Etičko hakiranje - alati i primjeri

Šimić, Mihovil

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:921952

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Preddiplomski studij

Etičko hakiranje – alati i primjeri

Završni rad

Mihovil Šimić

Osijek, 2021.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 07.07.2021.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime studenta:	Mihovil Šimić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4279, 26.07.2018.
OIB studenta:	48525482169
Mentor:	Izv. prof. dr. sc. Krešimir Grgić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Etičko hakiranje - alati i primjeri
Znanstvena grana rada:	**Telekomunikacije i informatika (zn. polje elektrotehnika)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	07.07.2021.
Datum potvrde ocjene Odbora:	14.07.2021.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 09.09.2021.

Ime i prezime studenta:	Mihovil Šimić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4279, 26.07.2018.
Turnitin podudaranje [%]:	5%

Ovom izjavom izjavljujem da je rad pod nazivom**: Etičko hakiranje - alati i primjeri**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Krešimir Grgić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD *1*

1.1. Zadatak završnog rada **1**

2. RAZLIKE IZMEĐU ETIČKOG I NEETIČKOG HAKIRANJA *2*

2.1. Etičko hakiranje **2**

2.1.1. Primjer etičkog hakiranja 2

2.2. Neetičko hakiranje **3**

2.2.1. Primjer neetičkog hakiranja 4

3. ALATI I OPERACIJSKI SUSTAV *5*

3.1. Kali Linux **5**

3.2. Alati za testiranje snage i oporavak lozinke **7**

3.2.1. John the Ripper 8

3.2.2. Hashcat 11

3.3. Alati za testiranje sigurnosti mrežnog prometa **13**

3.3.1. Nmap 13

3.3.2. Wireshark 16

3.4. Alati za testiranje sigurnosti Internet stranica **18**

3.4.1. Burp Suite 19

3.5. Alati za testiranje sigurnosti Wi-Fi mreža **21**

3.5.1. Aircrack-ng 22

3.6. Alati za prikupljanje informacija o meti **26**

3.6.1. theHarvester 26

3.6.2. Osintgram 29

4. UREĐAJI *30*

5. FAZE ETIČKOG HAKIRANJA *37*

5.1. Izviđanje **37**

5.2. Skeniranje **38**

5.3. Stjecanje pristupa **40**

5.4. Održavanje pristupa **42**

5.5. Pokrivanje tragova **43**

6. ZAKLJUČAK.....*46*

LITERATURA.....*47*

POPIS SLIKA.....*50*

SAŽETAK.....*52*

ABSTRACT.....*53*

1. UVOD

U današnjem više nego ikada umreženom svijetu mnogi se korisnici oslanjaju na digitalne usluge kako bi olakšali obavljanje svakodnevnih zadatka. Pri tome koriste razne aplikacije i Internet stranice bez razmišljanja o sigurnosti istih. Izlaganjem svojih osobnih informacija na društvenim mrežama i bezobzirnim postupcima tijekom pretraživanja Internet stranica svakodnevno se izlažu riziku kibernetičkog napada, poput krađe broja kreditne kartice, instaliranja virusa na svoje računalo i slično. Stopa kibernetičkog kriminala rapidno raste zbog sve većeg broja uređaja spojenih na Internet i svakodnevnih korisnika Internet usluga. Prelazak na mrežni način rada postaje gotovo neizbježan za svaku ozbiljnu tvrtku sa visokim brojem klijenata te zbog velike baze podataka korisnika, kibernetički napad postaje najveća prijetnja za svaku tvrtku. Svakim danom se povećava broj pronađenih sigurnosnih propusta Internet stranica, što dovodi u rizik osobne podatke svih korisnika stranica. Etički hakeri imaju zadatak otkrivanja sigurnosnih propusta Internet stranica prije kibernetičkih kriminalaca kako bi tvrtke uspješno sačuvale sigurnost svojih klijenata i vlastiti kredibilitet. Osim Internet stranica etički hakeri rade i na očuvanju sigurnosti mobilnih aplikacija, operacijskih sustava, računalnog sklopovlja, Interneta objekata te *blockchain* tehnologija. Sigurnost korisnika pri korištenju digitalnih usluga od velike je važnosti stoga je završni rad usredotočen na alate i tehnike koje pomažu pri održavanju iste.

U prvom dijelu rada objašnjena je razlika između etičkog i neetičkog hakiranja te su opisani i objašnjeni primjeri za svaki tip hakiranja. Najčešće korišteni alati te primjeri njihova korištenja pri ispitivanju sigurnosti sustava opisani su u drugom dijelu rada. U trećem dijelu opisani su najčešće korišteni uređaji. Četvrti dio definira faze pri etičkom hakiranju te se za svaku fazu opisuje i objašnjava kojim metodama, alatima i uređajima se dolazi do željenog rezultata.

1.1. Zadatak završnog rada

U završnom radu potrebno je detaljno opisati i objasniti pojam etičkog hakiranja, te istaknuti i objasniti razlike u odnosu na "neetičko" hakiranje. Opisati neke najčešće alate i postupke koji se pri tome koriste, te ilustrirati primjerima realnih scenarija praktične primjene.

2. RAZLIKE IZMEĐU ETIČKOG I NEETIČKOG HAKIRANJA

2.1. Etičko hakiranje

Iako zvuči kao oksimoron, etičko hakiranje poprima sve veću važnost zbog brzo rastuće digitalizacije društva. Tvrtke diljem svijeta angažiraju etičke hakere kako bi što bolje zaštitile svoje klijente i njihove osobne podatke. Etički hakeri imaju zadatak probijanja sigurnosne zaštite sustava ili mreže korištenjem specijaliziranih alata, tehnika i znanja o prijašnjim sigurnosnim propustima u svrhu stjecanja informacija o sigurnosnim propustima sustava tvrtke, koja ih je angažirala. Sigurnosni propust predstavlja slabost sustava koju haker može iskoristiti u svrhu dobivanja neautoriziranog pristupa računalnom sustavu ili mreži. Sigurnosni propusti mogu nastati zbog više različitih razloga poput mana operacijskog sustava, pogrešno konfiguriranih servera s bazom podataka, loše enkripcije podataka, nedovoljne provjere unosa korisničkih podataka te nedostatka podučavanja radnika o mogućim prijetnjama. Etički hakeri pomoću specijaliziranih alata za pojedini sigurnosni propust pokušavaju otkriti postoji li isti u sustavu, te ako postoji potrebno ga je ukloniti i objasniti kako i zašto se ubuduće isti treba izbjegavati. Jedan od često korištenih načina za upad u računalni sustav tvrtke je društveni inženjering (engl. *social engineering*). Pojam društvenog inženjeringa opisuje tehnike manipulacije ljudima u svrhu stjecanja informacija o tvrtki koje bi olakšale upad u sustav iste. Društveni inženjering predstavlja veliku prijetnju za svaku tvrtku ako se ne poduzmu mjere za podučavanje radnika o opasnosti, prepoznavanju i izbjegavanju istoga. Informacije stečene pomoću društvenog inženjeringa mogu dovesti do saznanja o drugim slabim točkama sustava. Kako bi izbjegle neautorizirane upade putem grešaka u računalnom sustavu, mnoge tvrtke nude program nagrada etičkim hakerima za pronalazak istih (engl. *bug bounty*). Tvrtka Microsoft je spremna platiti 250 000 dolara za pronađene greške u Microsoft Hyper-V-u, popularnom programu za korištenje virtualnih uređaja. Nagradu je moguće ostvariti pronalaskom određenih sigurnosnih propusta koji bi hakerima omogućili otkrivanje privatnih informacija korisnika, daljinsko izvršavanje koda ili uskraćivanje usluga [1]. Na temelju prethodno navedenih tvrdnji i primjera vidljivo je da etičko hakiranje igra veliku ulogu u zaštiti sigurnosti i privatnosti korisnika, te postaje neophodno za uspješan razvoj Internet poslovanja svake tvrtke.

2.1.1. Primjer etičkog hakiranja

Zbog svih događaja u svijetu koji su potaknuli prelazak na mrežni način rada, razmjena informacija putem video konferencije postala je svakodnevica za mnoge tvrtke. Platforma Zoom postala je jedna od najkorištenijih platformi za video konferenciju u kratkom roku, stoga je pitanje sigurnosti

iste prešlo u prvi plan. Istraživanjem sigurnosti platforme, Patrick Wardle i Felix Seele pronašli su dva ozbiljna sigurnosna propusta koji su uvelike ugrožavali sigurnost korisnika Mac operacijskog sustava pri korištenju Zoom-a. Prvi propust bi hakeru omogućio administratorski pristup korisnikovom računalu, odnosno haker bi imao pristup svim podatcima i informacijama na korisnikovom računalu. Drugi propust bi omogućio hakeru snimanje Zoom sastanka te pristup mikrofonu i kameri u bilo kojem trenutku bez znanja korisnika. Oba propusta predstavljaju veliku prijetnju za korisnike platforme, te su Wardle i Seele odmah nakon objave navedenih propusta savjetovali svima ako cijene svoju privatnost i sigurnost da prestanu koristiti Zoom [2].

2.2. Neetičko hakiranje

Glavna razlika između etičkog i neetičkog hakiranja je ovlaštenje. Neetički hakeri nemaju ovlaštenje za probijanje sigurnosne zaštite i pristup sustavu, te se iz tog razloga svaki neautorizirani upad u sustav vodi kao kibernetički zločin. Neetički hakeri uglavnom koriste identične alate i tehnike kao i etički hakeri pri probijanju u sustav. Razlozi neautoriziranih upada u sustav su krađa i prodaja podataka o korisnicima, oštećenje sustava u svrhu nemogućnosti rada, otkrivanje osjetljivih informacija tvrtke i slično. Upadi u sustav se najčešće ostvaruju pomoću društvenog inženjeringa jer tvrtke uvijek imaju dosta informacija o kontaktima radnika dostupnih na Internet stranici. Nakon slanja e-mail poruke hakeri čekaju trenutak kada će radnik kliknuti na link ili preuzeti datoteku sadržanu u poruci, kako bi dobili pristup sustavu. Nakon dobivenog pristupa mogu instalirati virus na računalo, pristupiti osobnim podatcima, blokirati pristup Internetu i slično. Društveni inženjering je ponekad i jedina opcija za neautorizirani upad ako se radi o tvrtki s visokim stupnjem kibernetičke sigurnosti. Tehnike društvenog inženjeringa su slanje e-mail poruka radniku tvrtke na temelju njegovih hobija i sklonosti kako bi ga navele na odlazak na određenu Internet stranicu ili preuzimanje datoteke u sadržaju poruke čime bi hakeri ostvarili pristup sustavu tvrtke, te pokušaj ostvarivanja pristupa sustavu putem USB memorijskog štapića koji je prethodno metodama manipulacije došao u posjed radnika tvrtke. Tvrtke koje ne provode redovna ažuriranja sustava predstavljaju laku metu za hakere, jer postoji mogućnost da će hakeri iskoristiti sigurnosni propust koji bi bio zakrpan posljednjim ažuriranjem. Sustavi bolnica su često zastarjeli i nesigurni, te su time česta meta za hakere. Jedan od primjera hakiranja sustava bolnice bio je WannaCry virus, koji je u kratkom roku zahvatio preko 200 000 računala diljem 150 zemalja. WannaCry je računalni crv, što znači da šalje svoje kopije na druga računala i ovim putem ima mogućnost zaraze velikog broja sustava u kratkom roku. Virus je zaključao računalo korisnika, kriptirao podatke, te ako nije uplaćena određena količina novca svi podatci bi bili obrisani. WannaCry je zaustavljen prije negoli je učinjena velika šteta [3].

2.2.1. Primjer neetičkog hakiranja

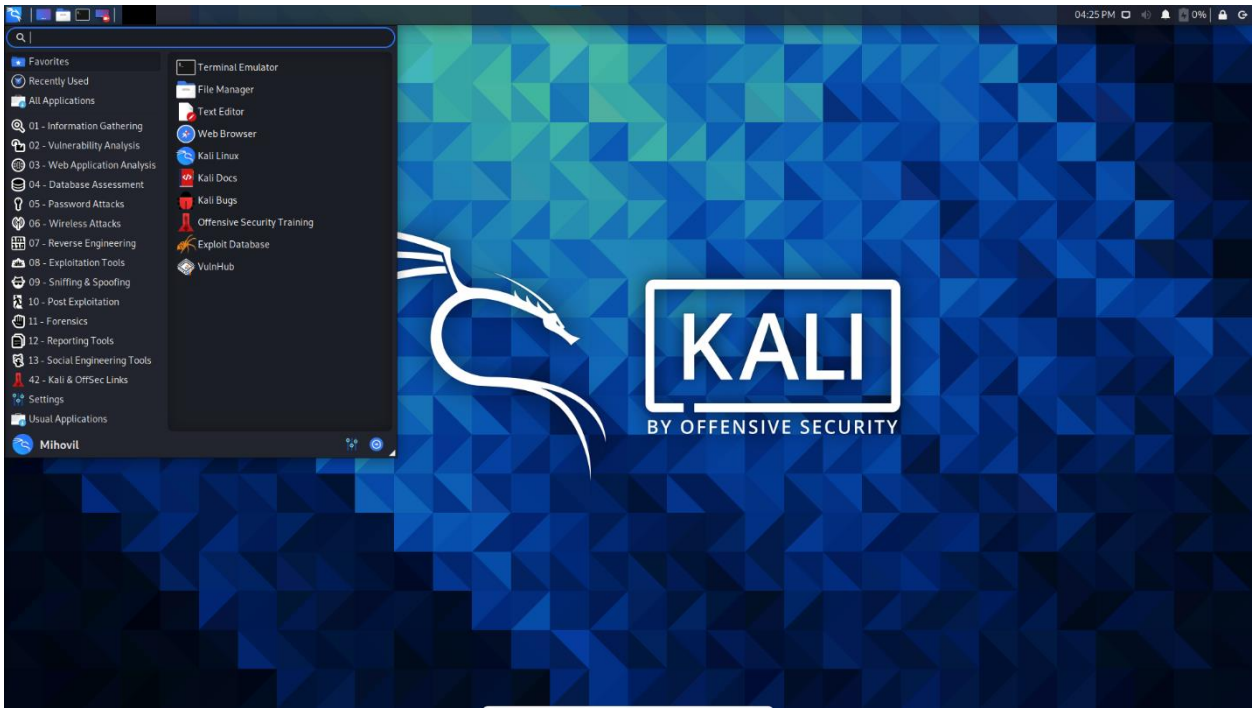
Grupa hakera je 2009. godine izvela jednu on najvećih krađa podataka kreditnih kartica ikada. Ukrali su podatke od 130 000 000 kreditnih kartica koristeći napad SQL ubrizgavanja. U napadu ubrizgavanja, haker unosi nepovjerljive ulazne podatke u program, te ih prevoditelj programa prevodi kao naredbe. Napadi ubrizgavanja su jedni od najstarijih i među najopasnijim hakerskim napadima, a moguće ih je izvesti zbog nedovoljne provjere unesenih podataka korisnika. SQL ubrizgavanje omogućuje korištenje naredbi SQL programskog jezika koje omogućuju kontroliranje servera s bazom podataka Internet stranice. Hakeri su nakon upada u sustav koristili programe za praćenje i analizu mrežnog prometa, te su na takav način uspjeli saznati informacije o kreditnim karticama koje su kasnije prodavali diljem svijeta i uzrokovali visoku stopu krađe identiteta korisnika. Svi hakeri iz grupe su uhićeni i osuđeni za kibernetički zločin [4].

3. ALATI I OPERACIJSKI SUSTAV

Etički hakeri koriste mnoge alate koji olakšavaju i poboljšavaju sigurnosno testiranje sustava. Trenutno postoji dosta besplatnih distribucija Linux-a koje su otvorenog koda te sadrže sve najpotrebnije alate za sigurnosno testiranje. Linux distribucija je gotovi operacijski sustav temeljen na Linux jezgri, a jezgra predstavlja dio softvera koji korisniku omogućuje upravljanje računalnim sklopovljem [5]. Među najpopularnijima distribucijama su Kali Linux, Parrot Security Edition, BlackArch Linux te BackBox Linux. Moj osobni odabir je Kali Linux jer je najkorišteniji, te su primjeri korištenja svih narednih alata odrađeni pomoću njega.

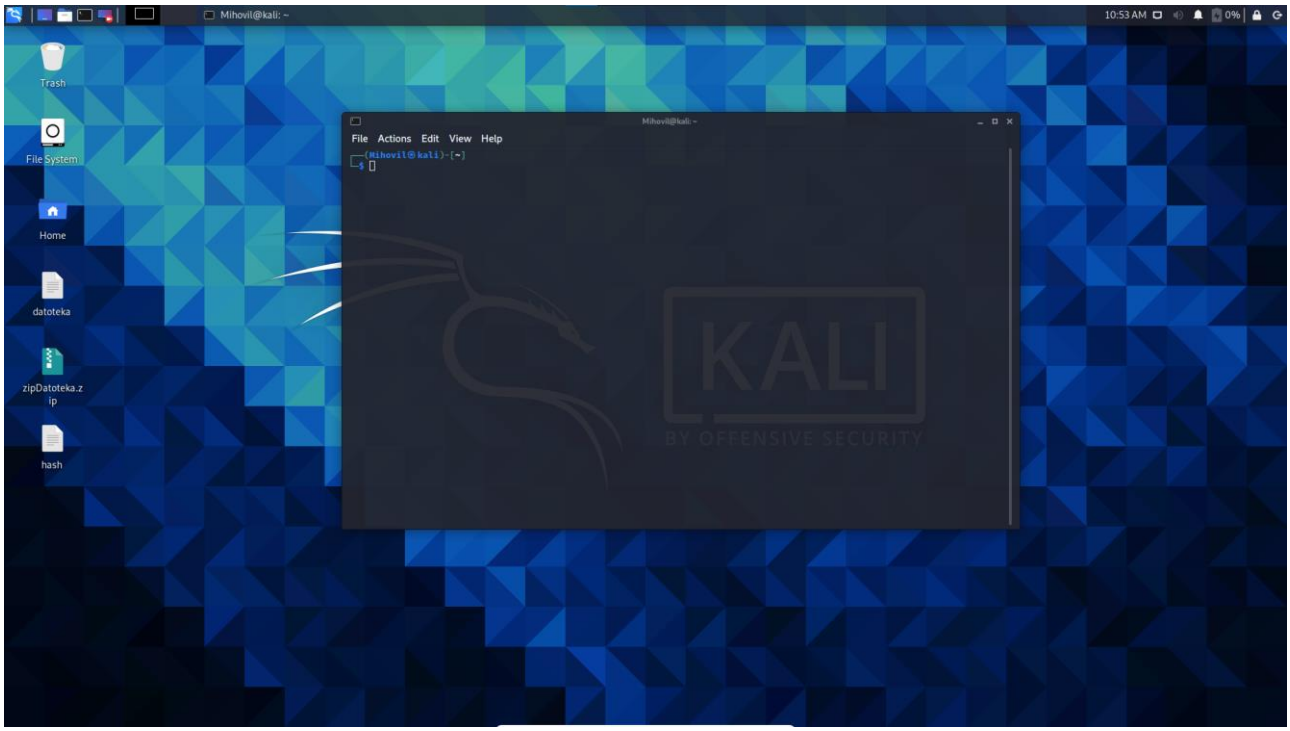
3.1. Kali Linux

Razvijen od strane tvrtke Offensive Security, Kali Linux temeljen je na Debian distribuciji Linuxa. Sadrži preko šesto alata za etičko hakiranje koji služe u svrhu testiranja sigurnosti web aplikacija, skupljanja informacija, testiranja sigurnosti bežičnih mreža, oporavak lozinki, forenziku, obrnutog inženjerstva, analize ranjivosti sustava, društvenog inženjerstva, te analizu sigurnosti baze podataka. Pošto je namijenjen ispunjavanju potreba i zahtjeva etičkih hakera u operacijski sustav implementirano je nekoliko temeljnih promjena koje ih odražavaju. Mrežne i bluetooth usluge onemogućene su prema zadanim postavkama kako bi se što bolje osigurala sigurnost korisnika. Prilagođena Linux jezgra zakrpana je za obranu od bežičnog ubrizgavanja, te je dostupan minimalan i pouzdan skup direktorija. Također postoji besplatna mobilna platforma otvorenog koda za sigurno testiranje, temeljena na Kali Linux-u, imena Kali NetHunter [5].



Slika 1. Radna površina i glavni izbornik Kali Linux-a

Na slici 1. vidljivi su prečaci u gornjem lijevom kutu, preko kojih je moguće pristupiti svim dokumentima te naredbenom retku. Pojam naredbeni redak definira se kao tekstualno sučelje koje omogućuje unos naredbi, njihovo izvršavanje te pregled rezultata. Tumač naredbenog retka koji rukuje unesenim naredbama te ih izvršava naziva se ljuska (engl. *shell*) [5]. Od 19. studenog 2020. godine zadana ljuska za Kali Linux operacijski sustav postaje ZSH, za razliku od prethodne Bash (engl. *Bourne Again Shell*) ljuske [6].



Slika 2. Sučelje naredbenog retka

Izgled sučelja naredbenog retka prikazan je na slici 2.

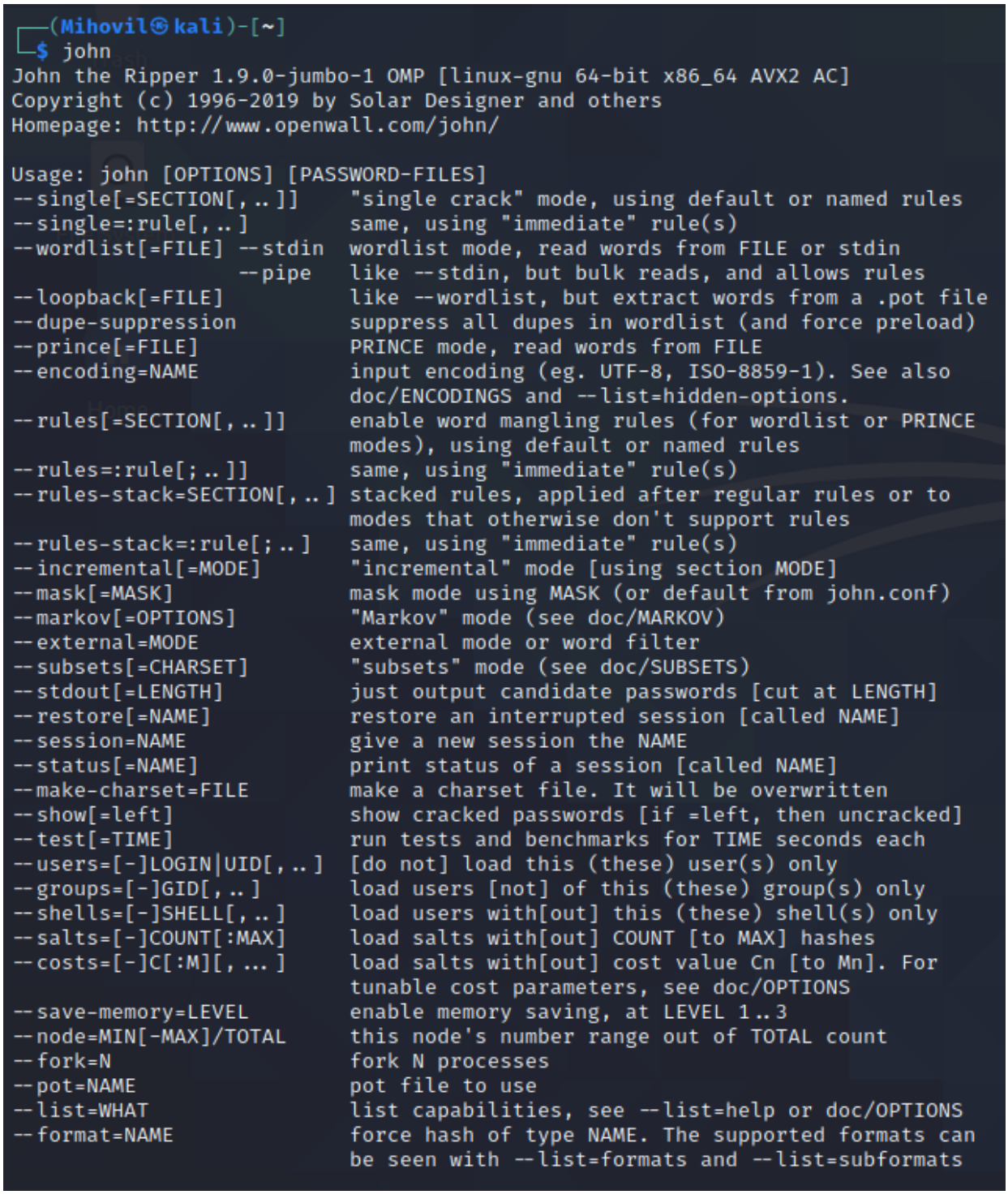
3.2. Alati za testiranje snage i oporavak lozinke

Kada je u pitanju sigurnost korisnika slabe i nesigurne lozinke predstavljaju jedan od najlakših načina hakerima za krađu podataka. Nesigurne lozinke su one koje su kratke, koriste uzastopan niz slova ili brojeva na tipkovnici ili se sastoje od često korištenih riječi bez upotrebe numeričkih znakova. Osnovni načini pohranjivanja lozinki su u obliku običnog tekstualnog dokumenta bez ikakvog oblika zaštite, u obliku kriptiranog tekstualnog dokumenta i u obliku *hash* tekstualnog dokumenta. Pohranjivanje u obliku običnog tekstualnog dokumenta bez ikakvog oblika zaštite je najjednostavniji i ujedno najnesigurniji način pohranjivanja. Zbog ovakvog načina pohranjivanja podatci od preko 32 000 000 korisnika usluga tvrtke RockYou ukradeni su tijekom hakerskog napada 2009. godine. Kod pohranjivanja u obliku kriptiranog tekstualnog dokumenta sve lozinke su zaštićene pomoću ključa bez kojeg se ne može pristupiti lozinkama, što predstavlja veći stupanj sigurnosti, ali u slučaju otkrivanja ključa hakeri će moći pristupiti svim lozinkama. Najbolji način pohranjivanja je pomoću *hash* funkcije. *Hash* funkcija je jednosmjerna funkcija koja pretvara niz znakova, u ovom slučaju lozinku, u drugačiji oblik niza uvijek jednakog broja znakova, koji je puno duži, kompliciraniji te nema nikakve direktne povezanosti sa znakovima sadržanim u lozinki. Funkcija je jednosmjerna jer nije moguće saznati lozinku iz *hash* vrijednosti. Iako je mnogo sigurnija od prethodnih metoda pohranjivanja, ova metoda također ima mane. Ako hakeri ostvare

pristup generiranim *hash* vrijednostima lozinki, izračunavanjem svake *hash* vrijednosti za svaku lozinku iz rječnika lozinki te uspoređivanjem istih, mogu saznati lozinke [7]. Rječnik lozinki je lista lozinki u tekstualnom dokumentu, koja sadrži sve lozinke korisnika koje su hakeri saznali prilikom krađe podataka. Jedna od najpopularnijih lista lozinki je od prethodno spomenute tvrtke RockYou, koja sadrži preko 14 000 000 jedinstvenih lozinki.

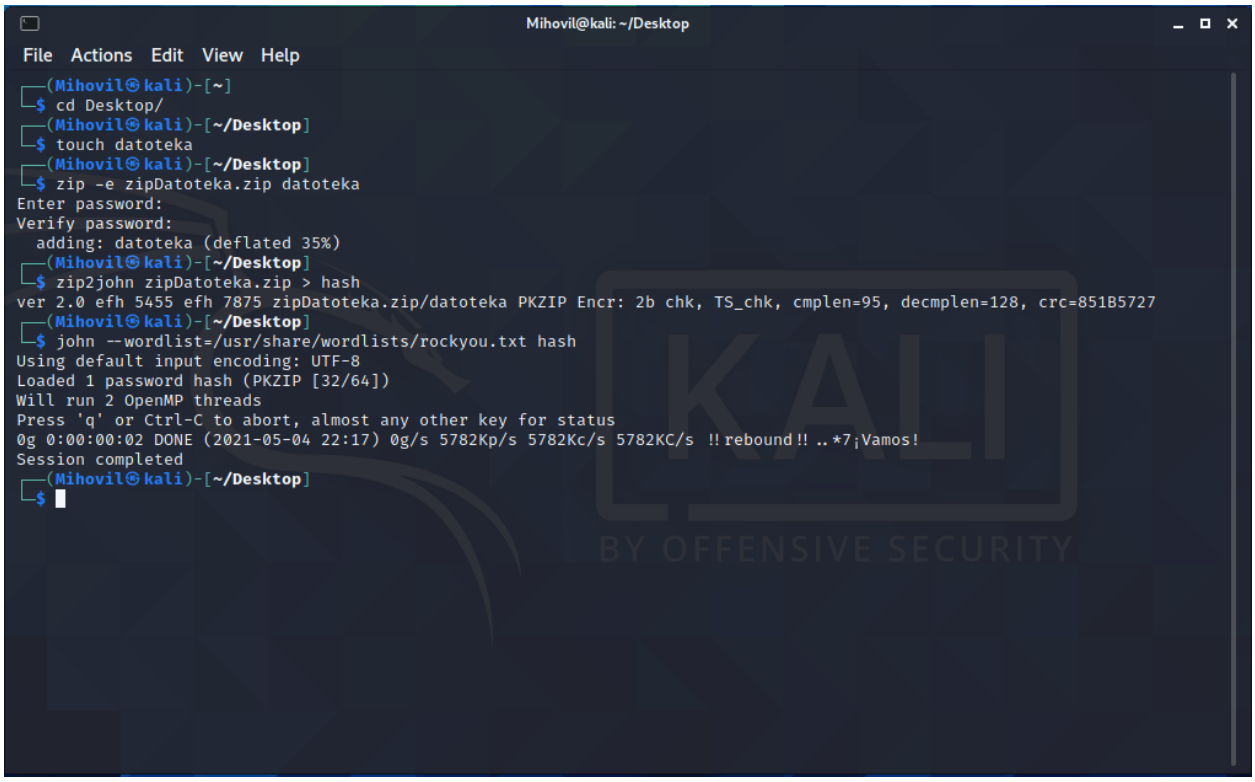
3.2.1. John the Ripper

Jedan od najpopularnijih alata za ispitivanje snage i oporavak lozinke je John the Ripper. Alat je otvorenog koda i dostupan je za mnoge operacijske sustave. Mogući načini probijanja lozinke su pojedinačni, inkrementalni, vanjski i pomoću liste riječi. Pojedinačni način koristi se za probijanje sigurnosti lozinke jednog korisnika jer upotrebljava korisničko ime, ime i prezime korisnika te velik broj pravila za rastavljanje navedenih riječi. Napad pomoću liste riječi je najjednostavniji način jer je jedino potrebno definirati listu riječi te po mogućnosti i potrebi pravila za rastavljanje riječi u listi. Inkrementalni način pokušava sve moguće kombinacije znakova, čime je ujedno i najmoćniji način. Pošto pokušava sve moguće kombinacije, uvijek postoji mogućnost da inkrementalni način nikada neće završiti jer je broj mogućih kombinacija prevelik. Kod vanjskog načina alat koristi funkcije koje je korisnik prethodno definirao [8].



Slika 3. Ispis nakon upisane naredbe *john*

Pokretanje alata izvršava se upisivanjem naredbe *john* bez dodatnih argumenata u sučelje naredbenog retka, te se prikazuje sažetak upotrebe alata koji je prikazan na slici 3. U sljedećem primjeru su pokazani i objašnjeni svi koraci u napadu s listom riječi.



Slika 4. Redoslijed naredbi pri napadu listom riječi

U primjeru na slici 4. prikazano je probijanje zaštite zip dokumenta u svrhu otkrivanja lozinke. Upisivanjem naredbe *touch* kreirana je tekstualna datoteka imena *datoteka*. Pomoću *zip* naredbe popraćene zastavicom *-e* kreirana je lozinkom zaštićena zip mapa imena *zipDatoteka*, u koju je spremljena prethodno stvorena tekstualna datoteka. Upisana lozinka je **7¡Vamos!* jer se ista nalazi u listi s lozinkama rockyou.txt. Naredba *zip2john* služi u svrhu otkrivanja *hash* vrijednosti lozinke zip mape, te je ista prikazana na slici 5. U poglavlju 3.1. opisan je izvor lozinki rockyou.txt datoteke, te je opisan pojam *hash* vrijednosti. Dodavanjem argumenta > preusmjeren je ispis naredbe *zip2john* u tekstualnu datoteku *hash* kako bi se olakšao daljnji rad s *hash* vrijednosti. Datoteka rockyou.txt nalazi se u wordlists direktoriju, a putanja do istoga je */usr/share/wordlists/rockyou.txt*. Upisivanjem naredbe *john* i argumenta *--wordlist* s putanjom do rockyou.txt datoteke, te upisivanjem imena datoteke, s *hash* vrijednosti lozinke zip mape, imena *hash*, započeto je otkrivanje lozinke. Iz ispisa je vidljivo kako je John the Ripper uspio pronaći navedenu lozinku u 2 sekunde. Napadi s listom riječi su jako brzi i efikasni ako se lozinka nalazi u listi.

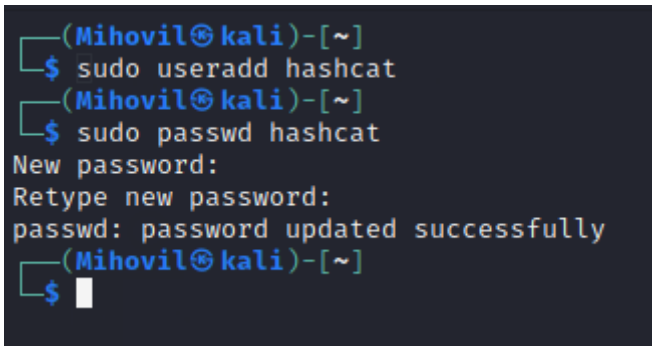
Slika 5. Kreirana *hash* vrijednosti pomoću naredbe *zip2john*

3.2.2. Hashcat

Hashcat je besplatni alat otvorenog koda za oporavak lozinke, dostupan je za Linux, Windows te Mac operacijske sustave, te se smatra jednim od najbržih i najboljih alata za navedenu svrhu. Alat podržava 5 jedinstvenih načina oporavka lozinke za preko 300 *hash* algoritama [9]. Prema [9] podržani načini su:

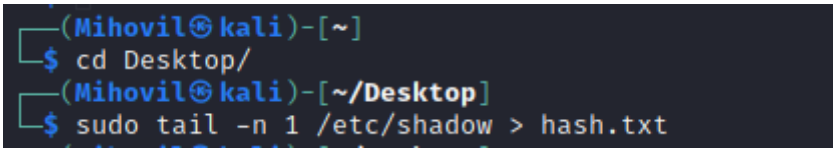
- Napad rječnikom, kod kojeg se pokušavaju sve riječi iz definirane liste riječi
- Kombinirani napad, kod kojeg se spajaju riječi iz više definiranih listi riječi
- Napad sirove snage, gdje se pokušavaju sve kombinacije iz definiranog niza znakova, što je ujedno i najjednostavniji napad
- Maskirani napad, koji je isti kao napad sirove snage, ali se uže definira niz znakova
- Hibridni napad, kod kojeg se spajaju lista riječi i niz znakova
- Napad udruživanja, gdje se definira korisničko ime, ime datoteke ili bilo koji drugi oblik informacije koji može imati utjecaj na otkrivanje lozinke

U idućem primjeru prikazan je oporavak lozinke korisnika sustava koristeći napad rječnikom. U svrhu primjera kreiran je novi korisnik pomoću funkcije *useradd* imena hashcat, te mu je dodijeljena lozinka *hashcat* koristeći funkciju *passwd*, što je prikazano na slici 6.



Slika 6. Dodavanje novog korisnika u sustav

Slika 7. prikazuje prvi korak kod oporavka lozinke, koji je pronalazak *hash* vrijednosti lozinke koja je nalazi u *etc* mapi u *shadow* datoteci.



Slika 7. Stvaranje datoteke s *hash* vrijednosti lozinke

Slika 8. prikazuje redoslijed naredbi te zastavica pri oporavku lozinke. Pošto je lozinka zadnje dodanog korisnika u sustav posljednja na popisu, pomoću naredbi *tail -n 1 /etc/shadow* ispis je

postavljen na zadnji redak u datoteci, te je isti pomoću > usmjeren na *hash.txt* datoteku. Nakon stvaranja datoteke s *hash* vrijednosti, sljedeći korak je otkrivanje lozinke iz navedene vrijednosti.

Slika 8. Naredbe i ispis nakon napada rječnikom

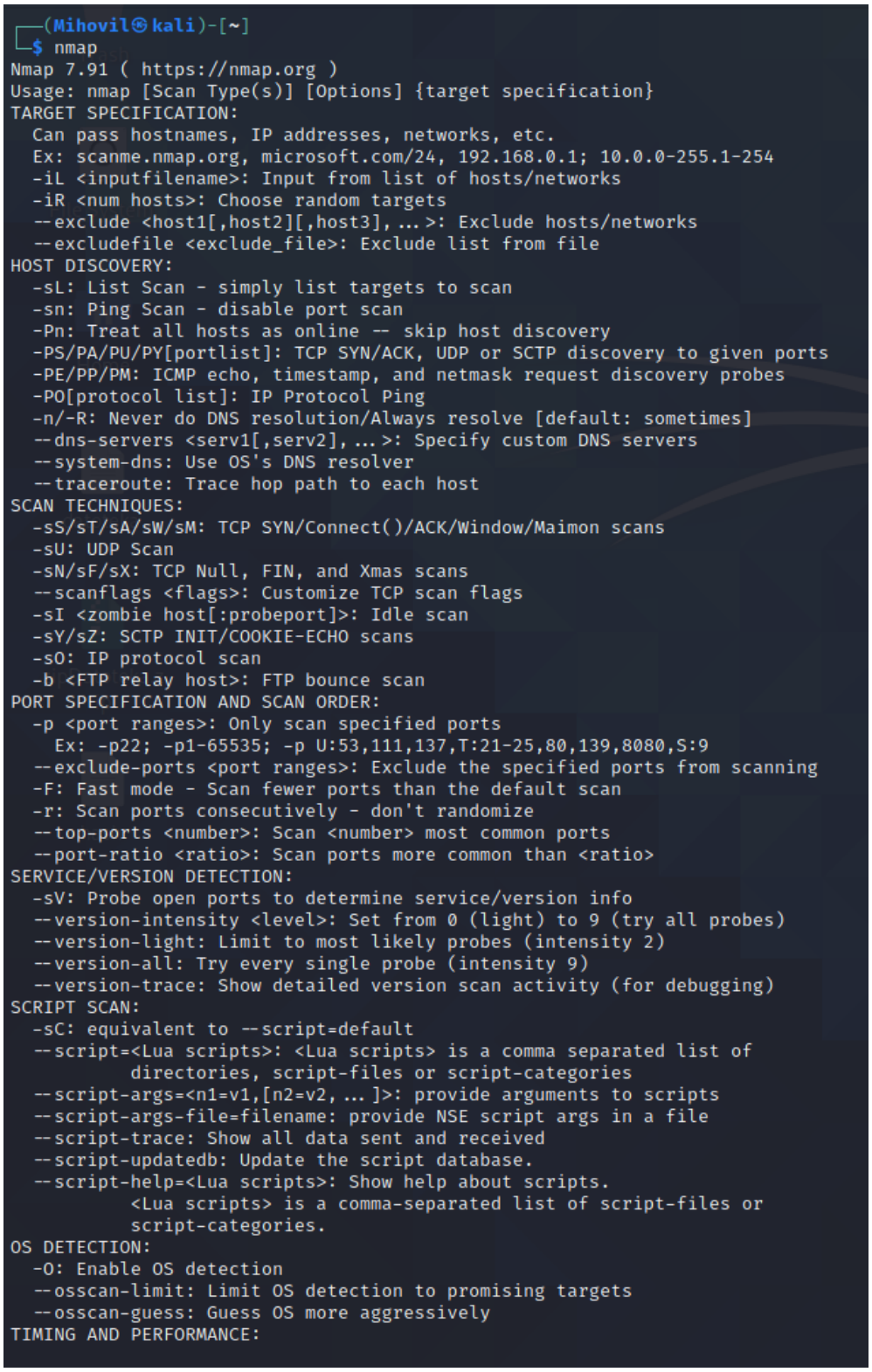
Pomoću naredbe *hashcat* pokrenut je alat, te je zastavicom *-a* definiran tip napada, koji je u ovom slučaju nula, to jest napad rječnikom. Nadalje, za korištenje kombiniranog napada potrebno je upisati broj 1, za napad sirove snage broj 3, za hibridni napad broj 6 ili 7, ovisno o potrebi. Za definiranje *hash* algoritma korištena je zastavica *-m*, te je popraćena brojem 1800 jer isti definira SHA-512 algoritam koji je korišten za šifriranje lozinke. Nadalje se definira datoteka u kojoj se nalazi *hash* vrijednost, što je u ovom slučaju *hash.txt*, te je na kraju dodan put do liste riječi rockyou.txt koji glasi */usr/share/wordlists/rockyou.txt*. Iz ispisa je vidljivo kako je alat pronašao lozinku na temelju predane *hash* vrijednosti unutar jedne sekunde.

3.3. Alati za testiranje sigurnosti mrežnog prometa

Pojam računalne mreže odnosi se na povezane računalne uređaje poput osobnih računala, servera, pametnih telefona i uređaja Interneta objekata kao što su kamere, pametni hladnjaci, zvučni sustavi i slično, koji međusobno komuniciraju. Podatci poslani putem mreže raspodjeljuju se u pakete, manje segmente podatka, koje odredišni uređaj ponovno sastavlja u originalni podatak. Mrežni paket uvijek potiče iz jednog mrežnog sučelja, to jest pošiljatelja, i uglavnom se šalje na jedno mrežno sučelje, koje se naziva odredište. Izraz mrežni promet opisuje pakete koji putuju mrežom u bilo kojem trenutku [10].

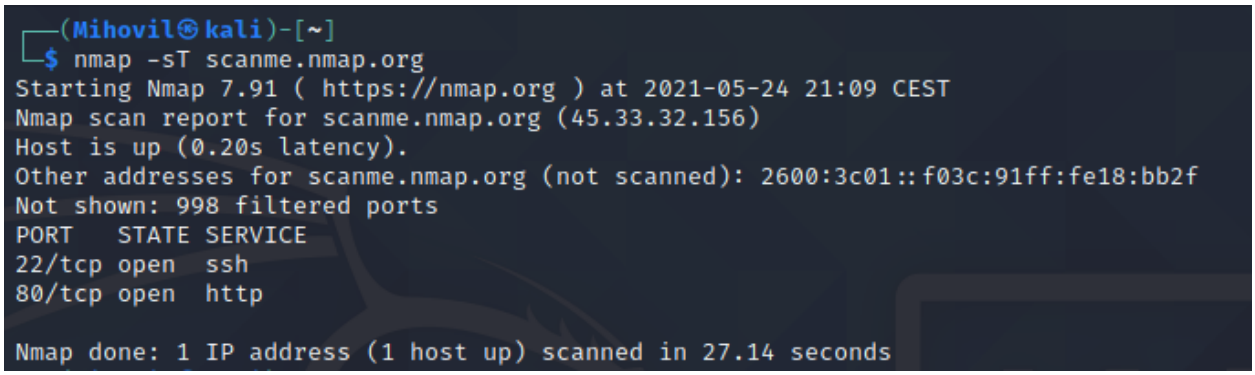
3.3.1. Nmap

Network Mapper je alat otvorenog koda za proučavanje i ispitivanje sigurnosti mreža dostupan za mnoge operacijske sustave. Gordon Lyon napisao je i objavio prvu verziju alata 1997. godine, a posljednja verzija je rezultat grupnog rada mnogih programera koji su doprinijeli razvoju Nmap-a. Pruža mogućnost skeniranja od jednog domaćina do skeniranja mreža s velikim brojem domaćina. Nmap koristi IP pakete u svrhu utvrđivanja koji su domaćini dostupni na mreži, koje usluge domaćini nude, koje operacijske sustave koriste, koji tipovi vatrozida se koriste te mnoge druge karakteristike. Nmap koriste i drugi stručnjaci van područja etičkog hakiranja poput mrežnih administratora za upravljanje rasporedom nadogradnje usluga. Nmap uključuje i Zenmap, grafičko korisničko sučelje za pregled rezultata, Ndiff za uspoređivanje rezultata skeniranja, Ncat za prijenos i preusmjerivanje podataka te Nping za generiranje paketa i analizu odgovora [11].



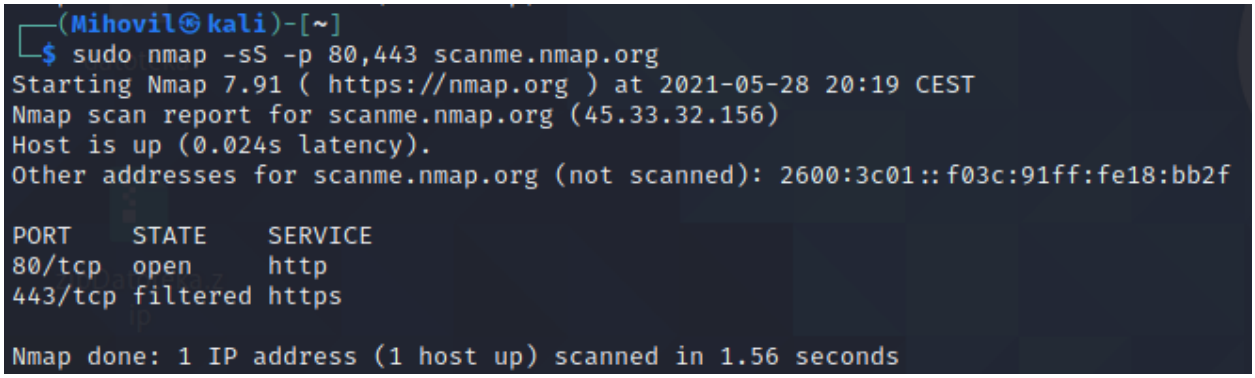
Slika 9. Ispis nakon upisane naredbe *nmap*

Pokretanje alata izvršava se upisivanjem naredbe *nmap* bez dodatnih argumenata u sučelje naredbenog retka, te se prikazuje sažetak upotrebe alata prikazan na slici 9. Na slici 10. pokazani su i objašnjeni svi koraci prilikom skeniranja mete. Za primjer skeniranja korištena je stranica scanme.nmap.org., koja je stvorena iz razloga testiranja Nmap alata od strane svih korisnika.



Slika 10. TCP skeniranje

Pomoću *-sT* zastavice pokrenuto je TCP skeniranje. Iz slike 10. je vidljivo kako je skeniranje dovršeno u 27.14 sekundi te su pronađena 2 otvorena porta, 80 i 22. Problem kod ovakvog skeniranja je što je dosta nametljivo. Sigurnosna značajka IDS (engl. *Intrusion Detection Systems*) koja uglavnom dolazi u sklopu vatrozida te analizira mrežni promet može otkriti ovakav oblik skeniranja i ako korisnik nije autoriziran od strane čiju mrežu ili stranicu skenira može doći do pravnih posljedica [10]. Iz navedenog razloga skeniranje je moguće obaviti na sigurniji način pomoću TCP SYN skeniranja prikazanog na slici 11.

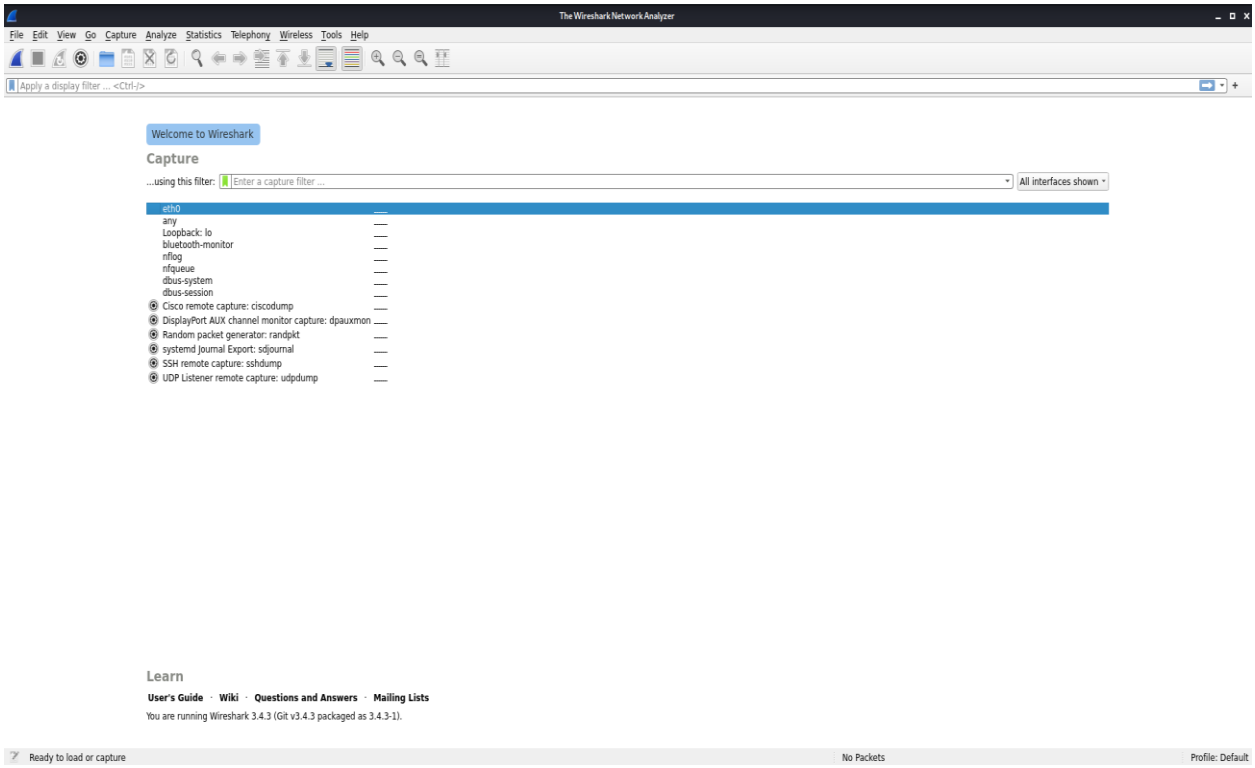


Slika 11. TCP SYN skeniranje

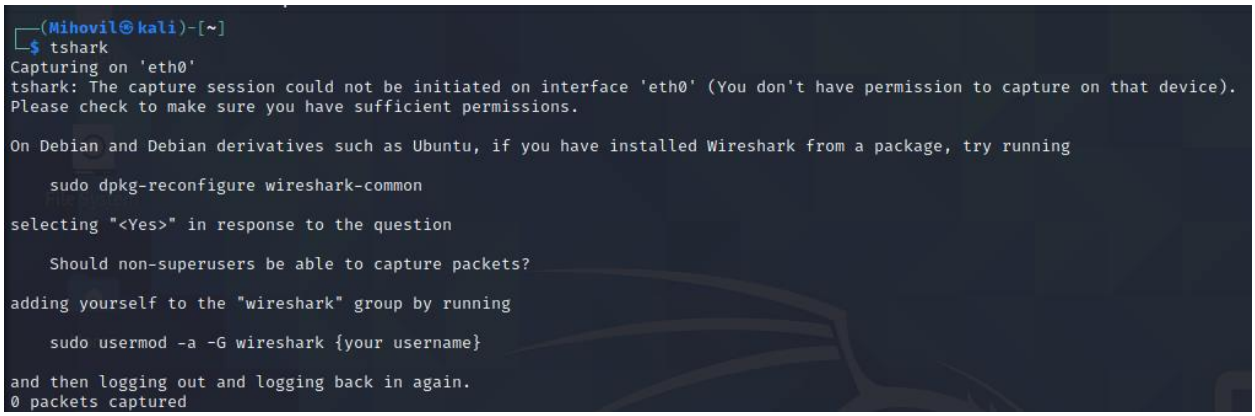
Pomoću zastavice *-sS* pokrenuto je TCP SYN skeniranje. Zastavicom *-p* određeno je koje je portove potrebno skenirati, što su u ovom slučaju portovi 80 i 443. Ako se ne odredi točno koje je portove potrebno skenirati, prema zadanim postavkama prvih 1000 portova ulazi u skeniranje. Prikazani portovi su izabrani iz razloga jer su najčešće otvoreni za promet i povezivanje na mrežu, te iz navedenog razloga predstavljaju najveću prijetnju za mrežu. Iz slike je vidljivo da je port 80 otvoren, a port 443 je filtriran, što znači da vatrozid ili bilo koja druga mrežna zapreka blokiraju port, te Nmap ne može odrediti da li je isti otvoren ili zatvoren. Ovakav oblik skeniranja se često naziva i polu otvoreno skeniranje jer se ne otvora potpuna TCP veza.

3.3.2. Wireshark

Gerald Combs započeo je razvoj alata Wireshark 1998. godine, jednog od najboljih i najkorištenijih alata za analizu mrežnih protokola. Dostupan je i besplatan za mnoge operacijske sustave te podržava analizu za preko 100 mrežnih protokola. Sadrži još mnogo korisnih značajki poput hvatanja komprimiranih datoteka te dekomprimiranja istih, dešifriranje za protokole poput WPA/WPA2, IPsec, Kerberos, hvatanje u stvarnom vremenu i analizu u van mrežnom načinu rada, te izvoz u različite formate kao što su XML, CSV i tekstualni format. U svrhu edukacije mrežnih administratora, mrežnih arhitekata, stručnjaka za mrežnu sigurnost te svih ostalih korisnika koji žele znati više o mrežama, organiziran je SharkFest. SharkFest je niz godišnjih obrazovnih konferencija u raznim dijelovima svijeta usmjerenih na razmjenu znanja, iskustva te trikova pri korištenju Wireshark-a između razvojnog tima i korisnika alata. Polaznici SharkFest-a usavršavaju svoje vještine u području analize paketa pohađanjem predavanja i laboratorijskih vježbi koje održavaju najiskusniji stručnjaci u industriji. Glavni razvojni programeri Wireshark-a okupljaju se tijekom održavanja konferencije kako bi poboljšali i što bolje razvili alat u svrhu održavanja relevantnosti istoga. Pokretanje alata moguće je obaviti upisivanjem naredbe *wireshark* u sučelje naredbenog retka, te je rezultat upisa naredbe prikazan na slici 12. Alat je moguće koristiti i bez pokretanja grafičkog korisničkog sučelja, upisivanjem naredbe *tshark*, što je prikazano na slici 13. Jedina razlika je u obliku korištenja, sve mogućnosti i značajke su iste [12].

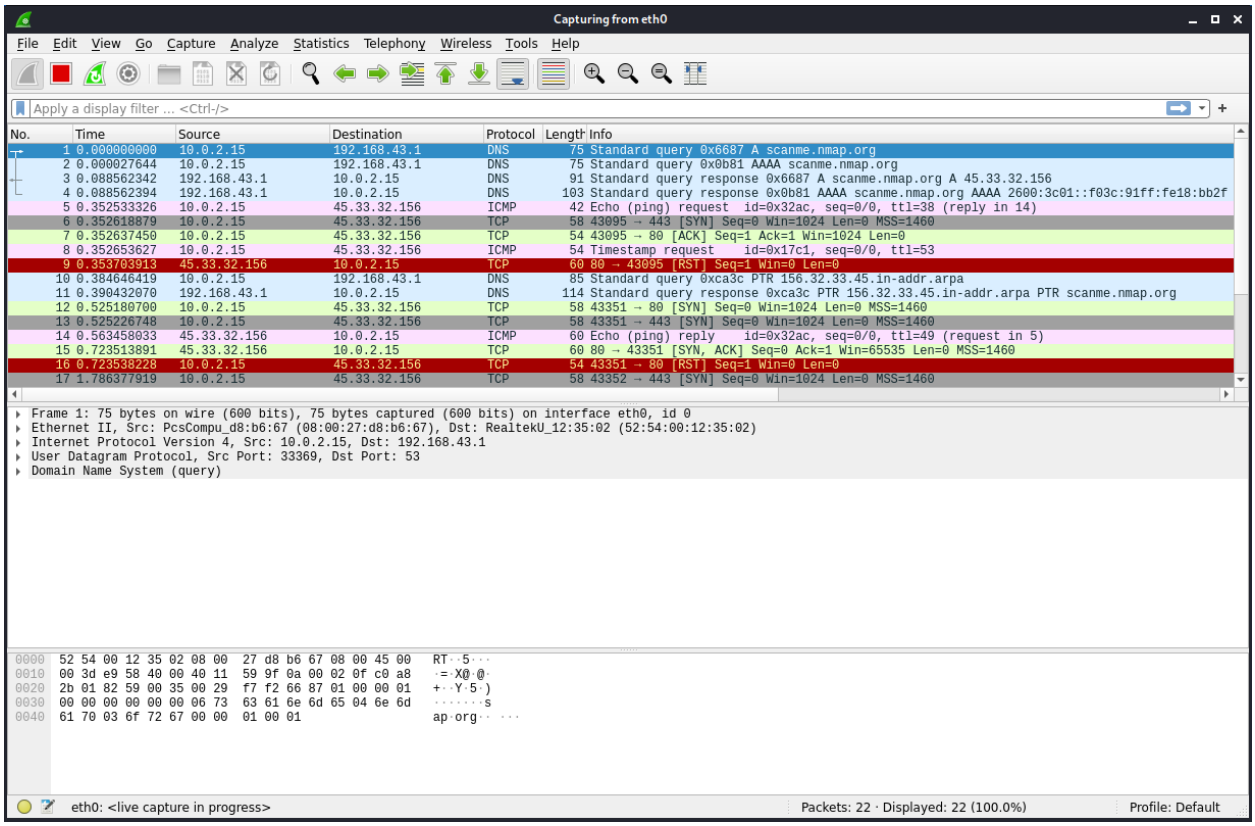


Slika 12. Sučelje Wireshark-a



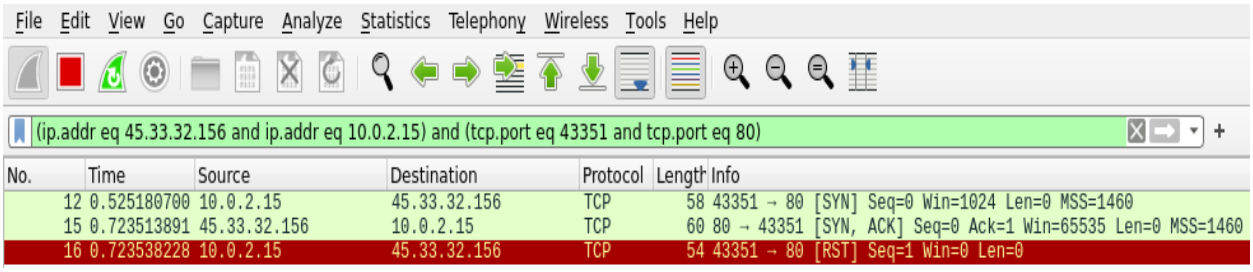
Slika 13. Ispis nakon upisane naredbe *tshark*

Na slici 14. prikazan je uhvaćen sav promet na mreži nakon upisivanja naredbi prikazanih na slici 11. Vidljivo je kako je teško raspoznati pakete s kojima je potrebno nastaviti istraživanje, te iz navedenog razloga filteri uvelike olakšavaju posao. Filtere je moguće dodati u gornju traku iznad liste paketa. Postoje dvije skupine filtera, a to su filteri za prikaz i filteri za hvatanje. Filteri za hvatanje određuju koji paketi će se hvatati, a filteri za prikaz se koriste kako bi se odredilo koji će uhvaćeni paketi biti ispisani u prozoru alata. Filteri za hvatanje se definiraju prije početka hvatanja i ne mogu se mijenjati tijekom, dok se filteri za prikaz mogu mijenjati i prije i tijekom [13].



Slika 14. Ukupni uhvaćeni promet

U sljedećem primjeru, na slici 15., upisan je filter *(ip.addr eq 45.33.32.156 and ip.addr eq 10.0.2.15) and (tcp.port eq 43351 and tcp.port eq 80)* pomoću kojeg je filtrirana izvorna IP adresa koja je u ovom slučaju 45.33.32.156 te odredišna IP adresa koja je 10.0.2.15. Nadalje naredba *tcp.port* označuje prikaz paketa s određenim TCP izvornim ili odredišnim portom, u ovom slučaju to su portovi 43351 te 80.



Slika 15. Uhvaćeni mrežni promet uz korištenje filtera

U prvom redu je vidljivo da je poslana SYN poruka s izvorne IP adrese 10.0.2.15 na odredišnu 45.33.32.156, s porta 43351 na port 80. Nakon toga poslana je SYN, ACK poruka s izvorne IP adrese 45.33.32.156 na odredišnu IP adresu 10.0.2.15, s porta 80 na port 43351. Nakon primljene SYN ACK poruke, poslana je RST poruka s IP adrese 10.0.2.15 na IP adresu 45.33.32.156 kako bi se zatvorila komunikacija. Iz navedenog primjera vidljivo je kako funkcionira hvatanje mrežnog prometa pomoću Wireshark-a, te TCP SYN skeniranje pomoću Nmap alata.

3.4. Alati za testiranje sigurnosti Internet stranica

Internet stranice koje nemaju puno prometa predstavljaju laku metu hakerima jer su ranjive na najčešće sigurnosne propuste, a to su:

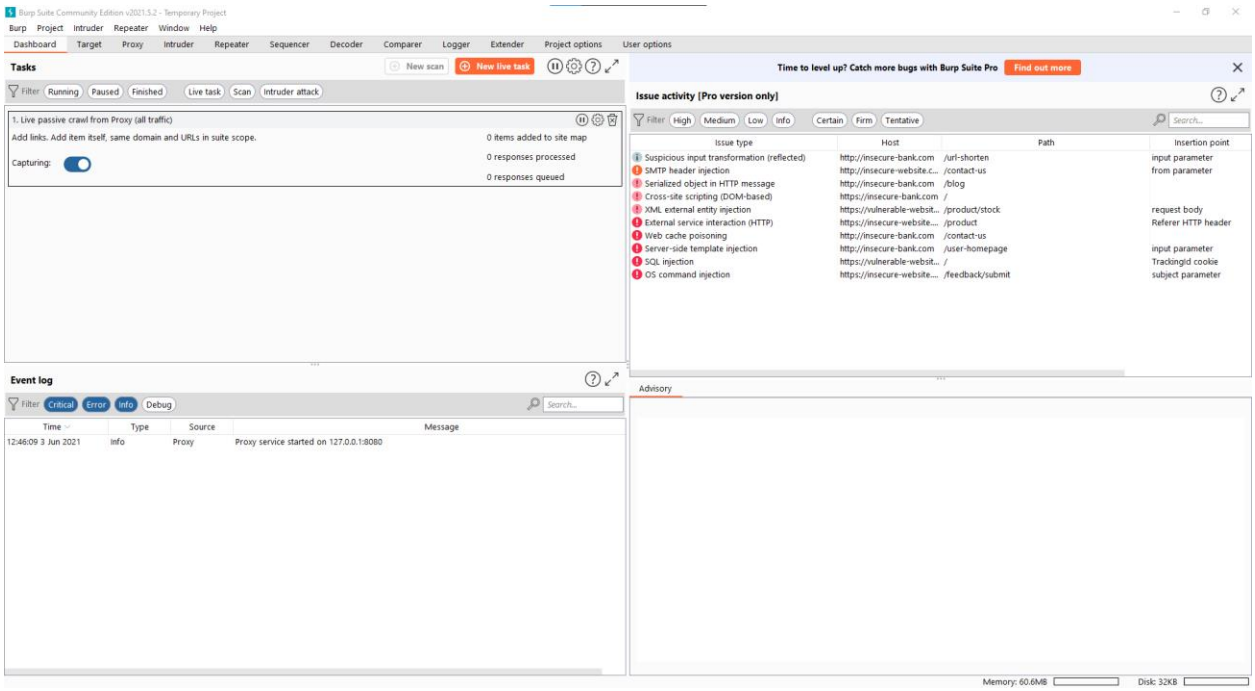
- preko-stranično skriptiranje ili XSS (engl. *cross-site scripting*)
- preko-stranično krivotvorenje zahtjeva ili CSRF (engl. *cross-site request forgery*)
- uskraćivanje usluga ili DOS (engl. *Denial of Service*)
- otmica klikova (engl. *clickjacking*)
- SQL ubrizgavanje (engl. *SQL injection*)

XSS je napad koji omogućuje ubrizgavanje zlonamjernih skripti u inače sigurne i pouzdane stranice. Ovakav tip napada je moguć zbog nedovoljne provjere unosa podataka korisnika stranice, što može rezultirati krađom podataka korisnika kojemu je poslana skripta. CSRF napad omogućuje napadaču izvršavanje radnji koristeći korisničke podatke drugog korisnika bez njegova znanja ili pristanka. Pomoću društvenog inženjeringa napadač manipulira korisnika na odlazak na određeni link u poruci, nakon čega napadač može izvršiti zahtjeve za promjenom stanja na stranici, u kojoj

je korisnik trenutno prijavljen, kao što su zahtjevi za prijenos novčanih sredstava. Uskraćivanje usluga se obično postiže preplavljivanjem ciljane stranice lažnim zahtjevima u svrhu onemogućavanja pristupa ostalim korisnicima. Zahtjevi mogu biti mnogobrojni ili pojedinačno mogu zahtijevati velike količine resursa. Kod otmice klikova, postavljen je nevidljivi element ispod stranice koju korisnik koristi, te prilikom izvršavanja određene radnje na glavnoj stranici, poput klika na gumb, korisnik zapravo izvrši radnju na skrivenoj stranici. Ovim putem napadač, može instalirati zlonamjerni program na korisnikovo računalo, ukrasti podatke i slično [14]. SQL ubrizgavanje objašnjeno je u poglavlju 2.2.1. Kako bi se što efikasnije izbjegli navedeni propusti, potrebno je koristiti alate koji automatiziraju potragu i popravljanje navedenih propusta. Trenutno na tržištu postoji više alata koje rade navedeni zadatak, a među najboljima su Burp Suite, Acunetix te Nessus.

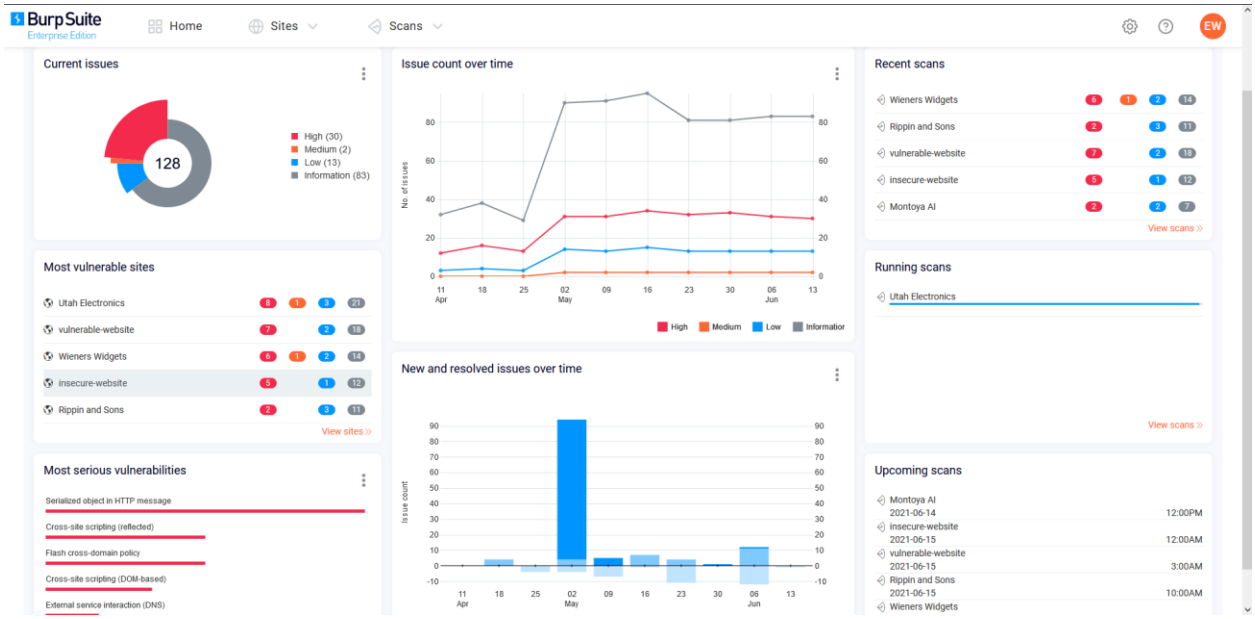
3.4.1. Burp Suite

Burp Suite sadrži mnoge alate koji poboljšavaju testiranje sigurnosti Internet stranica, te se vodi kao jedan od najpotrebnijih alata kada je u pitanju sigurnost istih. Prva verzija alata objavljena je 2003. godine od strane tvrtke PortSwigger. Trenutno postoje 3 izdanja alata, a to su Enterprise, Professional te Community izdanje. Enterprise izdanje je orijentirano za potrebe i korištenje od strane organizacija koje žele povećati sigurnost svojih proizvoda. Neke od značajki navedenog izdanja su konfiguriranje stranica, skeniranje te sustav za praćenje i pronalazak računalnih pogrešaka. Community izdanje je besplatno, te je radi toga jedan od najboljih odabira za početnike u sigurnosnom testiranju Internet stranica. Sučelje istoga prikazano je na slici 16. Professional izdanje košta 349 eura, ali dolazi s puno više značajki i mogućnosti za razliku od Community izdanja. Značajka automatizacije zadataka pri pronalasku sigurnosnih ranjivosti uvelike poboljšava rezultate i smanjuje vrijeme rada. Postoji još mnogo značajki poput mogućnosti izmjene svih HTTP i HTTPS komunikacija koje prolaze kroz preglednik, pronalazak nevidljivog sadržaja te mogućnost korištenja preko 250 proširenja za alat. Neki od najviše korištenih proširenja su Autorize za testiranje sigurnosnih ranjivosti pri ovjeri, Turbo Intruder za automatizirane napade kad je nužna velika brzina ili složenost, Burp Bounty za prilagođavanje mogućnosti skeniranja prema vlastitim potrebama te Param Miner za olakšanu potragu za sigurnosnim ranjivostima u pred memoriji Internet stranice [15].



Slika 16. Sučelje alata Burp Suite Community verzije

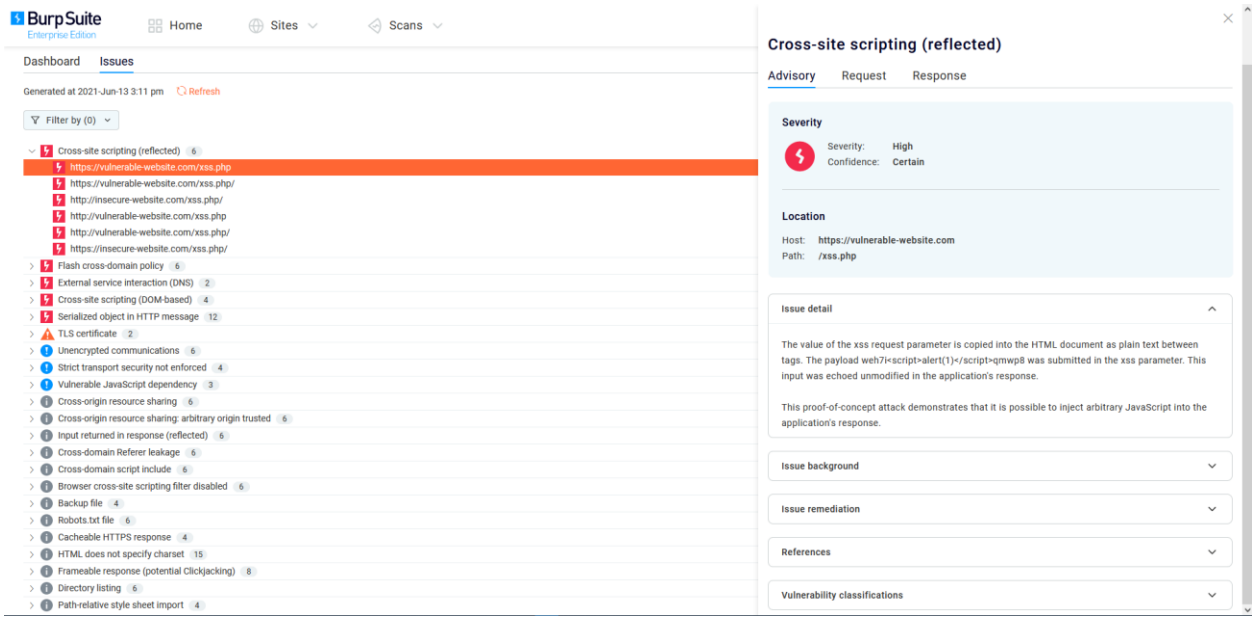
U idućem primjeru prikazano je korištenje Burp Suite Enterprise verzije. Primjer je prikazan pomoću demo verzije alata kojoj je moguće pristupiti na stranici alata [16]. Na slici 17. prikazano je sučelje alata.



Slika 17. Sučelje Burp Suite Enterprise verzije

Na nadzornoj ploči mogu se pronaći odjeljci za trenutne probleme, broj problema kroz vrijeme, nedavna izvješća nakon skeniranja, skeniranja u tijeku, nadolazeća skeniranja, najranjivije stranice, najozbiljniji problemi te novi i riješeni problemi tijekom vremena. Jasno je kako program nudi jednostavno i efikasno sučelje za održavanje sigurnosti stranice. Jedna od mogućnosti

programa je pregled svih pronađenih problema, te mogućnost detaljnog pregleda pojedinog problema.



Slika 18. Prikaz svih pronađenih problema i detalja o XSS problemu

Na slici 18. prikazan je ispis svih pronađenih problema te detaljan prikaz problema koji nastaje u slučaju XSS napada. Iz detalja o sigurnosnom propustu moguće je saznati točno kako je nastao te zašto bi predstavljao problem u budućnosti ako ga netko iskoristi u zlonamjerne svrhe. Također je moguće pročitati o pozadini problema i o mogućim rješenjima. Moguća rješenja su detaljno opisana na koji način se trebaju implementirati i što se dobiva ako se implementira programski kod na navedeni način.

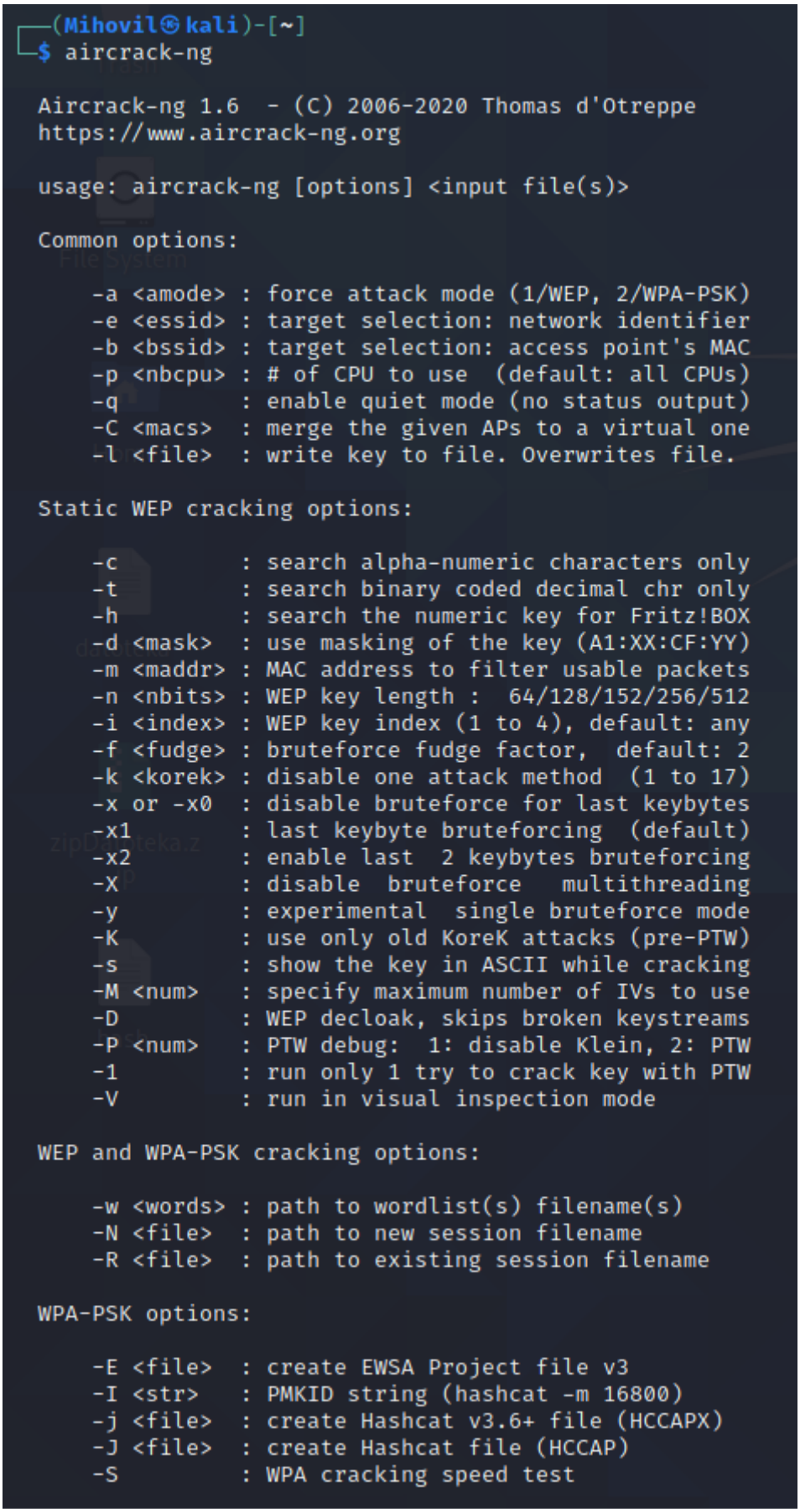
3.5. Alati za testiranje sigurnosti Wi-Fi mreža

Wi-Fi je bežična tehnologija koja omogućuje uređajima poput laptopa, osobnih računala i pametnih telefona korištenje Interneta te međusobnu komunikaciju. IEEE 802.11 standard definira protokole koji omogućuju komunikaciju s bežičnim uređajima koji podržavaju Wi-Fi, kao što su bežični usmjerivači te bežične pristupne točke. Bežična pristupna točka omogućuje bežičnim uređajima spajanje na bežičnu mrežu. Pristupna točka uzima propusnost (engl. *bandwidth*) koja dolazi od usmjerivača i proteže ju tako da se uređaji s većih udaljenosti mogu spojiti na mrežu. Zbog navedenog načina spajanja bežične mreže mogu vrlo lako postati meta hakera koji mogu

iskoristiti nedovoljan stupanj zaštite u svrhu dobivanja neautoriziranog pristupa. Hakeri prilikom prisluškivanja mrežnog prometa mogu presresti pakete u trenutku kada se korisnik prijavljuje u sustav i ako sigurnosne postavke nisu dobro konfigurirane, haker može saznati lozinku i korisničko ime iz navedenih paketa. Putem prisluškivanja hakeri mogu saznati i SSID (engl. *Service Set Identifier*) mreže jer ga nesigurne bežične mreže povremeno emitiraju, koji će im pomoći pri spajanju na mrežu. Za što bolju sigurnost potrebno je koristit WEP (engl. *wired equivalent privacy*), WPA (engl. *Wi-Fi Protected Access*) ili WPA2 algoritme za sigurnu komunikaciju putem bežičnih mreža [10]. Za testiranje sigurnosti Wi-Fi mreže te pravovremeno otkrivanje ranjivosti koriste se alati kao što su Aircrack-ng, Ettercap te inSSIDer.

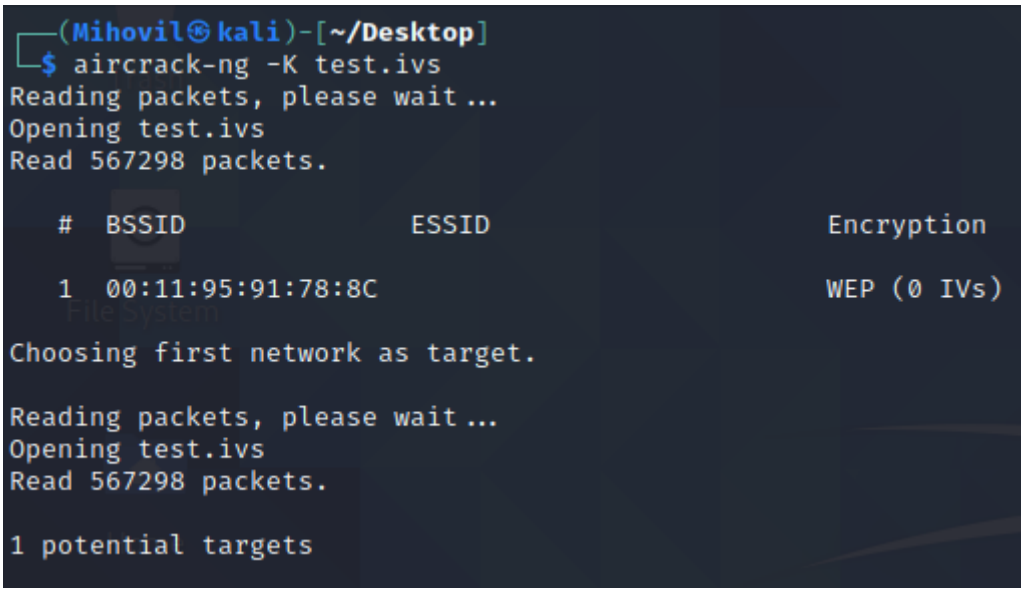
3.5.1. Aircrack-ng

Aircrack-ng je paket alata za testiranje sigurnosti Wi-Fi mreža. Sadržani alati su airbase-ng, aircrack-ng, airdecap-ng, airdecloak-ng, airdrop-ng, aireplay-ng, airgraph-ng, airmon-ng, airodump-ng, airolib-ng, airserv-ng, airtun-ng, besside-ng, dcrack, easside-ng, packetforge-ng, tkiptun-ng te wesside-ng. Alat je dostupan za mnoge operacijske sustave te ima upotrebu u području sigurnosti Wi-Fi mreža, poput nadgledanja, napada, testiranja te probijanja sigurnosti u svrhu otkrivanja lozinke. Pojam nadgledanja podrazumijeva hvatanje paketa i izvoz podataka u tekstualne datoteke za daljnju obradu pomoću alata nezavisnih proizvođača. Pojam napada označuje mogućnost izrade lažne pristupne točke [17]. Upisivanjem naredbe *aircrack-ng* bez dodatnih argumenata, ispisuje se sažetak mogućnosti alata, prikazan na slici 19.



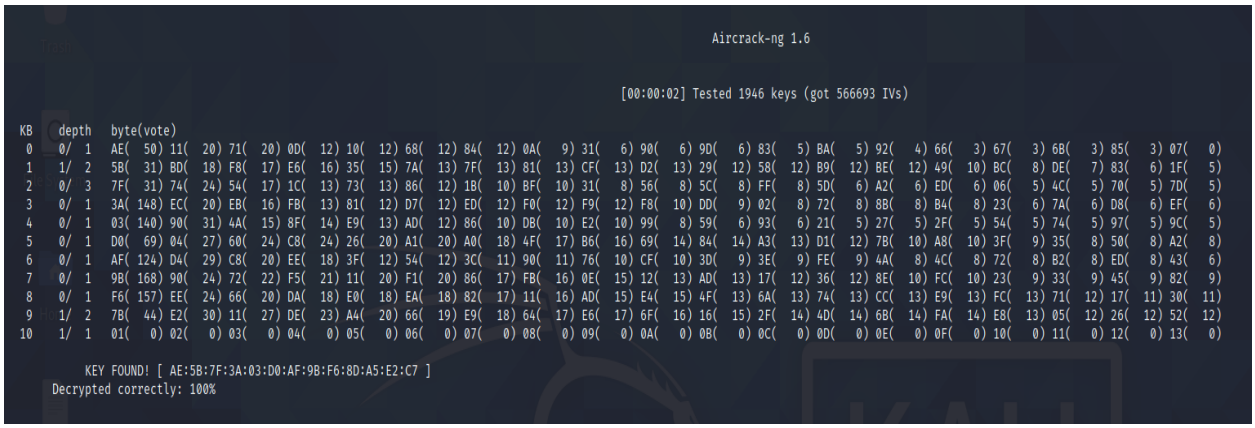
Slika 19. Ispis nakon upisane naredbe *aircrack-ng*

Za primjer korištenja alata preuzete su datoteke sa službene stranice, koje su napravljene u svrhu učenja alata. Za prvi primjer korištena je datoteka sa 128 bit WEP ključem preuzeta s [18]. WEP (engl. *wired equivalent privacy*) je protokol dizajniran kako bi bežični prijenos podatak bio što sigurniji. WEP šifrira podatke koristeći ključ od 40 ili 128 bitova. Za korištenje WEP protokola klijent i poslužitelj moraju znati koji se ključevi za šifriranje koriste. WEP ima nekoliko sigurnosnih propusta koji mogu biti iskorišteni u svrhu dobivanja neautoriziranog pristupa, te iz tog razloga je sigurnije koristiti WPA ili WPA2 protokole.



Slika 20. KoreK napad za pronalazak ključa

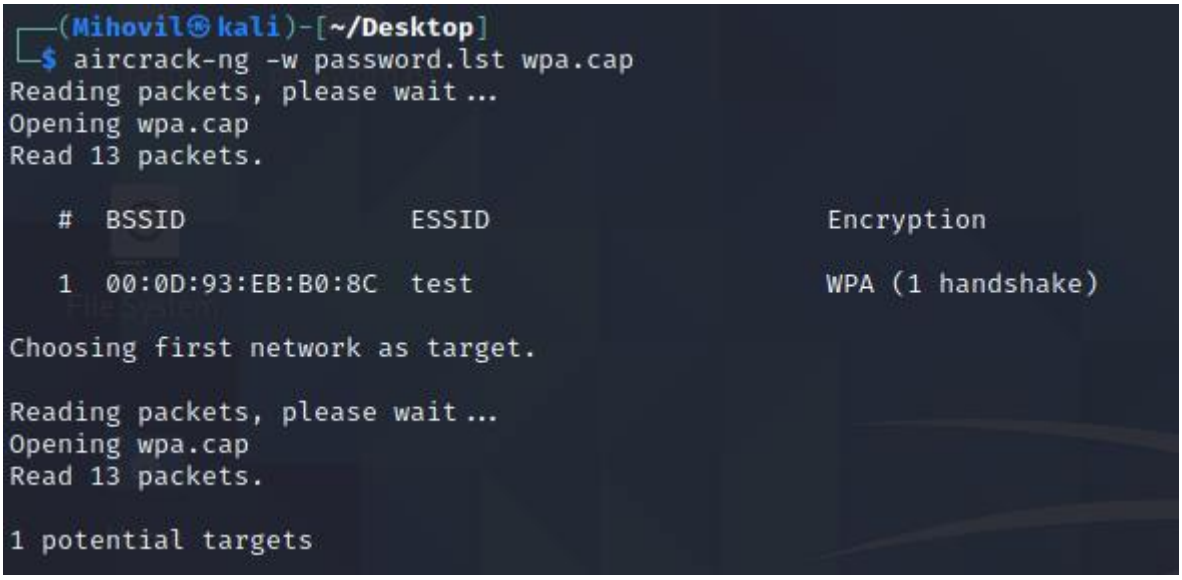
Slika 20. prikazuje ispis nakon KoreK napada. Korištenjem zastavice *-K* pokrenut je KoreK napad na datoteku imena *test.ivs*. IVs datoteka sadrži vektore inicijalizacije (engl. *initialization vectors*), koji služe u svrhu generiranja šifriranih podataka na mreži, pomoću kojih se može otkriti WEP ključ. KoreK je najjednostavniji napad za dobivanje WEP ključa, te je dobio ime po anonimnom programeru koji je objavio kod na forumu *NetStumbler,* koji opisuje 17 napada na WEP protokol [19].



Slika 21. Pronađeni ključ pomoću KoreK napada

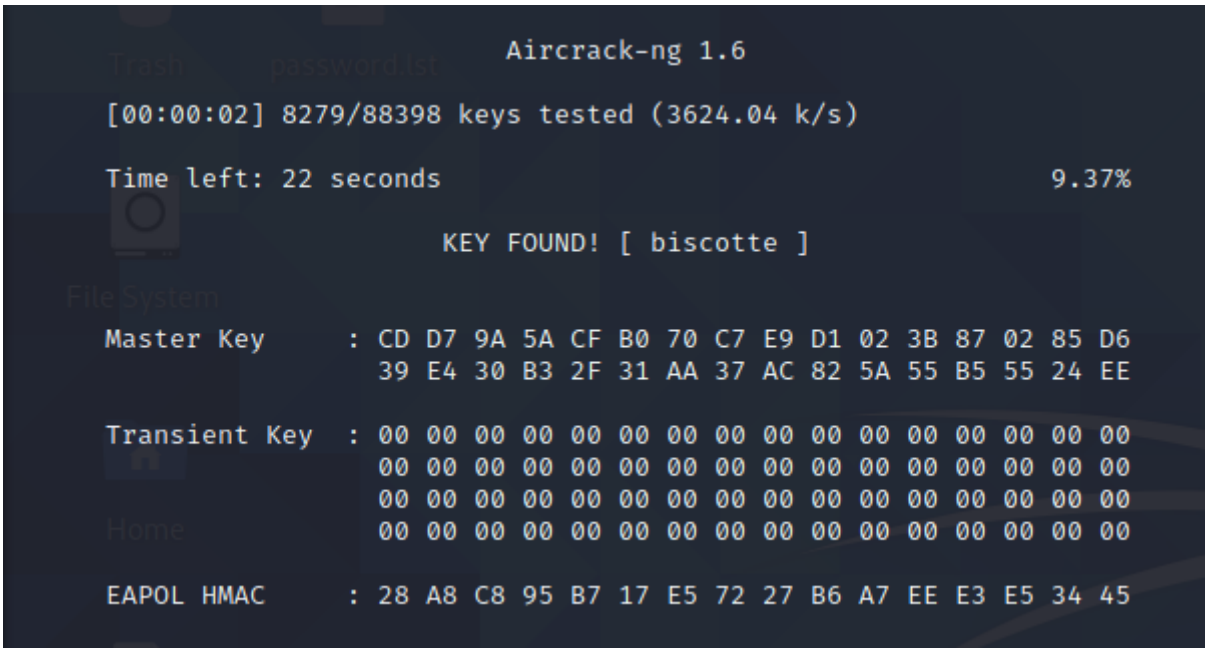
Iz ispisa na slici 21. je vidljivo kako je bilo potrebno 2 sekunde za testiranje 1946 ključeva, te da je traženi ključ AE:5B:7F:3A:03:D0:AF:9B:F6:8D:A5:E2:C7. Pomoću pronađenog ključa moguće je spajanje na mrežu.

U sljedećem primjeru, na slici 22., objašnjeno je otkrivanje WPA lozinke. Za potrebe primjera preuzeta je datoteka sa stranice [20] koja sadrži WPA rukovanje (engl. *WPA handshake*).



Slika 22. Napad za otkrivanje WPA lozinke pomoću liste riječi

Korištenjem zastavice *-w* te naziva datoteke koja sadrži listu riječi, u ovom slučaju *password.lst* pokrenuto je otkrivanje WPA lozinke napadom pomoću liste riječi.



Slika 23. Pronađeni ključ pomoću napada listom riječi

Iz ispisa prikazanog na slici 23. vidljivo je kako je ključ pronađen u 22 sekunde, te isti glasi *biscotte*. Pomoću pronađenog ključa može se spojiti na mrežu.

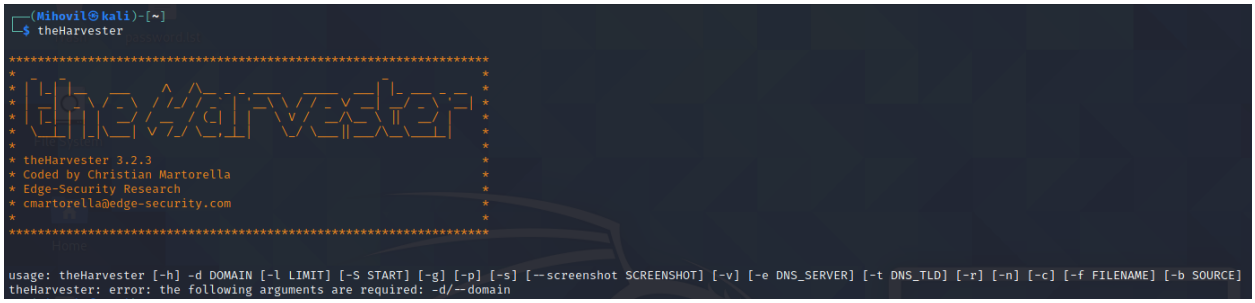
3.6. Alati za prikupljanje informacija o meti

Kako bi se odredila količina javno dostupnih informacija o tvrtki na Internetu koje napadač može saznati u svakom trenutku koriste se alati za prikupljanje javno dostupnih informacija na pretraživačima. Ovakav oblik prikupljanja informacija naziva se inteligencija otvorenog koda (engl. *Open-source intelligence*) ili skraćeno OSINT. Informacije je moguće prikupiti putem Interneta, masovnih medija, istraživanja, fotografija te geoprostornih informacija. Ovakav oblik skupljanja informacija ima nizak rizik, jeftin je i najčešće jako učinkovit. Prikupljene informacije napadači mogu iskoristi u svrhu društvenog inženjeringa. Etički hakeri pomoću alata koji uvelike olakšaju navedeni oblik prikupljana informacija svode javno dostupne informacije o tvrtki i njezinim radnicima na minimum [21].

3.6.1. theHarvester

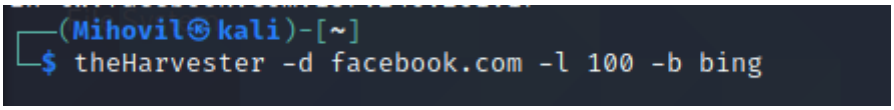
Alat theHarvester se koristi u ranim fazama etičkog hakiranja za prikupljanje javno dostupnih informacija o meti. Moguće je prikupiti informacije o e-mail adresama, imenima, pod domenama,

IP adresama, otvorenim portovima te ostalim javno dostupnim informacijama na mnogim pretraživačima [22].



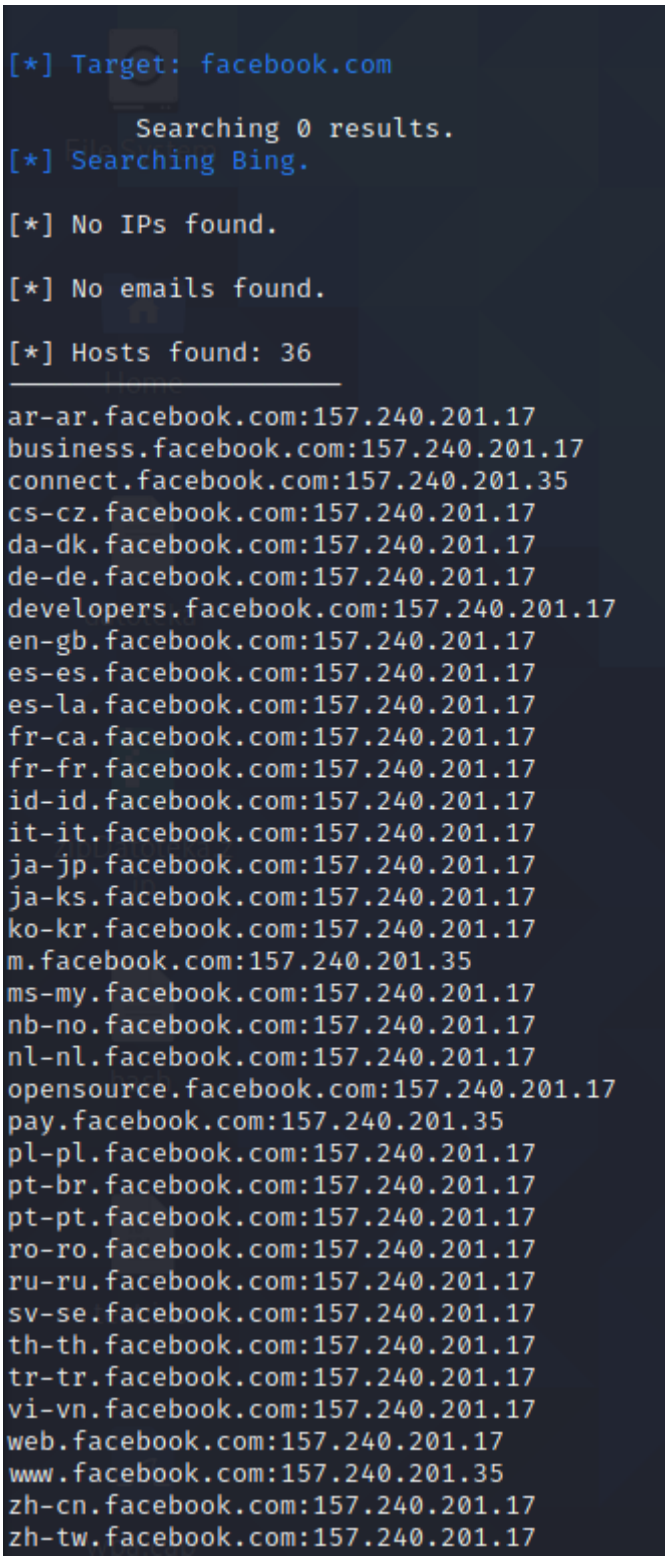
Slika 24. Ispis nakon upisane naredbe *theHarvester*

Pokretanje alata izvršava se upisivanjem naredbe *theHarvester* u sučelje naredbenog retka te se ispisuju informacije o verziji alata, informacije za kontakt autora i osnovne naredbe koje su prikazane na slici 24. U idućem primjeru, na slici 25., prikazano je prikupljanje informacija o domeni facebook.com koristeći Bing tražilicu kao izvor.



Slika 25. Naredbe za pronalazak informacija o facebook.com domeni

Pomoću zastavice *-d* pretraga je usmjerena na domenu *facebook.com*. Zastavicom *-l* ograničen je broj rezultata na 100 koji je po zadanim postavkama postavljen na 500. Kao izvor postavljena je *Bing* tražilica koristeći zastavicu *-b*.



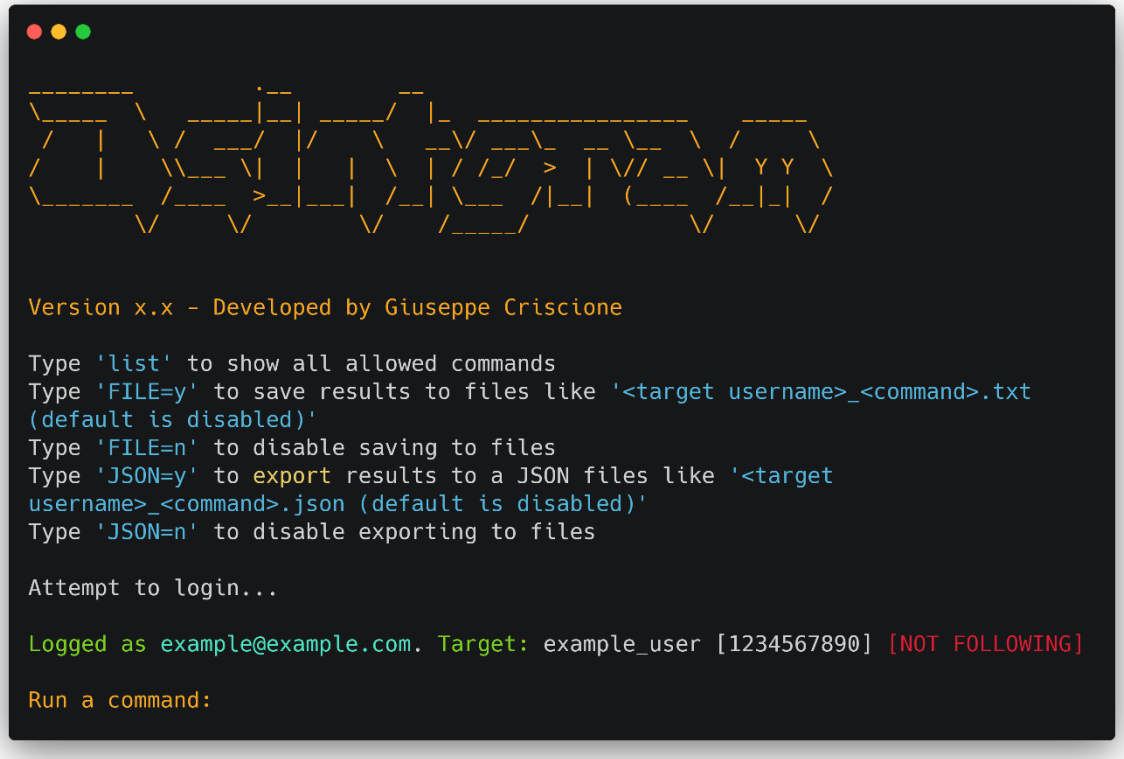
Slika 26. Rezultati pretraživanja facebook.com domene

Iz ispisa prikazanog na slici 26. vidljivo je da je pronađeno 36 domaćina, a niti jedna e-mail i IP adresa.

3.6.2. Osintgram

Osintgram je besplatni alat otvorenog koda za korištenje inteligencije otvorenog koda na aplikaciji Instagram u svrhu prikupljanja i analiziranja podataka mete. Alat je razvio Giuseppe Criscione u Python programskom jeziku. Instagram je društvena mreža pomoću koje korisnici mogu izrađivati i dijeliti fotografije, priče i video zapise za svojim prijateljima i pratiocima. Analiza se provodi na računu bilo kojeg korisnika pomoću njegovog korisničkog imena [23]. Prema [23] alat nudi mogućnosti za prikupljanje podataka na računu kao što su:

- Registrirane adrese fotografija mete
- Opisi slika
- Sveukupan broj komentara i sviđanja na objavama
- Broj korisnika koji prate metu te koje meta prati
- Brojevi mobitela te e-mail adrese korisnika koji prate metu te koje meta prati
- Preuzimanje svih fotografija i priča korisnika



Slika 27. Sučelje alata Osintgram [24]

Ispis na pokretanja alata prikazan je na slici 27. Za korištenje alata potrebno je unijeti korisničko ime i lozinku vlastitog Instagram računa u credentials.ini datoteku, koja se nalazi u config mapi. Sljedeći korak je unos korisničkog imena mete te odabir željene opcije za prikupljanje podataka.

4. UREĐAJI

Uređaji za sigurnosno testiranje mogu poboljšavati ili u potpunosti zamijeniti programske alate. U nekim slučajevima su zbog visokog stupnja sigurnosti tvrtke ili pojedinca neizbježni. Uređaji se koriste za testiranje sigurnosti Wi-Fi mreža, dobivanje udaljenog pristupa sustavu te za napade gdje računalo prepoznaje spojeni uređaj kao tipkovnicu, i time postoji mogućnost brzog i jednostavnog unosa naredbi koje mogu dovesti do neautoriziranog pristupa. Postoji još uređaja koji se koriste u svrhu etičkog hakiranja, ali nisu napravljeni isključivo za navedenu svrhu. Wi-Fi mrežni adapter koji ima mogućnost nadgledanja i ubrizgavanja paketa koristi se kod testiranja sigurnosti bežičnih mreža. Pošto je ponekad potreban rad na terenu, etički hakeri koriste računalo veoma malih dimenzija Raspberry Pi, jer je na navedeno moguća instalacija Kali Linux operacijskog sustava te najčešće korišteni alati ne zahtijevaju veliku računalnu snagu za korištenje. Slika 28. prikazuje veličinu Raspberry Pi računala.

Slika 28. Raspberry Pi [25]

USB Rubber Ducky je uređaj koji omogućuje unošenje slijeda znakova velikom brzinom u računalo, jer ga računalo prepozna kao tipkovnicu. USB Rubber Ducky, prikazan na slici 29., izgleda kao normalan memorijski štapić koji putem društvenog inženjeringa može dospjeti u posjed mete koja će ga spojiti na računalo, a pošto će ga računalo prepoznati kao HID (engl. *Human Interface Device*), otvara se mogućnost za razne napade. Automatizacija napada je moguća putem specijaliziranog programskog jezika Ducky Script. Trenutno postoji dosta gotovih skripti za sve važnije napade koje se mogu besplatno preuzeti sa službene stranice uređaja, ali postoji i mogućnost pisanja vlastitih skripti za razne namjene u navedenom programskom jeziku. Moguće je koristiti napade za krađu Wi-Fi lozinki, krađu podataka i datoteka korisnika, onesposobljavanje Windows sigurnosne zaštite, pokretanje raznih programa, instalaciju programa za udaljeni pristup i slično [26].

Slika 29. USB Rubber Ducky [27]

U idućem primjeru, na slici 30., prikazana je jednostavna skripta napisana u Ducky Script programskom jeziku za otvaranje Internet preglednika Firefox i odlazak na stranicu fakulteta.

```
REM Primjer korištenja Ducky Script jezika
WINDOWS r
DELAY 100
STRING firefox
ENTER
DELAY 200
STRING ferit.unios.hr
ENTER
```

Slika 30. Primjer programa napisanog u Ducky Script programskog jeziku

Naredba *REM* se koristi za pisanje komentara, te kao takva se ne izvršava. Naredba *WINDOWS* imitira pritisak Windows tipke. Korištenjem navedene naredbe popraćene slovom *r*, otvara se skočni prozor koji omogućuje upisivanje naredbi i pokretanje programa. *DELAY* funkcija zaustavlja skriptu za navedeni broj milisekundi. Putem funkcije *STRING* moguće je unijeti niz znakova. Funkcija *ENTER* imitira tipku Enter.

Uređaj Packet Squirrel, prikazan na slici 31., koristi se za napade čovjekom u sredini, hvatanje paketa, ostvarivanje udaljenog pristupa i kloniranje MAC adrese. Moguće je jednostavno hvatanje paketa koji putuju mrežom, te kasnija analiza uhvaćenih paketa u programu Wireshark, obrađenog u poglavlju 3.3.2. Kod napada čovjekom u sredini napadač može presresti promet između dvije strane. Postoji mogućnost pisanja vlastitih skripti u prethodno objašnjenom programskom jeziku Ducky Script ili preuzimanja gotovih skripti sa službene stranice uređaja. Uređaj se može koristi i u svrhu zaštite privatnosti u obliku VPN (engl. *Virtual Private Network*) usmjerivača. VPN radi tako da skriva pravu IP adresu uređaja i šifrira sav promet tako da nitko ne može vidjeti sadržaj koji se pretražuje na internetu. Zbog male veličine i težine te mogućnosti brzog mijenjanja funkcije za koju se koristi uređaj je pogodan i prikladan za rad na terenu [28].

Slika 31. Packet Squirrel [29]

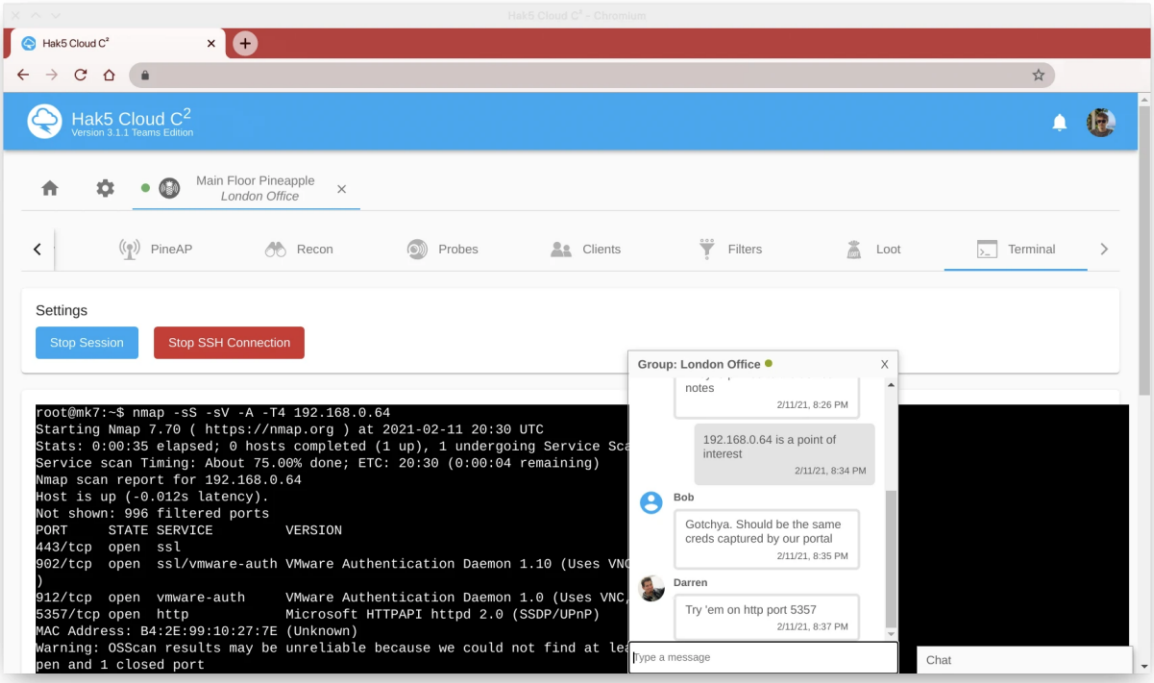
Wi-Fi Pineapple, prikazan na slici 32., koristi se u svrhu sigurnosnog testiranja Wi-Fi mreža, nadziranja Wi-Fi mreža i analizu rezultata, prikupljanje informacija te napada čovjekom u sredini. Uređaj se može konfigurirati tako da izgleda kao normalna Wi-Fi mreža koju mete misle da koriste. Nakon što se meta spoji na lažnu Wi-Fi mrežu napadači mogu tehnikama objašnjenim u poglavlju 3.5 ukrasti podatke mete, kao što su bankovni podatci. Iz navedenog razloga javno dostupne Wi-Fi mreže predstavljaju veliki sigurnosni rizik. Sučelje uređaja se koristi putem Internet preglednika i ne zahtijeva nikakvu instalaciju programa na računalo što ga čini jako prikladnim za korištenje na terenu ili u manjku vremena. Postoji mogućnost vizualizacije Wi-Fi mreže i povezanosti između pristupnih točaka i uređaja na njoj. Filterima za MAC i SSID vrijednosti je moguće smanjiti kolateralnu štetu i ograničiti opseg uređaja koje se može napasti [30]. MAC (engl. *Media Access Control*) adresa ili fizička adresa je jedinstvena mrežna adresa svakog uređaja na svijetu. SSID (engl. *Service Set Identifier*) je ime koje definira bežičnu mrežu, te ga je potrebno znati pri spajanju na mrežu [10].

Slika 32. Wi-Fi Pineapple [31]

Key Croc, prikazan na slici 33., je zapisnik pritisnutih tipki (engl. *keylogger*) koji može pokrenuti napade kada je određeni niz znakova unesen u računalo. Podražava mnoge napade kao što je napad unosa slijeda tipki te krađa mrežnih podataka. Postoji mogućnost preuzimanja gotovih skripti ili pisanja vlastitih u Ducky Script programskog jeziku. Uređaj ima mogućnost spajanja na Internet i time omogućuje jednostavan udaljeni pristup sustavu na koji je spojen. Van navedenih mogućnosti, svojstvo uređaja se pritiskom na tipku može prebaciti na memorijski štapić što omogućava jednostavnu promjenu postavki te pristup alatima za sigurnosno testiranje kao što su Nmap, Responder, Impacket i Metasploit [32].

Slika 33. Key Croc [33]

Sve navedeno uređaje proizvodi tvrtka Hak5, koja nudi i jednostavnu platformu za kontrolu svih uređaja koje korisnik posjeduje Cloud C^2, prikazanu na slici 34. Platforma ima podršku za sve opisane uređaje te nudi poboljšanje trenutnih mogućnosti i korištenje novih. Trenutno postoje tri verzije, a to su Community, Professional te Teams verzija. Community verzija je besplatna i podržava do 10 uređaja, Professional košta 500 dolara i podržava 50 uređaja, a Teams verzija se plaća ovisno o broju članova, od 1000 do 2000 dolara s podrškom od 50 do 200 uređaja [34].



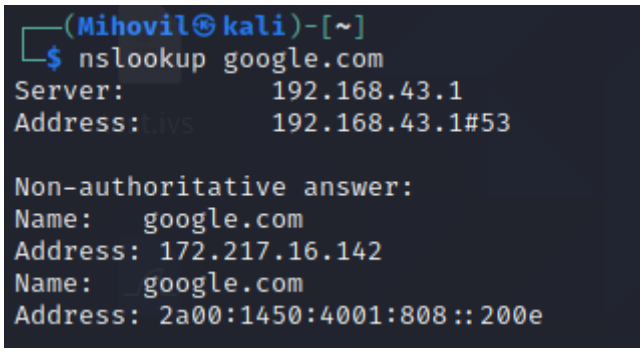
Slika 34. Sučelje Cloud C² Teams izdanja alata [35]

5. FAZE ETIČKOG HAKIRANJA

Kako bi se etičko hakiranje izvelo što efikasnije potrebna je dobra organizacija vremena i rada. Iz navedenog razloga proces etičkog hakiranja dijeli se u 5 faza, a to su izviđanje, skeniranje, stjecanje pristupa, održavanje pristupa te pokrivanje tragova [36]. Svaka faza igra bitnu ulogu u procesu sigurnosnog testiranja, sve faze su međusobno povezane te bi loša izvedba jedne mogla dovesti do neuspjeha u traženju sigurnosnih popusta sustava. Potrebno je naglasiti kako sve faze ne zahtijevaju jednaku vremensku posvećenost, jer faza izviđanja može potrajati i po nekoliko mjeseci, ako su zaposlenici tvrtke svjesni o mogućim pokušajima dobivanja neautoriziranog pristupa sustavu od strane hakera pomoću tehnika društvenog inženjeringa.

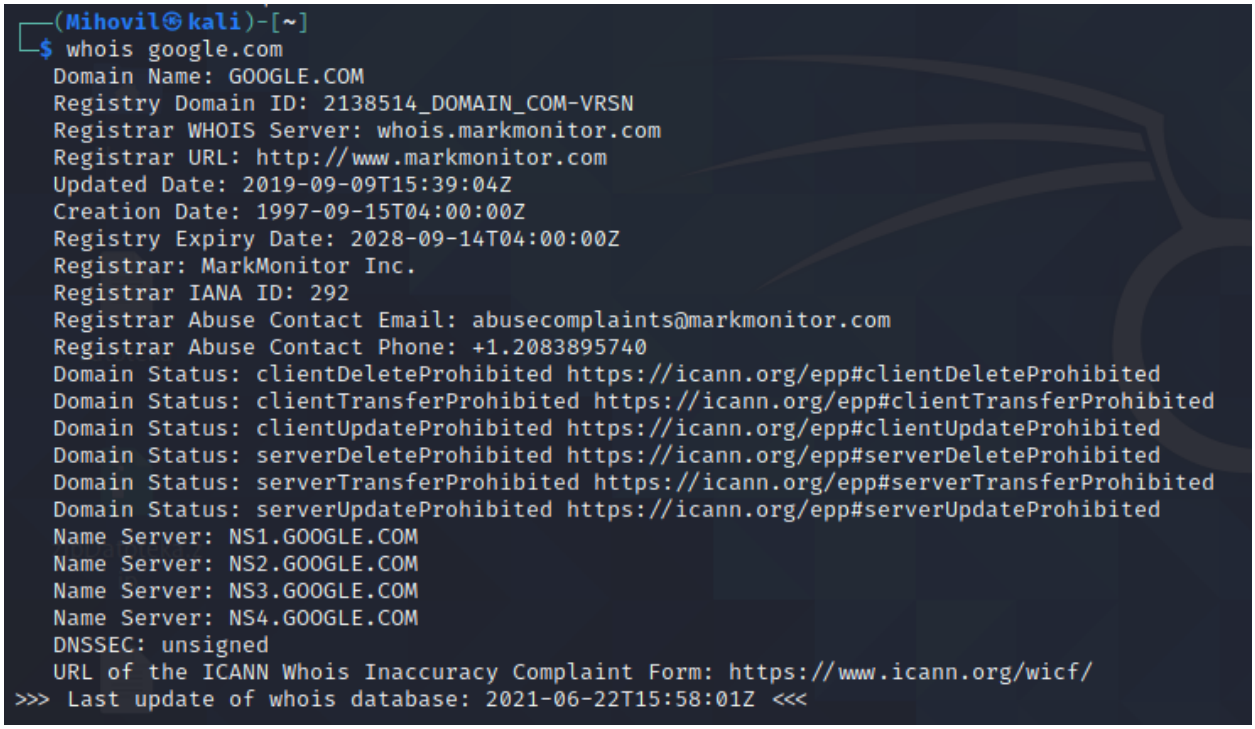
5.1. Izviđanje

Faza izviđanja je faza pripreme, što znači da je potrebno skupiti što je više moguće informacija o meti prije provođenja sigurnosnog testiranja. Postoje dva pristupa fazi izviđanja, a to su aktivno i pasivno. Aktivno izviđanje uključuje traženje otvorenih portova, lokacija usmjerivača te detalja o operacijskim sustavima. Aktivno izviđanje je slično fazi skeniranja, razlika je u tome što je skeniranje puno opširnije i detaljnije. Kod pasivnog izviđanja etički haker ne komunicira direktno sa sustavom nego koristi javno dostupne informacije za prikupljane podataka [37]. Za prikupljane navedenih informacija koristi se inteligencija otvorenog koda te alati objašnjeni u poglavlju 3.6. Pomoću navedenih alata moguće je prikupiti informacije koje mogu dovesti do potencijalnih sigurnosnih propusta. Prikupljanje informacija je moguće i putem društvenog inženjeringa. Informacije koje mogu pomoći su brojevi mobitela, e-mail adrese te informacije o sustavu. Alati i naredbe koje mogu dodatno olakšati fazu izviđanja osim prethodno navedenih su *nslookup*, *traceroute* te *whois*. *Nslookup* naredba se koristi za pronalazak informacija koje se mogu koristiti za dijagnozu DNS (engl. *Domain Name System*) infrastrukture [38] te je korištenje iste prikazano na slici 35. za google.com domenu. DNS je naziv servisa za internetske adrese koji prevodi nazive domena neslužbeni numeričke adrese internetskog protokola (IP) [39].



Slika 35. Ispis *nslookup* naredbe

Traceroute se koristi za pregled puta kojim paketi idu od izvorne do odredišne IP adrese. *Whois* baza podataka sadrži informacije o domeni kao što su status domene, datumi stvaranja, nadograđivanja i isticanja domene, imena servera te razne kontaktne informacije [40]. Primjer korištenja navedene je prikazan na slici 36. za google.com domenu.



Slika 36. Ispis *whois* naredbe

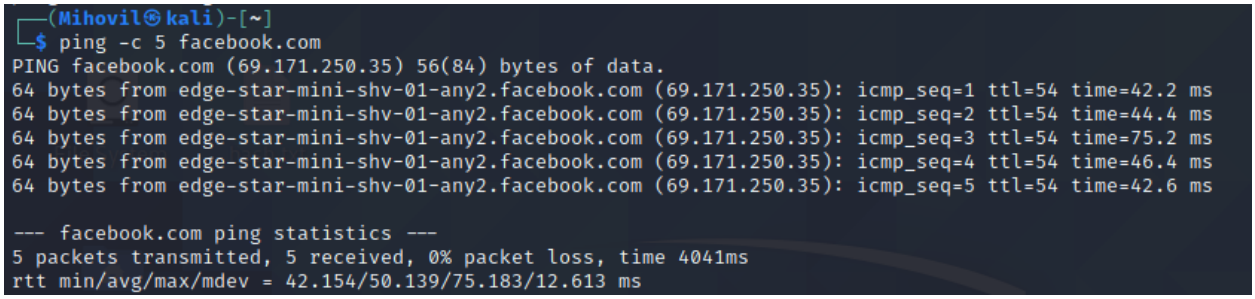
5.2. Skeniranje

Nakon uspješno odrađene faze izviđanja, potrebno je započeti fazu skeniranja mete. U fazi skeniranja koriste se detalji i informacije prikupljeni u fazi izviđanja za pronalazak mogućih sigurnosnih propusta. Prema [41] skeniranje se dijeli u 7 koraka:

- Provjera postoje li trenutno aktivni sustavi
- Traženje otvorenih portova

- Skeniranje izvan dosega IDS-a
- Prikupljanje informacija o operacijskim sustavima i servisima
- Skeniranje u svrhu pronalaska sigurnosnih ranjivosti
- Izrada mrežnih dijagrama
- Priprema posredničkog poslužitelja (engl. *proxy*)

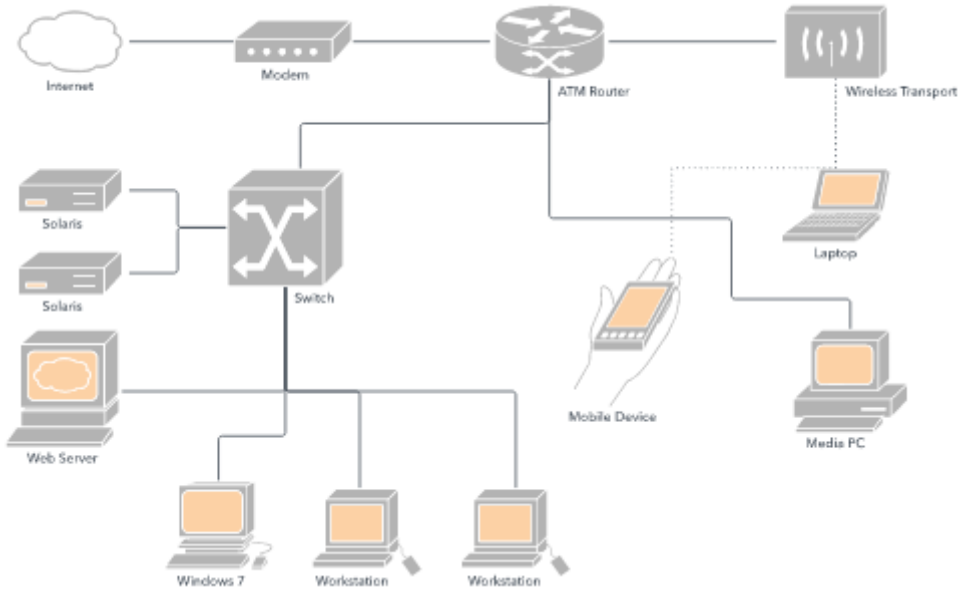
Korake nije potrebno raditi navedenim redoslijedom, ali ih je bitno odraditi sve. Prvi korak je moguće odraditi pomoću naredbe *ping*, naredbe koja služi za provjeru povezanosti s odredišnim računalom, a korištenje iste je prikazano na slici 37.



Slika 37. Ispis *ping* naredbe

Traženje otvorenih portova, prikupljanje informacija o operacijskim sustavima i servisima te skeniranje izvan dosega IDS-a može se obaviti putem alata Nmap, čije je korištenje objašnjeno u 3.3.1. poglavlju. Za skeniranje u svrhu pronalaska sigurnosnih ranjivosti koriste se tehnike i alati objašnjeni u poglavlju 3.4. Mrežni dijagram je vizualni prikaz računale ili telekomunikacijske mreže. Pokazuje elemente koji čine mrežu i kako međusobno komuniciraju, uključujući usmjerivače, uređaje, vatrozid i slično. Može biti logički i fizički. Logičkim dijagramom se prikazuje kako informacije putuju mrežom, a fizičkim kako su uređaji koji čine tu mrežu

postavljeni [42]. Korištenjem mrežnom dijagrama može se dobiti prikladniji prikaz mreže za koju se treba naći ranjivosti. Slika 38. prikazuje jednostavni mrežni dijagram.



Slika 38. Mrežni dijagram [43]

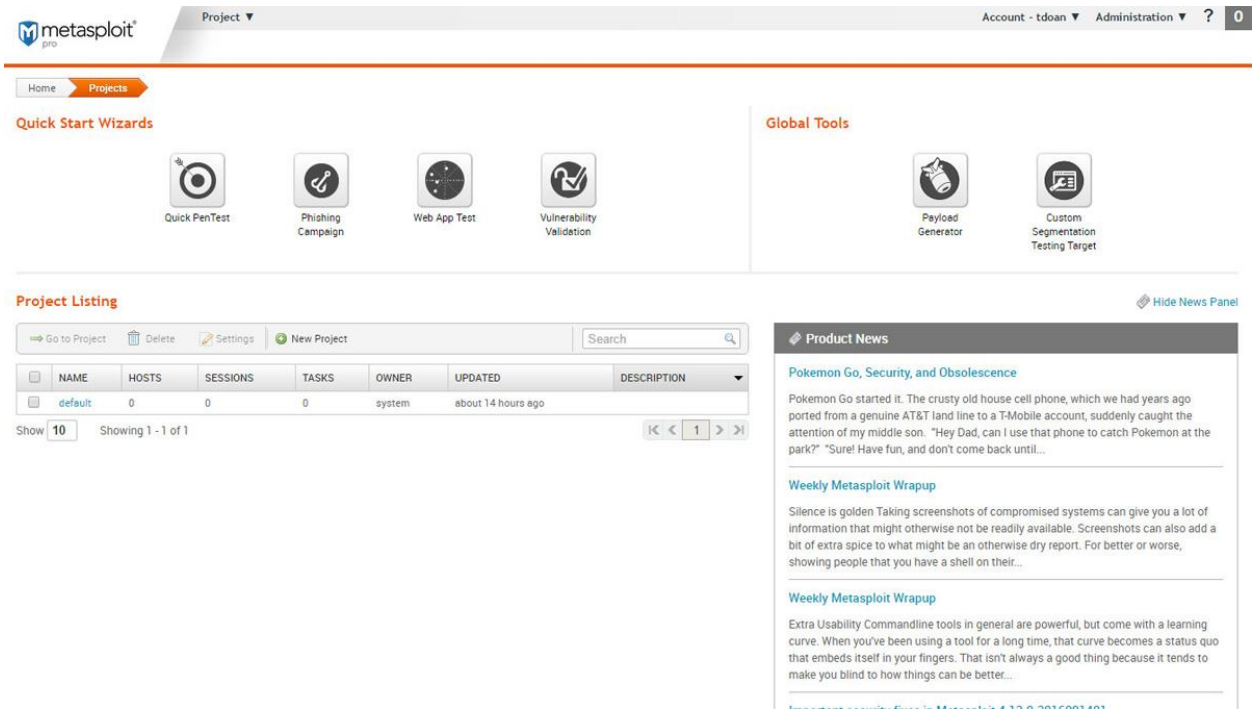
Posrednički poslužitelj se koristi u svrhu skrivanja tako da etički haker šalje naredbe i zahtjeve posredničkom poslužitelju, a on ih prenosi odredišnom poslužitelju. Skrivanje na ovakav način se može obaviti s jednim posredničkim poslužiteljem ili više njih zbog veće sigurnosti [40].

5.3. Stjecanje pristupa

Na temelju svih informacija o sustavu i radnicima prikupljenih u prethodnim fazama potrebno je ostvariti pristup korištenjem određenih napada. Prije faze stjecanja pristupa potrebno je imati kvalitetne informacije o sigurnosti sustava, radnicima, operacijskim sustavima te mogućim sigurnosnim propustima. Probijanje sigurnosti lozinke predstavlja najprikladniji način stjecanja pristupa. Problem kod probijanja sigurnosne zaštite lozinke je stupanj povlastica korisnika, to jest ako korisnik nema administratorski pristup sustavu, mogućnost upravljanja njegovim računom ima minimalne koristi. Prema [40] postoje četiri načina za ostvarenje administratorskih povlastica u sustavu, a to su:

- Probijanje sigurnosne zaštite lozinke administratorskog računa
- Iskorištavanje sigurnosnih propusta u svrhu uporabe administratorskih prava
- Korištenje alata za pronalazak sigurnosnih ranjivosti
- Društveni inženjering

Načini su poredani po važnosti, stoga je probijanje sigurnosne zaštite lozinke administratorskog računa primarni cilj. Navedeno se postiže alatima i tehnikama opisanim u poglavlju 3.2. Kod drugog načina je upitno postoje li uopće sigurnosni propusti koji će omogućiti navedeno, pošto većina sustava često dobiva sigurnosne zakrpe koje onemogućavaju iste. Kod navedenog načina napadač također treba imati i veliko znanje o prethodno pronađenim propustima kako bi mogao provjeriti postoje li isti kod mete. Treći način je korištenje alata koji će omogućiti željeni pristup. Jedan od najkorištenijih alata za navedenu svrhu je Metasploit, a sučelje navedenog je prikazano na slici 39.

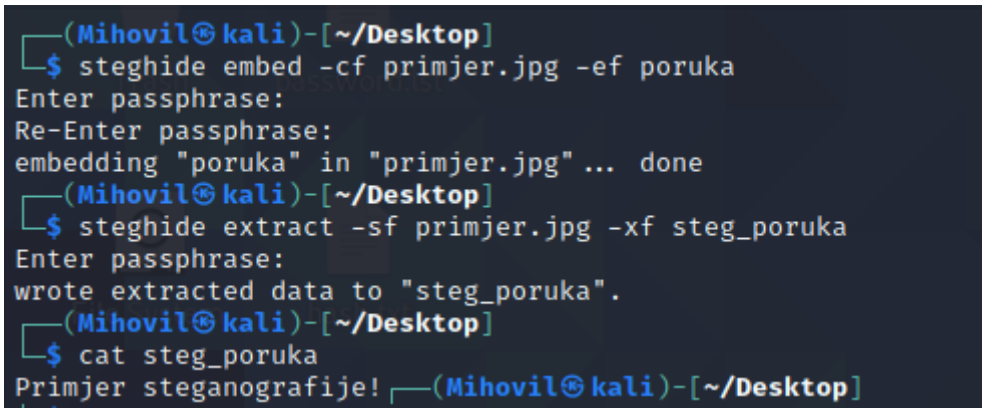


Slika 39. Sučelje alata Metasploit [41]

Prema [42] koraci kod rada s alatom su sljedeći: stvaranje projekta, sastavljanje podataka od meti, proučavanje i upravljanje podatcima domaćina, pokretanje skeniranja za potragu sigurnosnih propusta, postavljanje alata za slušanje, iskorištavanje pronađenih sigurnosnih propusta, prikupljanje dokaza, čišćenje tragova te pisanje izvještaja. Ovisno o cilju korištenja, redoslijed koraka se može promijeniti, neki koraci se mogu izostaviti te se mogu dodati novi. Alat nudi automatsko iskorištavanje sigurnosnih propusta za prethodno definiranu metu te kasnije istraživanje propusta. Četvrti način uključuje manipuliranje korisnika na preuzimanje datoteke ili odlazak na link koji će omogućiti izvršavanje skripte za dobivanje administratorskog pristupa. Navedeni način je s razlogom na zadnjem mjestu, iako je daleko najlakši i najbrži od spomenutih, kod tvrtki s viskom stupnjem sigurnosti i znanja radnika teško će uspjeti.

5.4. Održavanje pristupa

Pošto stjecanje pristupa sustavu može biti vremenski i financijski zahtjevno, potrebno je održavati pristup sustavu. Ovisno o cilju, nekada je potrebno više puta pristupiti sustavu te je iz tog razloga bitno imati jednostavan pristup istome. Ponovni pristup sustavu može se osigurati putem tehnika trojanskog konja ili stražnjih vrata. Trojanski konj je program koji se lažno predstavlja da izvodi funkciju koju korisnik očekuje prije instalacije ili pokretanja, ali zapravo izvodi funkciju, uglavnom bez korisnikova znanja, koja krade informacije, omogućava udaljeni pristup sustavu ili na bilo koji drugi način ugrožava sigurnost sustava. Putem trojanskog konja moguće je stvoriti mogućnost korištenja stražnjih vrata. Stražnja vrata (engl. *backdoor*) su skrivena mogućnost u sustavu ili programu u svrhu zaobilaska računalnog sustava ovjere. Stražnja vrata često ne mogu opstati nakon nadogradnje sustava, zbog novih sigurnosnih zakrpa. Ako je napadač ostvario administratorski pristup u prethodnom koraku, otvara se mogućnost dodavanja novog korisnika s administratorskim pravima što u budućnosti znači vrlo lak i skriven pristup sustavu. Održavanje pristupa se može osigurati korištenjem uređaja ili programa koji zapisuju pritisnute tipke, jer ako administrator odluči promijeniti lozinke, napadač će ih s lakoćom saznati. Pristup se može održati i putem sakrivanja određenih skripti ili programa u medijima poput slika ili videa, postupkom koji se naziva steganografija. Steganografija je tehnika skrivanja podataka unutar drugog medija, poput slike, video zapisa ili datoteke. Jedan od primjera je steganografija najmanje značajnog bita ili LSB (engl. *least significat bit*) steganografija. U navedenoj vrsti steganografije informacija se skriva u najmanje značajan bit medija, na primjer slike. Kod slike svaki se piksel sastoji od 3 bajta podataka koji odgovaraju crvenoj, zelenoj i plavoj boji. Kod LSB steganografije zamjenjuje se zadnji bit svakog bajta kako bi se sakrio 1 bit podatka, što znači kako bi se sakrio 1 megabajt podataka potrebno je imati sliku veliku 8 megabajta. Pošto promjena zadnjeg bita vrijednosti piksela ne rezultira vidljivom promjenom izgleda slike, osoba koja uspoređuje originalnu sliku i sliku na koju je primijenjena tehnika steganografije neće moći pronaći razlike [45]. Tehnika otkrivanja primjene steganografije u datotekama se naziva stegoanaliza. U idućem primjeru, na slici 40., prikazano je skrivanje poruke u sliku korištenjem alata *steghide*.

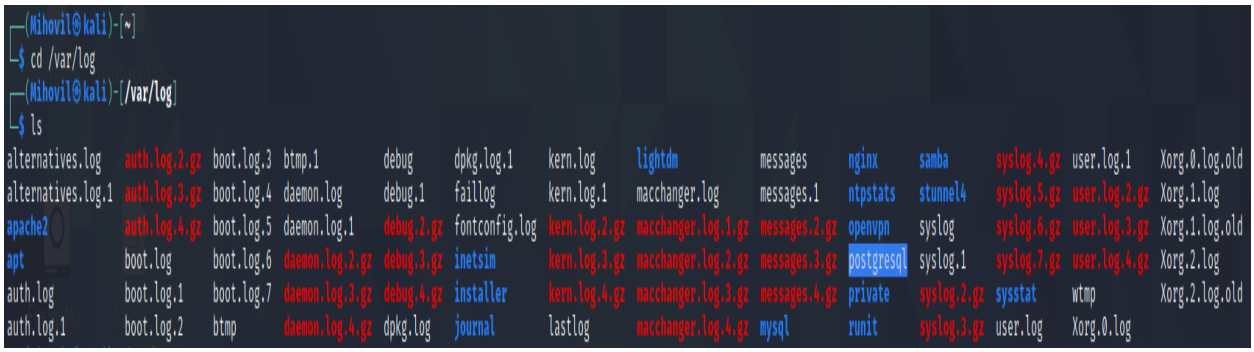


Slika 40. Korištenje *steghide* funkcije

Naredbom *embed* naznačeno je kako da će se koristiti funkcija ugrađivanja poruke u datoteku, zastavicom *-cf* definirana je datoteka koja se koristi, što je u ovom slučaju slika naziva *primjer.jpg*, te je zastavicom *-ef* definirana datoteka u kojoj se nalazi poruka, datoteka je naziva *poruka*, a njezin sadržaj je rečenica *Primjer steganografije!*. Nakon unošenja navedenih naredbi potrebno je definirati lozinku koju je potrebno znati pri izvlačenju poruke iz slike, radi jednostavnosti primjera upisana lozinka je 1234. Kako bi se pokrenulo izvlačenje podataka iz datoteke potrebno je upisati naredbu *extract*, zatim zastavicom *-sf* definirati datoteku iz koje se izvlači poruka, te zastavicom *-xf* odrediti datoteku u koju će se upisati pronađena poruka, što je u ovom slučaju *steg_poruka*. Nakon upisivanja naredbi i lozinke, postupak je gotov. Naredba *cat* služi za ispis sadržaja datoteke, te je kao takva korištena za ispis sadržaja *steg_poruka* datoteke. Iz ispisa je vidljivo kako je poruka uspješno izvađena iz slike.

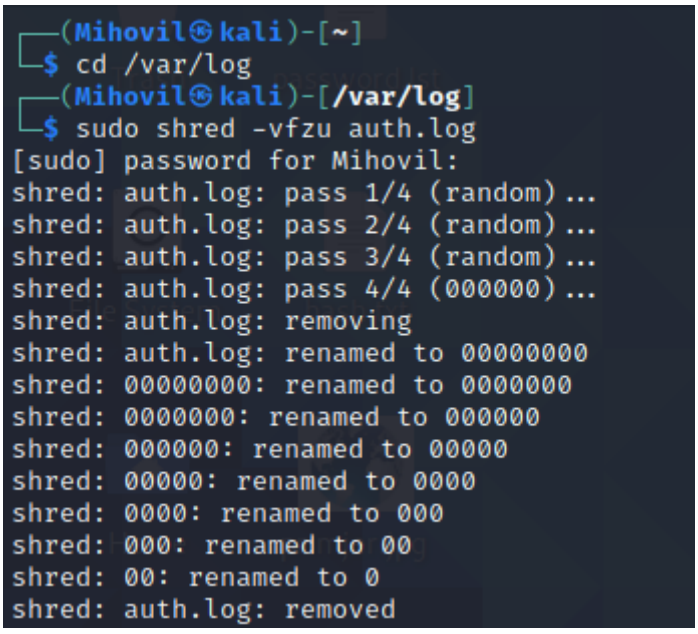
5.5. Pokrivanje tragova

Pokrivanje svih tragova koji su nastali u prethodnim fazama je neophodno kako bi se smanjilo potrebno vrijeme za otkrivanje proboja od strane sustavnih administratora ili potpuno onemogućilo otkrivanje istoga. Da bi se postiglo navedeno potrebno je pronaći i obrisati sve datoteke zapisnika koje služe u svrhu rješavanja problema i kasnije forenzike. U Kali Linux operacijskom sustavu datoteke zapisnika nalaze se u log direktoriju, a put do navedenog je */var/log*, te je sadržaj istoga prikazan na slici 41.



Slika 41. Sadržaj log direktorija

Kod navedene radnje potrebno je obratiti pozornost koji točno sadržaj se treba obrisati, pošto administratori sustava često pregledavaju datoteke zapisnika, te bi veliki nedostatak podataka podigao veliku sumnju. Još jedan od načina je isključivanje zapisivanja događaja u datoteku zapisnika, ali kod navedenog načina je potrebno ograničiti događaje na poruke o pogreškama, poruke o nedostatku resursa i slično jer bi velika vremenska rupa u zapisniku dovela do sumnji. Također je potrebno obrisati i sve poruke o neuspjelim prijavama i poruke o pogreškama koje su nastale tijekom prethodnih faza zbog korištenja određenih napada. U nekim slučajevima je potrebno obrisati cijeli sadržaj datoteke zapisnika, na primjer kada se zbog prirode napada očekuje brzo otkrivanje i odgovor na napad. Tada potpuno brisanje datoteka zapisnika predstavlja siguran način pokrivanja tragova. Navedeno se može postići korištenjem alata *shred*, što je i prikazano na slici 42. Alat je izričito uspješan jer više puta piše preko sadržaja datoteke, što uvelike otežava oporavak sadržaja.



Slika 42. Korištenje *shred* alata

Izabrana datoteka za brisanje je *auth.log*, u koju se spremaju zapisi o ovjeri, te kao takva predstavlja bitan trag kod korištenja sustava. Korištenjem naredbe *shred* pokrenut je alat, te je zastavicom *-v* omogućeno prikazivanje napretka brisanja, zastavicom *-f* se mijenjaju dozvole kako bi se po potrebi omogućilo pisanje u datoteku, zastavica *-z* definira završno pisanje preko sadržaja s nulama kako bi se sakrio postupak brisanja, i na kraju zastavica *-u* vraća zauzetu memoriju i briše datoteku nakon pisanja preko sadržaja. Još jedna od često korištenih tehnika za izbjegavanje otkrivanja tragova je steganografija, objašnjena u prethodnom poglavlju 5.4.

6. ZAKLJUČAK

Ovim završim radom obrađena je tema etičkog hakiranja i alata koji se koriste tijekom istoga. U radu je detaljno opisan pojam etičkog hakiranja, te se istaknute i objašnjene razlike u odnosu na neetičko hakiranje. Za svaki opisani alat priložen je primjer korištenja u okruženju operacijskog sustava Kali Linux. Opisani su i fizički uređaji koji se koriste pri etičkom hakiranju, te je putem opisa faza etičkog hakiranja objašnjen postupak sigurnosnog testiranja sustava.

Na početku rada nakon detaljnog teorijskog objašnjena dva tipa hakiranja, opisani su događaji koji su realistični primjer svakog tipa. Pomoću teorijske podloge i primjera istaknuta je važnost etičkog hakiranja u današnjem digitaliziranom svijetu. Objašnjene su koristi koje mogu imati tvrtke i pojedinci od usluga koje nude etički hakeri. Nakon objašnjena operacijskog sustava Kali Linux, opisani su i ilustrirani primjerima praktične primjene najčešće korišteni alati u okruženju navedenog operacijskog sustava. Alati djeluju na području testiranja snage i oporavka lozinke, testiranja sigurnosti mrežnog prometa, Internet stanica, Wi-Fi mreža te prikupljanja javno dostupnih informacija o meti. Svaki primjer alata se temeljio na iskorištavanju čestog sigurnosnog propusta. Prikazani primjeri su dokaz razine nesigurnosti stvari koje na dnevnoj bazi koriste tvrtke i pojedinci. Opisom i objašnjenjem uređaja, prikazano je kako naizgled normalni i bezopasni uređaji mogu u vrlo kratkom roku učiniti veliku štetu i uvelike ugroziti sigurnost tvrtke ili pojedinca. Proces etičkog hakiranja je radi bolje učinkovitosti podijeljen u faze. Svaka faza je detaljno opisana i objašnjena je njena važnost, te su navedeni alati, uređaji i tehnike koje pomažu pri uspješnom izvršavanju iste.

Na temelju primjera odrađenih alata, svaki pojedinac može naučiti kako promijeniti svoje svakodnevne nesigurne radnje na Internetu, poput korištenja nesigurnih lozinki, postavljanja previše osobnih informacija na društvene mreže te nedovoljne provjere kredibilnosti pošiljatelja primljenih poruka. Svakim danom raste broj korisnika Internet usluga, što otvara mogućnosti hakerima za ispunjavanje zlonamjernih radnji. Etički hakeri svakog dana sudjeluju u podizanju stupnja sigurnosti Internet usluga te omogućavaju prosječnim korisnicima sigurno korištenje društvenih mreža. Ako se korisnike poduči o mogućim prijetnjama digitalnog svijeta, proboji u sustave i krađe osobnih informacija se mogu svesti na minimum.

LITERATURA

[1] Microsoft Hyper-V Bounty Program, URL: microsoft.com/en-us/msrc/bounty-hyper-v?SilentAuth=1&rtc=1 [15.04.2021]

[2] The 'S' in Zoom, Stands for Security, uncovering (local) security flaws in Zoom's latest macOS client, URL: objective-see.com/blog/blog_0x56.html [15.04.2021]

[3] What is WannaCry ransomware?, URL: kaspersky.com/resource-center/threats/ransomware-wannacry [15.04.2021]

[4] Alleged International Hacker Indicted for Massive Attack on U.S. Retail and Banking Networks, URL: justice.gov/opa/pr/alleged-international-hacker-indicted-massive-attack-us-retail-and-banking-networks [15.04.2021]

[5] R. Hertzog, J. O'Gorman, M. Aharoni, J. Long, Kali Linux Revealed: Mastering the Penetration Testing Distribution, Offsec Press, New York, 2021.

[6] Kali Linux 2020.4 Release (ZSH, Bash, CME, MOTD, AWS, Docs, Win-KeX & Vagrant), URL: kali.org/blog/kali-linux-2020-4-release [17.04.2021]

[7] E. Bauman, Y. Lu, Z. Lin, Half a Century of Practice: Who Is Still Storing Plaintext Passwords?. Information Security Practice and Experience. ISPEC 2015.

[8] John the Ripper password cracker, URL: openwall.com/john [18.04.2021]

[9] Hashcat advanced password recovery, URL: hashcat.net/hashcat [20.04.2021]

[10] D. Lowe, Networking All-in-One For Dummies®, 7th Edition, John Wiley & Sons, Inc., Hoboken, New Jersey, 2018.

[11] The Official Nmap Project Guide to Network Discovery and Security Scanning, URL: nmap.org/book/toc.html [24.04.2021]

[12] Wireshark User's Guide, URL: wireshark.org/docs/wsug_html_chunked [25.04.2021]

[13] Defining And Saving Filters, URL: wireshark.org/docs/wsug_html_chunked /ChWorkDefineFilterSection.html [25.04.2021]

[14] Web Security Academy, URL: portswigger.net/web-security/learning-path [1.05.2021]

[15] Burp Suite is the choice of security professionals worldwide, URL: portswigger.net/burp [1.05.2021]

[16] Burp Suite Enterprise Edition , URL: enterprise-demo.portswigger.net/ [2.05.2021]

[17] Aircrack-ng, URL: aircrack-ng.org/doku.php [4.05.2021]

[18] Aircrack-ng, URL: download.aircrack-ng.org/wiki-files/other/test.ivs [4.05.2021]

[19] R. Chaabouni, Break WEP Faster with StatisticalAnalysis, School of Computer and Communication SciencesSemester Project, 2006.

[20] Aircrack-ng, URL: github.com/aircrack-ng/aircrack-ng/raw/master/test/wpa.cap [4.05.2021]

[21] OSINT: What is open source intelligence and how is it used?, URL: portswigger.net/daily-swig/osint-what-is-open-source-intelligence-and-how-is-it-used [10.05.2021]

[22] theHarvester, URL: github.com/laramies/theHarvester [10.05.2021]

[23] Osintgram, URL: github.com/Datalux/Osintgram [10.05.2021]

[24] Osintgram, URL: raw.githubusercontent.com/Datalux/_Osintgram/master/.img/_carbon.png [10.05.2021]

[25] Raspberry Pi, URL: raspberrypi.org/homepage-9df4b/static/wide-hero-shot-fd4b988f932ff88a1a5dfd6375b28d22.png [17.05.2021]

[26] USB Rubber Ducky, URL: hak5.org/products/usb-rubber-ducky-deluxe [17.05.2021]

[27] USB Rubber Ducky ,URL: cdn.shopify.com/s/files/1/0068/2142/products/rubber_ducky_800x.jpg?v=1590788897 [17.05.2021]

[28] Packet Squirrel, URL: hak5.org/products/packet-squirrel [20.05.2021]

[29] Packet Squirrel, URL: cdn.shopify.com/s/files/1/0068/2142/products/_Packet_Squirrel_800x.jpg?v=1508535547 [20.05.2021]

[30] WiFi Pineapple, URL: hak5.org/products/wifi-pineapple [21.05.2021]

[31] WiFi Pineapple, URL: cdn.shopify.com/s/files/1/0068/2142/products/wp-mk7_81d03a53-bf1a-426f-9425-a34c8b3d9c85_800x.jpg?v=1599680489 [21.05.2021]

[32] Key Croc, URL: hak5.org/products/key-croc [22.05.2021]

[33] Key Croc, URL: cdn.shopify.com/s/files/1/0068/2142/products/keycroc1b_800x .png?v=1589166058 [22.05.2021]

[34] Cloud C2 – Remote pentesting made easy, URL: hak5.org/products/c2 [27.05.2021]

[35] Cloud C2 – Remote pentesting made easy, URL: cdn.shopify.com/s/files/1/0068 /21 42/files/teams1.png?v=1614035533 [27.05.2021]

[36] Certified Ethical Hacking, The 5 Phases Every Hacker Must Follow, EC-Council

[37] nslookup, URL: docs.microsoft.com/en-us/windows-server/administration/windows-commands/nslookup [3.06.2021]

[38] Opis Domain Name System (DNS), URL: support.microsoft.com/hr-hr/topic/opis-domain-name-system-dns-d7476f12-818e-1db7-aa7b-7066fb5e382a [3.06.2021]

[39] About WHOIS, URL: support.google.com/domains/answer/3288171?hl=en [3.06.2021]

[40] M. Walker, CEH™ Certified Ethical Hacker All-in-One Exam Guide, Fourth Edition, McGraw-Hill Education, New York, 2019.

[41] What is a Network Diagram, URL: lucidchart.com/pages/network-diagram [15.06.2021]

[42] What is a Network Diagram, URL: d2slcw3kip6qmk.cloudfront.ne t/marketing/pages/chart/seo/network/discovery/cisco-network-diagram.svg [15.06.2021]

[43] Metasploit, URL: rapid7.com/globalassets/_images/product/metasploit/metasploit-product-hero-image-alt.jpg [15.06.2021]

[44] Metasploit, URL: rapid7.com/products/metasploit/ [15.06.2021]

[45] What is steganography? A complete guide to the ancient art of concealing messages, URL: portswigger.net/daily-swig/what-is-steganography-a-complete-guide-to-the-ancient-art-of-concealing-messages [22.06.2021]

POPIS SLIKA

Slika 1. Radna površina i glavni izbornik Kali Linux-a 6
Slika 2. Sučelje naredbenog retka 7
Slika 3. Ispis nakon upisane naredbe *john* 9
Slika 4. Redoslijed naredbi pri napadu listom riječi 10
Slika 5. Kreirana *hash* vrijednosti pomoću naredbe *zip2john* 10
Slika 6. Dodavanje novog korisnika u sustav 11
Slika 7. Stvaranje datoteke s *hash* vrijednosti lozinke 11
Slika 8. Naredbe i ispis nakon napada rječnikom 12
Slika 9. Ispis nakon upisane naredbe *nmap* 14
Slika 10. TCP skeniranje 15
Slika 11. TCP SYN skeniranje 15
Slika 12. Sučelje Wireshark-a 16
Slika 13. Ispis nakon upisane naredbe *tshark* 17
Slika 14. Ukupni uhvaćeni promet 17
Slika 15. Uhvaćeni mrežni promet uz korištenje filtera 18
Slika 16. Sučelje alata Burp Suite Community verzije 20
Slika 17. Sučelje Burp Suite Enterprise verzije 20
Slika 18. Prikaz svih pronađenih problema i detalja o XSS problemu 21
Slika 19. Ispis nakon upisane naredbe *aircrack-ng* 23
Slika 20. KoreK napad za pronalazak ključa 24
Slika 21. Pronađeni ključ pomoću KoreK napada 25
Slika 22. Napad za otkrivanje WPA lozinke pomoću liste riječi 25
Slika 23. Pronađeni ključ pomoću napada listom riječi 26
Slika 24. Ispis nakon upisane naredbe *theHarvester* 27
Slika 25. Naredbe za pronalazak informacija o facebook.com domeni 27
Slika 26. Rezultati pretraživanja facebook.com domene 28
Slika 27. Sučelje alata Osintgram [24] 29
Slika 28. Raspberry Pi [25] 30
Slika 29. USB Rubber Ducky [27] 31
Slika 30. Primjer programa napisanog u Ducky Script programskog jeziku 32
Slika 31. Packet Squirrel [29] 33
Slika 32. Wi-Fi Pineapple [31] 34

Slika 33. Key Croc [33]..... 35
Slika 34. Sučelje Cloud C^2 Teams izdanja alata [35] 36
Slika 35. Ispis *nslookup* naredbe..... 38
Slika 36. Ispis *whois* naredbe 38
Slika 37. Ispis *ping* naredbe..... 39
Slika 38. Mrežni dijagram [43]..... 40
Slika 39. Sučelje alata Metasploit [41] 41
Slika 40. Korištenje *steghide* funkcije 43
Slika 41. Sadržaj log direktorija 44
Slika 42. Korištenje *shred* alata 44

SAŽETAK

Cilj ovog rada bio je definirati i objasniti pojam etičkog hakiranja, definirati razlike u odnosu na neetičko te istaknuti relevantnost istoga u današnjem društvu. Objašnjeni su najčešći sigurnosni propusti i ishodi ako se isti ne otkriju na vrijeme. Kroz demonstraciju alata prikazan je postupak sigurnosnog testiranja u svrhu pronalaska određenih propusta te su predloženi postupci za popravljanje i buduće izbjegavanje istih. Za svaki je alat detaljno objašnjena teorija područja na kojemu se alat koristi, opisane su značajke te su praktičnim primjerom prikazane mogućnosti i važnost alata. Zatim su objašnjeni uređaji koji poboljšavaju rad alata i olakšavaju sigurnosno testiranje. Za svaki uređaj objašnjeno je područje primjene i opisane su najvažnije značajke. Na kraju su objašnjene i opisane faze etičkog hakiranja. Za svaku je fazu istaknuto koji se prethodno objašnjeni alati i uređaji koriste, što je cilj provođenja faze te važnost pravilnog izvođenja zbog uspješnog provođenja postupka sigurnosnog testiranja.

Ključne riječi: alati, etičko hakiranje, faze etičkog hakiranja, uređaji

ABSTRACT

The aim of this paper was to define and elaborate on the concept of ethical hacking, differentiate it from the unethical hacking and emphasize its relevance in today's society. The most common security vulnerabilities along with consequences of their late detection are listed. Demonstrating the tools, the security testing procedure for finding certain vulnerabilities accompanied by procedures for fixing and avoiding them in the future is explained. The field of use and tool features are elaborated on and exemplified. Furthermore, devices which enhance or replace the tools are explained. Each device’s main features and usages are analyzed. Finally, the phases of ethical hacking are described relating each phase with previously described tools and devices. Each phase aim and the importance of proper execution for the purpose of successful security testing procedure implementation is clarified.

Keywords: tools, ethical hacking, stages of ethical hacking, devices