print("Running code")
    OTP = generateOTP()
    print(OTP)
    verifyOTP(OTP)
```

Εικόνα 29: Βιβλιοθήκη discodelib

Ο κώδικας ξεκινά με την προετοιμασία του περιβάλλοντος MPI. Συγκεκριμένα ξεκινάμε με την ανάθεση του καθολικού communicator στην μεταβλητή comm. Ο καθολικός communicator χρησιμοποιείται για την επικοινωνία των υπολογιστικών σταθμών πάνω από την διεπαφή. Στην συνέχεια χρησιμοποιούμε τις μεθόδους comm.Get_rank() και comm.Get_size() για την ανάθεση των αναγνωριστικών και του μεγέθους του συστήματος αντίστοιχα. Κάθε υπολογιστικός σταθμός θα του ανατεθεί μια τιμή από το 0 έως το N-1, όπου N το πλήθος όλων των σταθμών.

Με την ολοκλήρωση της ανάθεσης των βασικών στοιχείων του MPI θα δημιουργήσουμε τις δομές που θα χρησιμοποιήσουμε για την επικοινωνία. Οι δομές αυτές

έχουν ως σκοπό την υλοποίηση μηνυμάτων που μπορούν να μεταδοθούν μέσω των συναρτήσεων του MPI. Σε αντίθεση με την γλώσσα C, το MPI στην γλώσσα Python δεν μπορεί να χρησιμοποιήσει μεμονωμένες τιμές με την μορφή δείκτη, αλλά τις χρησιμοποιεί ως δεικτοδοτούμενα αντικείμενα (λίστες, λεξικά). Για παράδειγμα, η αποστολή ενός ακεραίου στην C γίνεται ως **MPI_Send(void* data, int count, MPI_Datatype datatype, int destination, int tag, MPI_Comm communicator)** ενώ η αντίστοιχη μέθοδος στην Python ορίζεται ως **comm.send(obj, dest, tag=0)**, όπου obj ένα οποιοδήποτε αντικείμενο και dest ο κόμβος προορισμού. Συνεπώς, κάθε μήνυμα θα αποτελεί ένα λεξικό, συναφές με τον σκοπό του μηνύματος.

6.2.4. Μηνύματα και Δομές

Τα μηνύματα που ανταλλάσσονται εντός της εφαρμογής αποτελούν την βάση της εφαρμογής. Όπως γνωρίζουμε, η διεπαφή MPI είναι η βασική για την υλοποίηση της ανταλλαγής μηνυμάτων και πως τα κατανεμημένα συστήματα αξιολογούνται με βάση την πολυπλοκότητα μηνυμάτων. Ως αποτέλεσμα έχουμε διαφορετικά μηνύματα που κινούνται εντός του συστήματος, αποκτώντας έτσι μια πρακτική επαφή με την θεωρητική φράση «Τα κατανεμημένα συστήματα εμφανίζουν πολλές ανταλλαγές μηνυμάτων». Τα μηνύματα που κινούνται εντός της εφαρμογής φαίνονται παρακάτω:

1. **Msg1:** Το μήνυμα αυτό αποτελεί ένα βασικό μήνυμα για την έναρξη της επικοινωνίας μεταξύ δύο σταθμών. Επιπλέον χρησιμοποιείται ως σήμα εκκίνησης ή απλά ως φορέας εισόδου στον επαληθευτή από τον χρήστη.

2. **Msg2:** Το μήνυμα χρησιμοποιείται για την αποστολή και παραλαβή του OTP από τον OTP-Server.

3. **Data:** Αποτελεί αντικείμενο που απλά θα παραλάβει κάποια δεδομένα.

4. **Retval:** Αποτελεί μήνυμα που γνωστοποιεί τα αποτελέσματα μιας διαδικασίας True/False σε κάποιον σταθμό. Χρησιμοποιείται για να ενημερώσει τον χρήστη για την σύνδεση του στο σύστημα ή την εισαγωγή του OTP.

5. **Entry:** Σε περίπτωση σύνδεσης του χρήστη, αυτό το μήνυμα περιέχει το όνομα χρήστη και τα στοιχεία της φυσικής μηχανής του χρήστη.

6. **Users:** Μια λίστα όπου θα τοποθετούνται εγγραφές χρηστών. Το αντικείμενο αυτό διαμοιράζεται σε όλους τους εξυπηρετητές αυθεντικοποίησης.

6.2.5. Κώδικας Πελάτη/Χρήστη

Έχοντας αναλύσει τα μηνύματα και τις σταθερές του συστήματος θα προχωρήσουμε στην ανάλυση των βασικών ενεργών στοιχείων του. Επιγραμματικά αυτά ορίζονται ως: Ο χρήστης, ο αυθεντικοποιητής, η ομάδα αυθεντικοποιητών και ο OTP-Server. Ο χρήστης του συστήματος ορίζεται ως ο κόμβος 0. Με την εκτέλεση του, ο χρήστης καλείται να επιλέξει ανάμεσα σε δύο λειτουργίες, την εγγραφή και την είσοδο στο σύστημα. Για την επιλογή της εγγραφής, ο χρήστης πρέπει να δώσει την τιμή 1 στην είσοδο του προγράμματος. Με την ανάγνωση της επιλογής, αποστέλλεται το μήνυμα msg1, το οποίο ενημερώνει τον εξυπηρετητή αυθεντικοποίησης πως ένας χρήστης επιθυμεί να εγγραφεί στο σύστημα. Στην συνέχεια ο πελάτης αναμένει την λήψη ενός OTP. Μετά από διαδικασία που θα αναλύσουμε παρακάτω έχουμε την παραλαβή του μηνύματος με την τιμή του OTP. Στην συνέχεια ο χρήστης καλείται να καταθέσει το OTP στον εξυπηρετητή αυθεντικοποίησης για την εκκίνηση της εγγραφής στο σύστημα. Με την κατάθεση του OTP, ο πελάτης αναμένει την παραλαβή της έκβασης της αυθεντικοποίησης. Με την επιστροφή έχουμε δύο περιπτώσεις:

1. Αν η τιμή επιστροφής μέσω του μηνύματος retval είναι True, τότε ο χρήστης θα στείλει το μήνυμα entry στον εξυπηρετητή. Το μήνυμα entry περιέχει πεδία για το επιθυμητό όνομα χρήστη και τα στοιχεία του φυσικού μηχανήματος που εξάγονται μέσω της συνάρτησης getInfo().

2. Αν η τιμή επιστροφής είναι False, τότε το μήνυμα entry γεμίζεται με τις τιμές None.

Ανεξάρτητα από την έκβαση του αποτελέσματος, ο χρήστης θα στείλει στον εξυπηρετητή αυθεντικοποίησης το μήνυμα entry. Με την διαδικασία αυτή ο χρήστης ολοκληρώνει την διαδικασία εγγραφής. Ο εξυπηρετητής αυθεντικοποίησης θα αποφασίσει αν ο χρήστης θα εγγραφεί ή όχι στο σύστημα.

Κατά την δεύτερη επιλογή, την διαδικασία της εισόδου στο σύστημα, προηγείται ο έλεγχος για το αν ο χρήστης είναι εγγεγραμμένος στο σύστημα. Ομοίως με την προηγούμενη διαδικασία, ο χρήστης αποστέλλει στον εξυπηρετητή αυθεντικοποίησης το όνομα χρήστη

του και αναμένει το αποτέλεσμα του ελέγχου. Ομοίως με την εγγραφή, ο χρήστης δέχεται δύο πιθανά αποτελέσματα:

1. Αν η τιμή είναι True, τότε ο χρήστης αποδέχεται ένα OTP και καλείται να το εισάγει για την είσοδο. Με την εισαγωγή του OTP ο χρήστης ενημερώνεται για την έκβαση της εισαγωγής. Αν το OTP είναι σωστό, τότε ο χρήστης αποστέλλει τα στοιχεία μηχανήματος για την περαιτέρω αυθεντικοποίηση. Με την αποστολή του μηνύματος entry, αναμένει το τελικό αποτέλεσμα για το αν έχει δικαίωμα εισόδου στο σύστημα.
2. Αν η τιμή είναι False τότε ο χρήστης απλά ενημερώνεται πως δεν είναι εγγεγραμμένος στο σύστημα.

```
if rank == 0:
    #   this is the client
    while(1):
        option = int(input("Sign-up (1) or Sign-in (2): "))
        msg1['option'] = option
        if(option == 1):
            #   this is the sign-up
            #   the user informs the master server about the sign-up request
            comm.send(msg1, dest=2,tag=11)
            #   the server will request the otp from the otp server
            msg2 = comm.recv(source=1, tag=12)
            print(rank, msg2['OTP'])
            #   time to check the OTP by sending it to the connection server
            msg1['word'] = input("Enter your OTP>>:")
            comm.send(msg1, dest=2, tag=13)
            #   wait for the response
            retval = comm.recv(source=2,tag=14)
            print(retval['message'])
            if(retval['flag']):
                #   this means the user has been accepted
                entry['usrname'] = input("Enter a username: ")
                entry['info'] = getInfo()
            else:
                entry['usrname'] = None
                entry['info'] = None
            #   send the entry to the server
            comm.send(entry,dest=2,tag=15)
```

Εικόνα 30: Κώδικας πελάτη (client) 1

```
elif(option == 2):
    #   this is the sign-in operation
    #   the user is already registered in the list
    #   1. he sends the username, the username is checked
    #   2. he gets the OTP and is authenticated
    #   3. before he is allowed entrance he is checked by the machine information field
    #   4. if all ok then the test is passed
    comm.send(msg1, dest=2, tag=11)
    entry['usrname'] = input("Enter your registered username: ")
    comm.send(entry, dest=2, tag=20)
    #   wait for the response from the authentication server
    retval = comm.recv(source=2,tag=21)
    if(retval['flag']):
        print(rank, "User found in system! Prepare to enter OTP!")
        #   the server sends for the OTP
        msg2 = comm.recv(source=1, tag=12)
        print(msg2['OTP'])
        msg1['word'] = input("Enter your OTP >>:")
        comm.send(msg1, dest=2, tag=23)
        retval = comm.recv(source=2,tag=24)
        if(retval['flag']):
            entry['info'] = getInfo()
        else:
            entry['info'] = None
        #   send the info to the server
        comm.send(entry, dest=2, tag=25)
        retval = comm.recv(source=2, tag=26)
```

Εικόνα 31: Κώδικας πελάτη (client) 2

```
            print(retval['message'])
        else:
            print("User not found in the system...try to sign-up\n")
    else:
        print("Invalid Option! Try again")
```

Εικόνα 32: Κώδικας πελάτη (client) 3

6.2.6. Κώδικας Επαληθευτή

Με την άνω περιγραφή έχει ολοκληρωθεί το κομμάτι του πελάτη/χρήστη. Στην συνέχεια θα μελετήσουμε το κομμάτι του εξυπηρετητή αυθεντικοποίησης. Ο εξυπηρετητής δέχεται το μήνυμα επιλογής του χρήστη. Αν πρόκειται για την επιλογή εγγραφής, ο εξυπηρετητής στέλνει άμεσα μήνυμα msg1 στον OTP-Server για την παραγωγή και αποστολή

των OTP. Με την σειρά του, ο εξυπηρετητής αποδέχεται το OTP, το τοποθετεί σε μια τοπική μεταβλητή και αναμένει για την αποστολή του OTP από τον πελάτη. Με την παραλαβή του μηνύματος ο εξυπηρετητής εκτελεί την σύγκριση και στέλνει το αποτέλεσμα της σύγκρισης στον πελάτη μαζί με το αντίστοιχο μήνυμα κειμένου. Στην συνέχεια ο εξυπηρετητής αναμένει την παραλαβή του μηνύματος entry. Με την παραλαβή του μηνύματος, αν το πεδίο ονόματος χρήστη είναι None, τότε σημαίνει πως η εγγραφή ήταν ανεπιτυχής, επομένως δεν κάνει καμία άλλη κίνηση. Στην εναλλακτική περίπτωση, ο εξυπηρετητής δέχεται το μήνυμα entry και το διαμοιράζει στους υπόλοιπους αυθεντικοποιητές, ώστε να είναι ενήμεροι. Ως τελικό βήμα, προσθέτει τον νέο χρήστη στην λίστα χρηστών.

Στην περίπτωση της εισαγωγής, ο εξυπηρετητής δέχεται το όνομα χρήστη. Ελέγχει στην τοπική λίστα αν το όνομα χρήστη υπάρχει. Αν υπάρχει τότε διατάζει τον OTP-Server να στείλει τα OTP. Στην συνέχεια ακολουθείται η ίδια διαδικασία με την εγγραφή. Ο εξυπηρετητής επαληθεύει μέσω του OTP τον χρήστη, αλλά στην συνέχεια χρειάζεται και τα στοιχεία του μηχανήματος. Ο χρήστης στέλνει τα στοιχεία του μηχανήματος. Αρχικά ο εξυπηρετητής κάνει τοπικό έλεγχο, και στην συνέχεια διαμοιράζει τα στοιχεία στην υπόλοιπη ομάδα, συγκεντρώνει τις ετυμηγορίες των υπολοίπων σταθμών και εξάγει την τελική απόφαση. Έστω και ένας εξυπηρετητής της ομάδας δεν συμφωνεί με τα στοιχεία, τότε η είσοδος απαγορεύεται. Στην αντίθετη περίπτωση ενημερώνει κατάλληλα τον χρήστη για την είσοδο του στο σύστημα.

```
if rank == 2:
    #   this is the main authenticator server
    while(1):
        msg1 = comm.recv(source=0, tag=11)
        #   check the user option
        if(msg1['option'] == 1):
            #   the user wishes to sign-up
            #   inform the otp server about it
            comm.send(msg1, dest=1, tag=11)
            #   receive the otp
            msg2 = comm.recv(source=1, tag=12)
            local_otp = msg2['OTP']
            #   get the user input
            msg1 = comm.recv(source=0,tag=13)
            #   check the otp
            if(msg1['word'] == local_otp):
                retval['flag'] = 1
                retval['message'] = "User ready to register, send username and computer info"
            else:
                retval['flag'] = 0
                retval['message'] = "Invalid OTP given!!"
            comm.send(retval, dest=0,tag=14)
            entry = comm.recv(source=0,tag=15)
            if(entry['usrname'] is None):
                continue
```

Εικόνα 33: Κώδικας αυθεντικοποιητή - εξυπηρετητή (authentication - server) 1

```
            else:
                #   send the user info to all the other servers
                for i in range(3,size):
                    comm.send(entry, dest=i, tag=16)
                #   update local user registry
                users.append(entry)
        elif(msg1['option'] == 2):
            #   this is the sign-in operation
            entry = comm.recv(source=0,tag=20)
            #   check the username in the list
            flag = 0
            for i in users:
                if i['usrname'] == entry['usrname']:
                    flag = 1
                    break
            #   get the retval message ready
            retval['flag'] = flag
            comm.send(retval, dest=0,tag=21)
            if(flag):
                #   if the user was found then send for the otp
                comm.send(msg1, dest=1, tag=11)
                msg2 = comm.recv(source=1, tag=12)
                local_OTP = msg2['OTP']
                msg1 = comm.recv(source=0,tag=23)
                if msg1['word'] == local_OTP:
                    retval['flag']=1
                    comm.send(retval, dest=0, tag=24)
```

Εικόνα 34: Κώδικας αυθεντικοποιητή - εξυπηρετητή (authentication - server) 2

```
                comm.send(retval, dest=0, tag=24)
                #   receive the information
                entry = comm.recv(source=0, tag=25)
                if(entry['info'] == None):
                    continue
                else:
                    #   check the entry with the other entries in the list
                    retval['flag'] = 0
                    retval['message'] = "Entry forbidden"
                    for i in users:
                        if(i['usrname'] == entry['usrname']) and (i['info'] == entry['info']):
                            retval['flag'] = 1
                            retval['message'] = "Entry permitted"
                            break
                    #   need to wait other flags to return from other machines
                    for i in range(3,size):
                        comm.send(msg1, dest=i, tag=27)
                        comm.send(entry, dest=i, tag=28)
                    acc = retval['flag']
                    for i in range(3,size):
                        retval = comm.recv(source=i, tag=29)
                        print("received")
                        acc *= retval['flag']
                    retval['flag'] = acc
                    comm.send(retval, dest=0, tag=26)
            else:
                continue
```

Εικόνα 35: Κώδικας αυθεντικοποιητή - εξυπηρετητή (authentication - server) 3

6.2.7. Κώδικας Ομάδας Επαληθευτών

Οι επαληθευτές χωρίζουν την λειτουργικότητα τους σε δύο περιπτώσεις. Στην περίπτωση εγγραφής, απλά ενημερώνουν τις τοπικές τους λίστες για τον νέο χρήστη. Στην περίπτωση της εισόδου ενεργοποιούνται από τον βασικό επαληθευτή και εκτελούν ακριβώς τον ίδιο έλεγχο εγκυρότητας. Με την ολοκλήρωση της διαδικασία, κάθε επαληθευτής στέλνει το αποτέλεσμα του στον αρχηγό τους.

```
if rank > 2:
    while(1):
        entry = comm.recv(source=2, tag=16)
        print(rank, entry)
        users.append(entry)
        msg1 = comm.recv(source=2, tag=27)
        #   time for authentication
        entry = comm.recv(source=2, tag=28)
        retval['flag'] = 0
        for i in users:
            if(i['usrname'] == entry['usrname']) and (i['info'] == entry['info']):
                retval['flag'] = 1
                retval['message'] = "Entry permitted"
                break
        if(retval['flag']):
            print("I",rank,"hereby declare you worthy!",entry['usrname'])
        comm.send(retval, dest=2, tag=29)
```

Εικόνα 36: Κώδικας ομάδας αυθεντικοποιητών

6.2.8. Κώδικας OTP-Server

Ο OTP-Server αποτελεί την πιο απλή υλοποίηση εντός του κώδικα. Ενεργοποιείται μόνο από τον επαληθευτή και το μόνο που κάνει είναι να παράγει ένα OTP μέσω της συνάρτησης generate OTP(). Με την παραγωγή του OTP, τότε ο εξυπηρετητής αποστέλλει ένα μήνυμα τύπου msg2 στον πελάτη και τον αρχηγό της επαλήθευσης.

```
if rank == 1:
    #   this is the otp server
    #   only the authenticator server talks with this server
    while(1):
        #   wait for the authenticator to send the message
        msg1 = comm.recv(source=2,tag=11)
        #   fill in the msg2 object
        msg2['OTP'] = generateOTP()
        msg2['len'] = len(msg2['OTP'])
        #   send the otp to the authenticator and the user
        comm.send(msg2, dest=0, tag=12)
        comm.send(msg2, dest=2, tag=12)
```

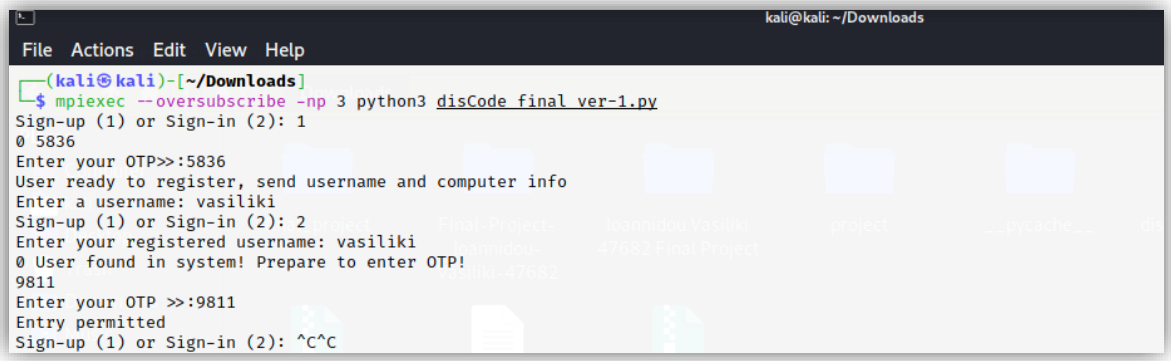
Εικόνα 37: Κώδικας OTP server

6.2.9. Εκτέλεση

Για την εκτέλεση του κώδικα χρειάζεται να τρέξουμε την εντολή:

Mpiexec --oversubscribe -np *number_of_processors* python disCode.py

Με την εκτέλεση της άνωθεν εντολής τότε ο κώδικας καλεί τον χρήστη να εισάγει τις απαραίτητες πληροφορίες. Στιγμιότυπα της εκτέλεσης φαίνονται παρακάτω:



Εικόνα 38: Εκτέλεση κώδικα disCode

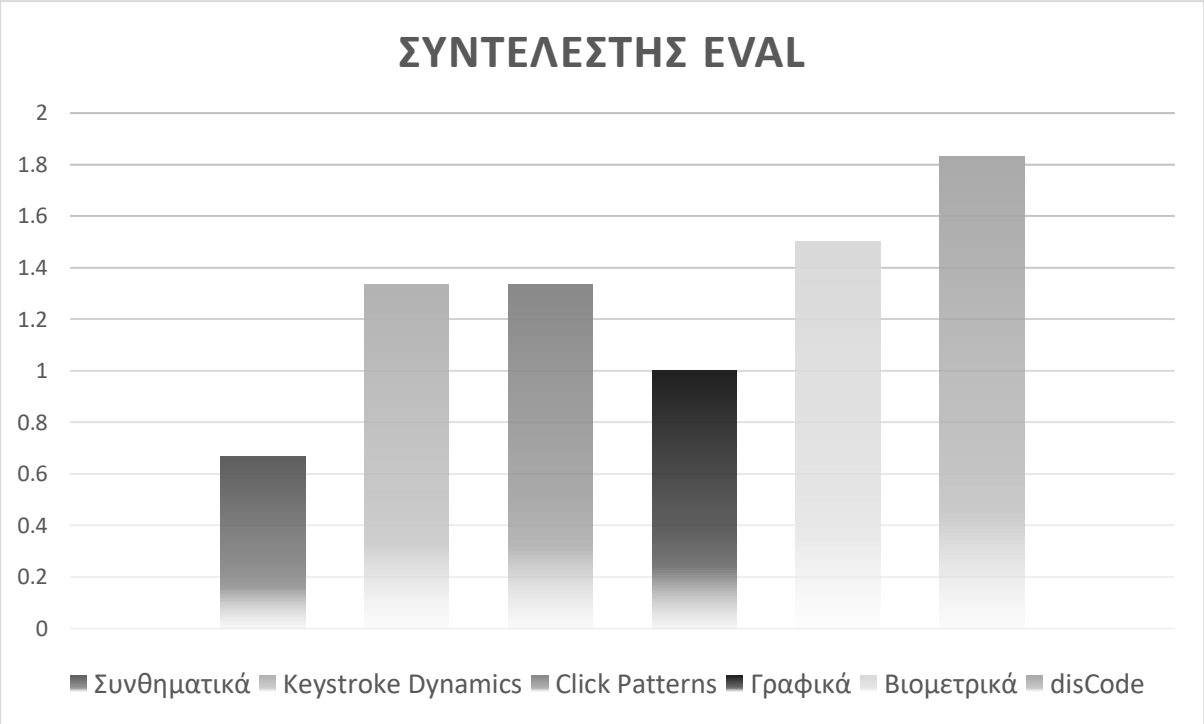
Κεφάλαιο 7: Συμπεράσματα

Με την ολοκλήρωση της υλοποίηση και της εκτέλεσης του κώδικα μπορούμε να διαπιστώσουμε πως υπάρχουν αρκετές μέθοδοι για την επιτυχή εγκατάλειψη των συνθηματικών εισόδου. Ειδικότερα η χρήση της επαλήθευσης μέσω OTP ή βιομετρικά αποτελεί μια ισχυρή εναλλακτική, αλλά ταυτόχρονα, η χρήση μεταδεδομένων μπορεί να ενισχύσει την διαδικασία.

Στην παρούσα εργασία προτείναμε ένα σύστημα επαλήθευσης χρήστη με βάση την πολυϋπολογιστική προσέγγιση. Συγκεκριμένα σχεδιάσαμε ένα σύστημα που θα έδινε πρόσβαση στον χρήστη χωρίς την χρήση συνθηματικών. Το σύστημα μας εκμεταλλεύεται αρχικά την λογική των OTP, αξιοποιώντας έναν εξυπηρετητή OTP για την υποστήριξη της αρχικής επικύρωσης του χρήστη, εν συνεχεία επαληθεύει και επιτρέπει την πρόσβαση στον χρήστη, αφού αυτός έχει πιστοποιηθεί από μια ομάδα εξυπηρετητών. Οι ομάδα εξυπηρετητών δύναται να αποφασίζει σε σύντομο διάστημα για την ταυτότητα και το δικαίωμα πρόσβασης του εκάστοτε χρήστη.

Θετική πτυχή του κώδικα μας είναι η εκμετάλλευση της πολυϋπολογιστικής τεχνολογίας για την επικύρωση του χρήστη, καθώς μας επιτρέπει να λαμβάνουμε αποφάσεις με βάση πολλαπλές αυτόνομες μηχανές. Επομένως ο χρήστης είναι επικυρωμένος ύστερα από βαθύ έλεγχο των χαρακτηριστικών εισόδου, των δεδομένων του υπολογιστικού συστήματος χρήστη. Επιπλέον, το σύστημα είναι ανθεκτικό σε οποιαδήποτε επίθεση συνθηματικού, καθώς αυτά απλά δεν χρησιμοποιούνται. Τέλος, δανειζόμενοι τις ιδιότητες των κατανεμημένων και πολυϋπολογιστικών συστημάτων, η λύση μας μπορεί τόσο να κλιμακωθεί όσο και να επεκταθεί, καθώς μπορεί να ενσωματωθεί σε μεγαλύτερα συστήματα. Η φύση του συστήματος ως κατανεμημένο και πολυϋπολογιστικό το καθιστά και αρκετά γρήγορο. Συλλογικά μπορούμε να αξιολογήσουμε την ποιότητα του συστήματος μας μέσω του συντελεστή Eval όπως αυτός ορίστηκε νωρίτερα. Παρακάτω φαίνεται ο ανανεωμένος πίνακας τιμών του συντελεστή.

Μέθοδος	Ανθεκτικότητα	Ειδικό Υλικό	Κόστος	Ψυχολογική Κατάσταση	Επίπεδο Προστασίας	Χρόνος
Συνθηματικά	Όχι	Όχι	Κανονικό	Όχι	Χαμηλό	Γρήγορο
Keystroke Dynamics	Shoulder Surfing, phishing, keyloggers	Όχι	Κανονικό	Ναι	Μέτριο	Μέτριος
Click Patterns	Shoulder Surfing, phishing, keyloggers	Όχι	Κανονικό	Ναι	Μέτριο	Μέτριος
Γραφικά	Shoulder Surfing	Ναι	Υψηλό	Ναι	Μέτριο	Αργός
Βιομετρικά	Shoulder Surfing, phishing, keyloggers, etc.	Ναι	Υψηλό	Όχι	Υψηλό	Αργός
disCode	Shoulder Surfing, phishing, keylogger, etc.	Ναι	Υψηλό	Όχι	Υψηλό	Μέτριος



Η τελική βαθμολογία του συστήματος disCode υπολογίστηκε στην τιμή 1.83 που είναι υψηλότερη, συγκριτικά με τις υπόλοιπες τιμές. Η τιμή αυτή οφείλεται στην διαφορά ταχύτητος μεταξύ της μεθόδου μας και των βιομετρικών. Στην περίπτωση των βιομετρικών, η αναγνώριση εικόνας και η σύγκριση της για την επιβεβαίωση είναι αρκετά πιο αργή από την συλλογή και αποστολή των δεδομένων μηχανής, αλλά και της απόφασης του συστήματος, καθώς αυτή λαμβάνεται κατανεμημένα. Επιπροσθέτως, το σύστημα μας είναι απόλυτα συμβατό με άλλες μεθόδους επαλήθευσης δίχως συνθηματικά, όπως τα βιομετρικά. Ως εκ τούτου, το σύστημα μπορεί να συνδυαστεί ώστε να επιτευχθεί η μέγιστη δυνατή ασφάλεια.

Φυσικά, καμία εφαρμογή και κανένα σύστημα δεν μπορεί να είναι αλάνθαστο. Για αρχή, το σύστημα μας εξαρτάται από την κατάσταση και την ασφάλεια του δικτύου. Η υλοποίηση μας προϋποθέτει μια υποτυπώδεις ασφάλεια που δεν θέτει σε κίνδυνο τα μεταδιδόμενα δεδομένα. Φυσικά θέματα κρυπτογράφησης και κωδικοποίησης των ροών μπορούν να καλύψουν αυτό το κενό, κάτι που όμως δεν αποτελεί αντικείμενο αυτής της μελέτης. Πιθανό πρόβλημα ίσως αποτελέσει και η κατάρρευση ενός εκ των εξυπηρετητών κατά την διάρκεια της επαλήθευσης, το οποίο βέβαια θεωρείτε γνωστό πρόβλημα στα κατανεμημένα συστήματα, το οποίο αντιμετωπίζεται με αλγόριθμους ευρωστίας (robust algorithms).

Εν κατακλείδι, μπορούμε να παρατηρήσουμε πως παρά τις αδυναμίες, που ίσως εμφανίζονται, αυτές μπορούν να αντιμετωπιστούν, και το σύστημα να συνεχίσει ακάθεκτο. Επομένως, η προσέγγιση μας μπορεί να θεωρηθεί αξιόπιστη και πλήρως συμβατή με το ύφος και τις ιδέες του θέματος μας. Τα αποτελέσματα και τα συμπεράσματα μας οδηγούν πως η μη χρήση των συνθηματικών εισόδου, δεν αποδυναμώνει το σύστημα, αλλά αντιθέτως, το ενδυναμώνει και το θωρακίζει ενάντια σε πληθώρα απειλών.

Βιβλιογραφία

1. Stalling Williams & Brown Lawrie (2015). *Ασφάλεια Υπολογιστών: Αρχές Πρακτικής 3η Αμερικάνικη Έκδοση* (μτφρ Γ. Στάμου και επιμ Π. Αρκουδέας), Αθήνα: Κλειδάριθμος (2016).
2. Καντζάβελου Ιωάννα (2020). *Διαφάνειες Access Control.pptx,* Αθήνα: Πανεπιστήμιο Δυτικής Αττικής
3. Kantzavelou Ioanna, Tzikopoulos F. Panagiotis, and Sokratis K. Katsikas (2013). *Detecting Intrusive Activities from Insiders in a Wireless Sensor Network using Game Theory*. In Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA'13), May 29 - 31, 2013, Island of Rhodes, Greece, ACM.
4. Vassilis Papaspirou, Leandros Maglaras, Mohamed Amine Ferrag, Ioanna Kantzavelou, Helge Janicke, Christos Douligeris (2020). *"A novel Two-Factor HoneyToken Authentication Mechanism"*, December 2020, arXiv:2012.08782v3 [cs.CR].
5. Vassilis Papaspirou, Maria Papathanasaki, Leandros Maglaras, Ioanna Kantzavelou, Christos Douligeris, Mohamed Amine Ferrag and Helge Janicke (2022). *"Security Revisited: Honeytokens meet Google Authenticator"*, 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM 2022), 23-25 September 2022, Ioannina, Greece.
6. OWASP. *Access Control*, https://owasp.org/www-community/Access_Control, 2021
7. [NISTIR 7316] Hu C. Vincent & Ferraiolo F. David & Kuhn D. Rick. *Assessment of Access Control Systems.* https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7316.pdf, Σεπτέμβριος 2006

8. [NISTSP 800-118] Scarfone Karen & Souppaya Murugiah. *Guide to Enterprise Password Management (Draft)*. https://csrc.nist.gov/CSRC/media/Publications/sp/800-118/archive/2009-04-21/documents/draft-sp800-118.pdf, Απρίλιος 2009
9. Piazza Dan. *NIST Password Guidelines in 2020*. https://stealthbits.com/blog/nist-password-guidelines/ , Αύγουστος 2020
10. Cisco. *What is Two Factor Authentication.* https://www.cisco.com/c/en/us/products/security/what-is-two-factor-authentication.html?dtid=osscdc000283
11. Cisco. *What is Multi Factor Authentication.* https://www.cisco.com/c/en/us/products/security/what-is-multi-factor-authentication.html
12. Fruhlinger Josh. *2FA Explained: How to enable it and how it works.* https://www.csoonline.com/article/3239144/2fa-explained-how-to-enable-it-and-how-it-works.html , Σεπτέμβριος 2019
13. Ducklin Paul. *Serious Security: How to store your users' passwords safely.* https://nakedsecurity.sophos.com/2013/11/20/serious-security-how-to-store-your-users-passwords-safely/ , Νοέμβριος 2013
14. Rountree Derrick. *Security for Microsoft Windows System Administrators*. https://www.sciencedirect.com/topics/computer-science/symmetric-key-algorithm , 2011
15. OWASP. *CheatSheets Series Team.* https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html , 2021
16. OKTA. *What are Salted Passwords and Password Hashing?* https://www.okta.com/blog/2019/03/what-are-salted-passwords-and-password-hashing/ , 2022
17. Gary C. Kessler. *Passwords – Strength and Weaknesses.* https://www.garykessler.net/library/password.html , 1997
18. G.S. Jackson. *The Disadvantages of Password Authentication Protocol.* https://itstillworks.com/disadvantages-password-authentication-protocol-2895.html
19. MITRE. *CAPEC-16: Dictionary-based Password Attack.* https://capec.mitre.org/data/definitions/16.html , 2021

20. TechTarget Contributor. *Dictionary Attack.* https://www.techtarget.com/searchsecurity/definition/dictionary-attack , 2021
21. MITRE. *CAPEC-112: Brute Force.* https://capec.mitre.org/data/definitions/112.html , 2021
22. MITRE. *CAPEC-49: Password Brute Forcing.* https://capec.mitre.org/data/definitions/49.html , 2021
23. ITPerfection. *Brute force attack.* https://www.itperfection.com/network-security/brute-force-attack-exhaustive-search-dictionary-password-cybersecurity-network-security/ , 2020
24. KALI. *John The Ripper*. https://www.kali.org/tools/john/
25. OpenWall. *John The Ripper*. https://www.openwall.com/john/doc/EXAMPLES.shtml
26. Motasem Hamdan. *Linux Security – Setting Password Policy.* https://www.youtube.com/watch?v=n8jTahnjhZw&ab_channel=MotasemHamdan
27. Vivek Gite. *Linux check passwords against a dictionary attack.* https://www.cyberciti.biz/tips/linux-check-passwords-against-a-dictionary-attack.html , 2006
28. Kevin Nelson. *Dictionary attack, How to hack a password*. https://dwisesoft.com/hack-a-password-using-dictionary-attack/ , 2020
29. Teju Shyamsundar, OKTA. *What is a One – Time Password (OTP)?* https://www.okta.com/blog/2020/06/what-is-a-one-time-password-otp/ , 2020
30. Jisc Community. *Passwords : Threads and Counter – Measures.* https://community.jisc.ac.uk/library/janet-services-documentation/passwords-threats-and-counter-measures
31. Swaroop Sham, OKTA. *What is a Multi – Factor Authentication (MFA)?* https://www.okta.com/blog/2021/08/multi-factor-authentication-mfa/ , 2021