Metode penetracijskog testiranja korištenjem Metasploit platforme

Maravić, Marin

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

Metode penetracijskog testiranja korištenjem Metasploit platforme

Marin Maravić

Split, rujan 2024.

Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu

Prirodoslovno-matematički fakultet

Odjel za informatiku

Ruđera Boškovića 33, 21000 Split, Hrvatska

METODE PENETRACIJSKOG TESTIRANJA KORIŠTENJEM METASPLOIT PLATFORME

Marin Maravić

SAŽETAK

Cilj ovog rada je pružiti sveobuhvatan pregled metoda penetracijskog testiranja koristeći Metasploit, te na taj način obogatiti znanje i praksu budućih sigurnosnih stručnjaka i istraživača u ovom važnom području. Rad započinje osnovnim teorijskim definicijama kibernetičke sigurnosti i penetracijskog testiranja, te objašnjenjem rada same Metasploit platforme, a zatim se detaljnije analiziraju faze penetracijskog testiranja, način rada modula i primjena modula unutar pojedinih faza.

Ključne riječi: Penetracijsko testiranje, Kali, Metasploit, kibernetička sigurnost

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad sadrži:	54 stranicu, 45 grafičkih prikaza i 18 literaturnih navoda. Izvornik je na hrvatskom jeziku.
Mentor:	**prof. dr. sc. Saša Mladenović,** *redoviti profesor Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*
Neposredni voditelj:	**Dino Nejašmić, mag. educ. math. et inf.,** *predavač Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*
Ocjenjivači:	**prof. dr. sc. Saša Mladenović,** *redoviti profesor Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*
	Antonela Prnjak, mag. educ. inf., *asistent Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*
	Dino Nejašmić, mag. educ. math. et inf., *predavač Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*

Rad prihvaćen: **rujan, 2024**

Basic documentation card

Master Thesis

University of Split

Faculty of Science

Department of Informatics

Ruđera Boškovića 33, 21000 Split, Croatia

METHODS OF PENETRATION TESTING USING THE METASPLOIT FRAMEWORK

Marin Maravić

ABSTRACT

The goal of this thesis is to provide a comprehensive overview of penetration testing methods using Metasploit, thereby enriching the knowledge and practice of future security professionals and researchers in this important field. The thesis begins with the basic theoretical definitions of cybersecurity and penetration testing, and an explanation of the work of the Metasploit platform itself, and then the phases of penetration testing, the mode of operation of modules and the application of modules within individual phases are analyzed in more detail.

Key words: penetration testing, Kali Linux, Metasploit, cybersecurity

Thesis deposited in library of Faculty of science, University of Split

Thesis consits of: 54 pages, 45 figures and 18 references.
Original language: Croatian.

Mentor: **Saša Mladenović, Ph.D.**, *Professor at the Faculty of Science and Mathematics, University of Split*

Supervisor: **Dino Nejašmić, mag. educ. math. et inf.,** *Lecturer at the Faculty of Science and Mathematics, University of Split*

Reviewers: **Saša Mladenović, Ph.D.**, *Professor at the Faculty of Science and Mathematics, University of Split*

Antonela Prnjak, mag. educ. inf., *Assistant at the Faculty of Science and Mathematics, University of Split*

Dino Nejašmić, mag. educ. math. et inf., *Lecturer at the Faculty of Science and Mathematics, University of Split*

Thesis accepted: **September, 2024**

IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam diplomski rad s naslovom METODE PENETRACIJSKOG TESTIRANJA KORIŠTENJEM METASPLOIT PLATFORME izradio samostalno pod voditeljstvom mag. educ. math. et inf. Dina Nejašmića. U radu sam primijenio metodologiju znanstvenoistraživačkog rada i koristio literaturu koja je navedena na kraju diplomskog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u diplomskom radu na uobičajen, standardan način citirao sam i povezao s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Student

Marin Maravić

ZAHVALA

Zahvaljujem se svom mentoru mag. educ. math. et inf. Dinu Nejašmiću na komentarima, stručnoj pomoći i svim savjetima tijekom izrade ovog diplomskog rada.

Zahvaljujem se svim bližnjima i prijateljima, posebno roditeljima, koji su mi pružali podršku i motivaciju tijekom cijelog studiranja i bili uz mene u najboljim i najgorim trenutcima.

Sadržaj

Uvod

Kibernetička sigurnost (engl. *cyber security*) je skup standarda, mjera i procesa koji se koriste za obranu raznih uređaja poput računala, servera, mobitela ali i mreža i podataka, od zlonamjernih napada treće strane. Primarni ciljevi sigurnosti su održavanje integriteta podataka, osiguranja privatnosti informacija korisnika i omogućavanje pristupa podacima samo ovlaštenim osobama. U današnjem vremenu kibernetičke prijetnje su u stalnom porastu i sve više su sofisticirane. Tome također pridonosi i rastuća ovisnost modernog društva o tehnologiji, pa čak i razvoj i korištenje umjetne inteligencije. Stoga je bitno osvijestiti se i znati poduzeti potrebne mjere kako ne bi došlo do neželjenih posljedica. U ovom radu definirat ćemo sami pojam kibernetičke sigurnosti, pokazati koje se platforme koriste u tom području ali i objasniti što je to penetracijsko testiranje, koje se metode koriste, kako se ono provodi te pokazati probijanje uz spomenute i opisane alate kao i usporedbu istih.

U ovom diplomskom radu bit će istražene metode i tehnike penetracijskog testiranja s posebnim naglaskom na korištenje Metasploita. Analizirat će se razne faze penetracijskog testiranja (primarno faze skeniranja, eksploatacije i post eksploatacije), uloga Metasploita u svakom od tih koraka te primjeri stvarnih scenarija u kojima je Metasploit korišten za identificiranje i eksploataciju ranjivosti. Kroz praktične primjere, rad će također prikazati kako Metasploit može biti korišten u stvarnim testiranjima, s ciljem pružanja jasnog uvida u način na koji ovaj alat doprinosi poboljšanju sigurnosti informacijskih sustava.

Cilj ovog rada je pružiti sveobuhvatan pregled penetracijskog testiranja koristeći Metasploit, te na taj način obogatiti znanje i praksu budućih sigurnosnih stručnjaka i istraživača u ovom važnom području.

1. Kibernetička sigurnost i penetracijsko testiranje

1.1. Osnove kibernetičke sigurnosti

Kibernetička sigurnost postala je pitanje globalnog interesa i važnosti. (Rossouw von Solms, 2013). Kibernetička sigurnost se, u većini literature, koristi kao sveobuhvatan pojam te njegove definicije variraju. Merriam Webster (Webster, n.d.) definira cyber sigurnost kao "mjere poduzete radi zaštite računala ili računalnog sustava (kao na Internetu) od neovlaštenog pristupa ili napada". Međunarodna telekomunikacijska unija (ITU) definira kibernetičku sigurnost malo drugačije. „Kibernetička sigurnost je skup alata, politika, sigurnosnih koncepata, sigurnosnih zaštita, smjernica, pristupa upravljanju rizicima, radnji, obuke, najboljih praksi, osiguranja i tehnologija koje se mogu koristiti za zaštitu kibernetičkog okruženja i imovine organizacije i korisnika.“ Von Solms i Van Rossouw (Rossouw von Solms, 2013) navode kako imovina organizacije i korisnika uključuje povezane računalne uređaje, osoblje, infrastrukturu, aplikacije, usluge, telekomunikacijske sustave i cjelokupnost prenesenih i/ili pohranjenih informacija u kibernetičkom okruženju. Kibernetička sigurnost teži osigurati postizanje i održavanje sigurnosnih svojstava imovine organizacije i korisnika protiv relevantnih sigurnosnih rizika u kibernetičkom okruženju, navode Von Solms i Van Niekerk. Opći sigurnosni ciljevi, po Von Solms-u i Van Niekerk-u, obuhvaćaju dostupnost, integritet (može uključivati autentičnost i neopozivost) i povjerljivost (ITU, 2008). Osiguranje podataka pojedinca i institucija na internetu je glavna svrha cyber sigurnosti, a njena pogreška može dovesti do ozbiljnih prijetnja jer netko s zlonamjernim namjerama može provaliti u uređaje preko mreže i preoteti (Wenye Wang, 2013) ili ukrasti korisničke podatke kao podatci o kreditnoj kartici ili lozinki korisničkog ID-a. Takvi napadi mogu uzrokovati financijsku štetu pojedincima, institucijama, velikim tvrtkama ali i državnim vladama. Danas cyber napadi nisu samo jednostavni napadi na računala, već i veliki poslovi koje podržavaju velike tvrtke i državne vlade, a njihovi napadi koštaju milijarde dolara (Aslan, 2024). Singh (Singh, 2022) navodi da kao etički haker i tester probijanja, najvažnije je biti ažuran i znati kako otkriti najnovije sigurnosne propuste. Zbog toga etički hakeri i testeri probijanja moraju biti opremljeni najnovijim znanjem, vještinama i alatima kako bi učinkovito identificirali i iskoristili skrivene sigurnosne propuste na ciljnim sustavima i mrežama. Također navodi da je važno razumjeti kako

prijetnje razmišljaju jer, iako testeri probijanja mogu imati sličan način razmišljanja, njihov je cilj otkriti i pomoći u rješavanju probijanja prije nego što dođe do stvarnog napada.

1.2. Platforma Metasploit

Platforma Metasploit (*engl. Metasploit Framework*) (David Kennedy, 2011) je platforma s otvorenim kodom koja pruža dosljednu, pouzdanu biblioteku s stalnim ažuriranim eksploatacijama. Metasploit nudi potpuno razvojno okruženje za izgradnju novih alata i automatizaciju svakog aspekta testiranja probijanja sustava. Posebnost Metasploit-a je što pruža drugačiju perspektivu o načinu provođenja i automatiziranja testova testiranja probijanja velikih razmjera. Metasploit Framework je poznat i po svojoj nestabilnosti, jer mu se kodna baza svakodnevno ažurira desetke puta zbog čega ima više od 3.000 različitih modula dostupnih za iskorištavanje različitih aplikacija, proizvoda i platformi, a taj se broj redovito povećava.

Metasploit (Rahalkar, 2017) je stvoren 2003. godine kada je H. D. Moore napisao prenosiv alat za mrežu koristeći Pearl, no do 2007. godine je prepisan u Ruby-u. Metasploit projekt je dobio značajno komercijalno pojačanje kada je tvrtka Rapid7 preuzela projekt 2009. godine, a u osnovi je snažan i fleksibilan okvir za testiranje sigurnosti sustava koji može obavljati sve zadatke koji su uključeni u ciklus testiranja. Metasploit nije aplikacija već platforma koja se može prilagoditi i proširiti prema korisničkim zahtjevima. Iako je vrlo moćan alat za testiranje sigurnosti, važno je razumjeti sposobnosti Metasploit-a kako bi se optimalno iskoristio.

Metasploit je namijenjen za istraživanja i testiranje probijanja. Jedini cilj za korištenje i stvaranje Metasploit-a je u dobrotvorne svrhe, tj. osiguravanje sigurnosti i integriteta poslužitelja. No nažalost, Metasploit je vrlo često korišten za hakiranje i iskorištavanje sustava radi financijske dobiti, ali i za namjerno nanošenje štete ciljanom poslužitelju. (SANS, n.d.) Velika evolucija tehnika napada stvorila je nove načine napada koji se temelje na skriptama što predstavlja veliku prijetnju mnogim sustavima. S ovakvom visokom produktivnošću, navode Gupta i Kumar (Gupta & Kumar, 2015), hakeri sada mogu lako razviti rane napadačke skripte za iskorištavanje postojećih ili čak novih ranjivosti.

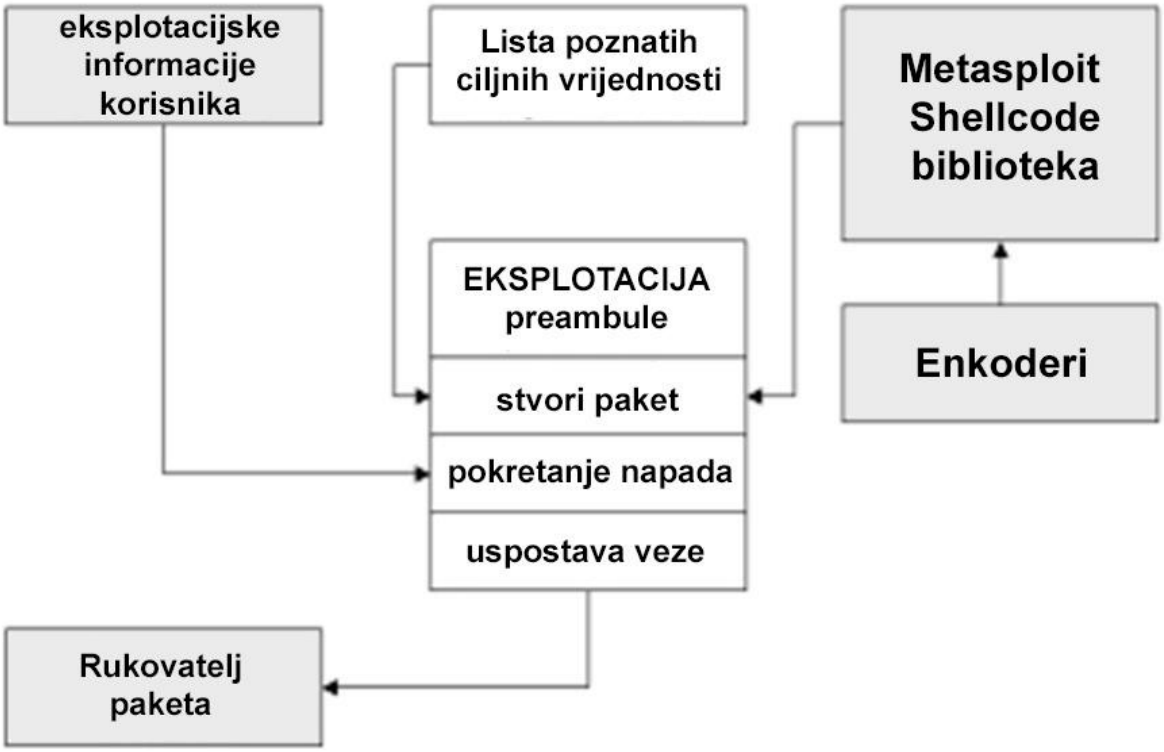
Napadačke skripte izvode napad kroz četiri ključna koraka:

1. Skripta istražuje verziju i okolinu izvršavanja putem weba.

2. Na temelju prikupljenih rezultata, i kroz informacijsku bazu skripta, identificira vrlo specifičnu ranjivost koja postoji na cilju, a često sadrži informacije kao npr. povratnu adresu ciljanog sustava kojeg želi napasti.

3. Zatim sastavlja paket koji se koristi za napad, prilagođen ciljnom sustavu.

4. Na kraju prosljeđuje paket ciljnom sustavu radi iskorištavanja otkrivene ranjivosti



Slika 1.1 Rad Metasploit-a

Slika 1.1 prikazuje rad Metasploit platforme. Informacije o ranjivosti koje korisnik dostavi koriste se za pokretanje napada stvaranjem veze i izvođenjem napada. Metasploit generira *shellcode* (koristeći Metasploit Shellcode Library) koristeći parametre koji su specificirani. Ovisno o strategiji i vrsti ranjivosti, različite skripte mogu imati različita ponašanja napada prilikom izvođenja navedenih koraka. Na primjer, napad pomoću *brute force* metode može neprestano stvarati i slati napadačke pakete s pretpostavljenim vrijednostima dok se ciljni sustav ne hakira ili kompromitira, dok *stealth* napad također može očistiti tragove u zapisniku (engl. *Log*) ciljnog sustava nakon isporuke paketa (engl. *Payload*). Stvaranje i slanje napadačkih paketa ključni su koraci za pokretanje bilo kojeg napada. Napadački paket obično je niz bajtova koji se sastoje od sljedećih elemenata: (a) posebnih bajtova koji mogu

iskoristiti određenu ranjivost; (b) *shellcode* koji hakeri ili testeri izvršavaju nakon što se ranjivost iskoristi odnosno eksploatira; (c) nasumično punjenje (na primjer NOP 0x90). NOP sled je jednostavna '*no operation*' instrukcija unutar arhitekture procesora. Prilikom prepunjivanja spremnika (engl. *Buffer*) koristi se za alokaciju prostora prije samog paketa, kako bi se omogućila pouzdana adresa za povratak u memoriju (umjesto poznavanja točne lokacije početka paketa dovoljno je odraditi NOP sled instrukciju kako bi se vratili na početak paketa) ili za usklađivanje registara. To omogućuje robusnost napadačkog paketa; (d) bajtovi formata potrebnih za mrežne protokole. Pomoću biblioteka skriptnih jezika i ugrađenih modula koje pruža platforma za napade na webu, napadačka skripta poziva API-je povezanih biblioteka ili modula kako bi pomogao u izvođenju svakog koraka, posebno u stvaranju napadačkog paketa.

```
1.  exploit_def()
2.  connection()
3.  exploit_preamble =  "\x00\x00\x01"
4.  version_find =  probing_ver()
5.  if  (version  equals  5)
6.  attack_payload =  prepare_payload5()
7.  else
8.  attack_payload =  prepare_payload4()
9.  end
10. exploit_preamble <<  payload_length
11. socket.put(exploit_preamble) //Reqd by the protocol
12. socket.get_once()
13. socket.put(attack_payload)     //sendingthe     attack

                payload
14. socket.get_once()
15. ... #  triggering vulnerability
16. end
17. def prepare_payload5()
18. attack_payload =  shellcode
19. attack_payload << rand_alpha(payload.length)
20. attack_payload <<  "\x010" +  [-117].pack("X")
21. attack_payload <<  "\xe\xf"
22. attack_payload   <<      get_target_ret(5)   //   Target

                  Version: 5
23. attack_payload <random_alpha(409)
24. return attack_payload
25. end
```

Slika 1.2 Isječak koda iz skripte za Metaploit

Slika 1.2 prikazuje isječak iz stvarne Metasploit skripte koji eksploatira ranjivu aplikaciju. U ovom primjeru, skripta ima dvije metode. **exploit_def()** je glavna metoda koja provodi korake za pokretanje napada. **prepare_payload5()** je jedna od metoda koja sastavlja napadački paket. Kada se ova skripta pokrene na Metasploit okviru, prvo se povezuje s ciljnim sustavom putem mreže (2. linija), a zatim otkriva verziju ciljnog sustava (4. linija). Ovdje su i **connection** i **probing_ver** metode ugrađenog modula za mrežni protokol. Na temelju otkrivene verzije, zatim poziva odgovarajuću metodu za stvaranje napadačkog paketa specifičnog za tu verziju cilja (linije 5-9). Kada se pozove **prepare_payload5()**, paket se prvo dodjeljuje modulu *shellcode*, koji vraća konfigurirani *shellcode* (18. linija)1.

Shellcode se može neovisno odabrati. Modul *shellcode* nudi mnogo *shellcodeova* za različite svrhe. Paketu se zatim dodaje (<<) s navedenim sadržajem (linije 19-23). Nasumično generirano slovno punjenje generira se pomoću *random alpha* kako bi se proširio paket na potrebnu veličinu mrežnog protokola koji se koristi. Konkretni bajtovi predstavljaju asemblerski kod koji ide u *shellcode*. **pack("X")** pretvara cijeli broj u bajt kao pomak za jedan JMP. **Get_target_ret** je još jedan API okvira za napade koji upita skriptnu informacijsku bazu. Nakon što je paket stvoren, skripta šalje paket **exploit_preamble** ciljnom sustavu, a zatim paket za eksploataciju ranjivosti (linije 11-13) (Gupta & Kumar, 2015).

1.3. Penetracijsko testiranje

Penetracijsko testiranje (Denis, Zena, & Hayajneh, 2016) je simulacija napada radi provjere sigurnosti sustava ili okoline koja se analizira. Ovaj test može se provesti putem fizičkih sredstava koristeći hardver ili putem socijalnog inženjeringa, a cilj mu je istražiti ponašanje sustava, mreža ili uređaja u ekstremnim okolnostima kako bi se identificirale njihove slabosti i ranjivosti. Što se tiče alata, postoje alati za testiranje probijanja koji jednostavno analiziraju sustav, kao i oni koji zapravo napadaju sustav kako bi pronašli ranjivosti. Kroz analogiju, testiranje probijanja bi bilo angažiranje nekoga da stvarno pokuša provaliti u kuću kako bi se otkrile sigurnosne mane i slabosti kuće. Testiranje probijanja se može automatizirati pomoću softverskih aplikacija ili ručno, a sam proces uključuje prikupljanje informacija o ciljnom sustavu prije testiranja, identificiranje mogućih točaka ulaska, pokušaj provaljivanja i izvještavanje o rezultatima.

Penetracijsko testiranje nije samo pokretanje niza od nekoliko automatiziranih alata nad nekim ciljem. To je cjelovit proces koji uključuje više faza, a svaka je faza jednako važna za uspjeh testiranja, odnosno obavljanja cilja testiranja. Za obavljanje svih zadataka u svim fazama penetracijskog testiranja, morali bismo koristiti različite alate i možda bismo trebali izvršiti neke zadatke ručno. Zatim trebamo kombinirati rezultate dobivene pomoću različitih alata zajedno kako bi proizveli jedno smisleno izvješće. Ovo je svakako zastrašujući zadatak. Bilo bi stvarno jednostavno i uštedjelo bi vrijeme da postoji samo jedan alat pomoću kojeg bi mogli izvršiti sve potrebne zadatke za penetracijsko testiranje (Sagar Rahalkar, 2019). Točno ova potreba zadovoljava platforma kao što je Metasploit.

Glavni cilj testiranja probijanja je utvrditi sigurnosne slabosti, ali se ono, također može koristiti za:

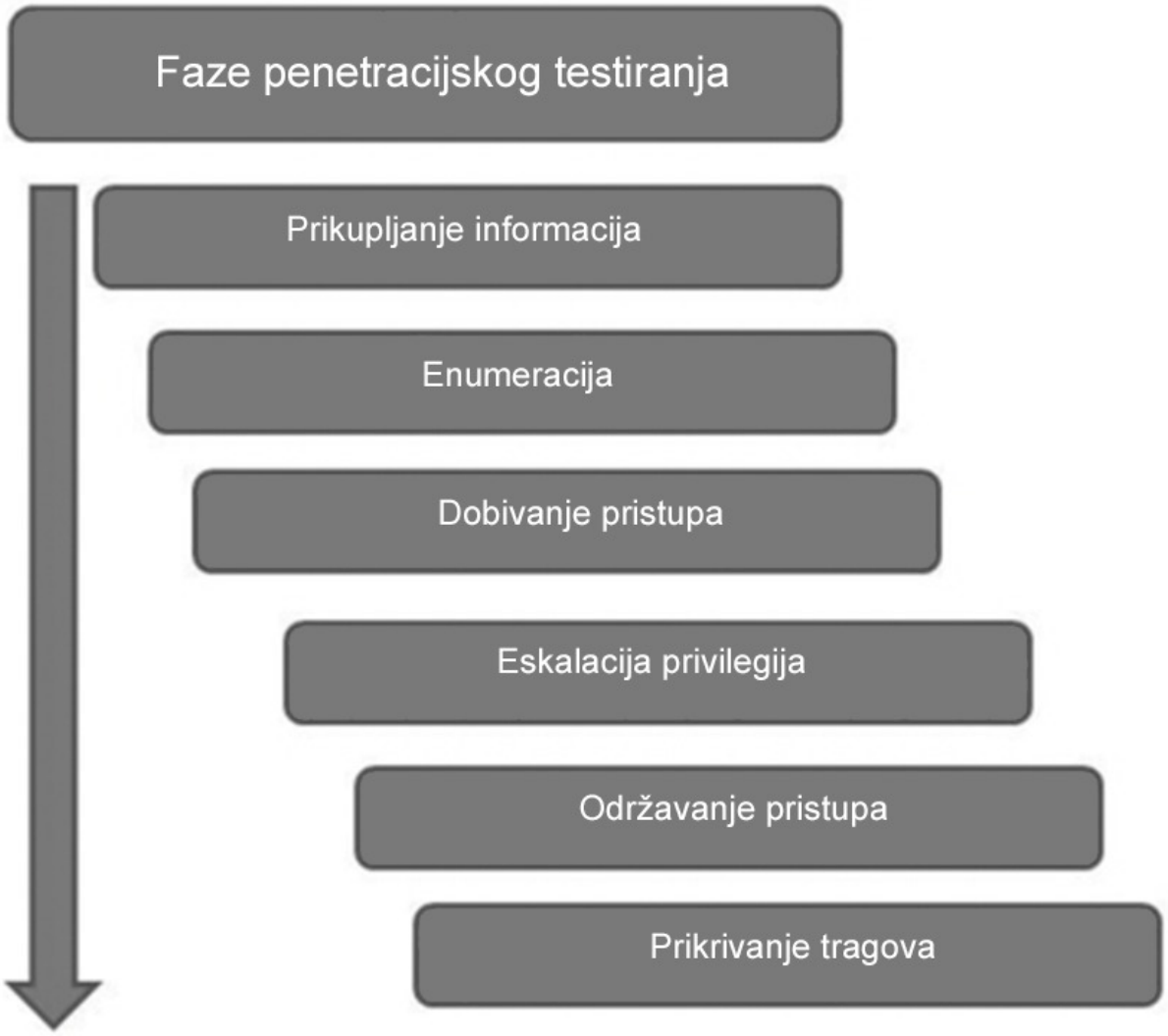
1. testiranje usklađenosti sigurnosne politike organizacije,
2. testiranje svjesnosti sigurnosti zaposlenika,
3. testiranje sposobnosti organizacije da odgovori na sigurnosne incidente.

Postoje četiri tipične vrste penetracijskog testiranja:

- vanjsko testiranje,
- unutarnje testiranje,
- slijepo testiranje
- dvostruko slijepo testiranje.

Vanjski test cilja na vanjski vidljive poslužitelje ili uređaje tvrtke, poput poslužitelja imenskog prostora (DNS), poslužitelja e-pošte, web poslužitelja ili vatrozida. Cilj u ovom slučaju je saznati može li vanjski napadač dobiti neovlašteni pristup i kakvu razinu pristupa može dobiti. Unutarnje testiranje simulira unutarnji napad iza vatrozida od strane ovlaštenog korisnika s standardnim pristupnim privilegijama. Slijepo testiranje simulira postupke i postupke pravog napadača ograničavajući strogo informacije dane osobi ili timu koji izvodi test unaprijed. U dvostrukom slijepom testiranju, slijepo testiranje ide još dalje, tako da samo nekoliko osoba unutar organizacije zna da se test provodi.

Proces testiranja sigurnosti sustava se sastoji od 6 ključnih koraka prikazanih na Slika 1.3.



Slika 1.3 Faze penetracijskog testiranja

Prvi korak je prikupljanje informacija. Ova faza uključuje prikupljanje što više podataka o ciljnom sustavu ili mreži. Ovo je najvažnija faza bilo kakvog testiranja ili hakiranja, što je više informacija poznato o meti, lakše je iskoristiti njegove ranjivosti. Ova faza može biti pasivna, koristeći javno dostupne izvore poput društvenih mreža i korištenjem tražilica poput Google pretraga, ili aktivna, uključujući specijalizirane alate poput skenera vrata i socijalnog inženjerstva (*engl. Social engineering)*.

Numeriranje i skeniranje je slijedeći korak nakon prikupljanja informacija, gdje numeriranje pomaže identificirati određene usluge, korisnike, zajedničke resurse i DNS unose na ciljnom sustavu/mreži, pružajući jasniji uvid u njegovu strukturu. Ova faza je direktnija od faze skeniranja jer pruža informacije koje se ne mogu dohvatiti u toj fazi. Neke od tehnika koje se koriste u fazi skeniranja i enumeracije su: provjera vatrozida i pravila, provjera otvorenih

portova, provjera pokrenutih servisa, provjera sigurnosnih ranjivosti i kreiranje topologije mreže mete.

Treća faza, odnosno faza dobivanja pristupa, iskorištava ranjivosti identificiranih tijekom prethodnih faza kako bi se ušlo u ciljni sustav, potencijalno zaobilazeći sigurnosne kontrole poput antivirusa i vatrozida. Ova faza može biti najzahtjevnija. Uspješnom eksploatacijom ranjivosti sustava može se dobiti udaljeno izvršavanje malicioznog koda, te time i udaljeni pristup nad ciljanim sustavom. Dobivanjem pristupa može se provoditi probijanje raznih lozinki na sustavu, eksploatacija drugih ranjivosti, eskalacija privilegija i sakrivanje datoteka. Eskalacija privilegija povećava pristupačnosti kako bi se stekla potpuna kontrola nad sustavom (administratorka prava), često neophodna za dublju infiltraciju i eksploataciju ostatka sustava. Ono što čini fazu dobivanja pristupa najzahtjevnijom je to što eksploatacija može raditi na jednom sustavu, ali ne i na nekom drugom. Točnije mogućnost eksploatacije varira od sustava do sustava. Jednom kada je eksploatacija uspješna i pristup sustavu je dobiven, iduća faza je očuvanje tog pristupa.

Nakon uspješne eksploatacije i dobivanja pristupa nekom sustavu, treba osigurati pristup tom sustavu održavanje pristupa. Faza održavanja pristupa uspostavlja trajni pristup ciljnom sustavu kako bi se osiguralo neprekidan pristup sustavu čak i nakon ponovnog pokretanja, odnosno dok god je sustav uključen. To se provodi kreiranjem takozvanog *backdoor* pristupa u sustav koji omogućuje stvaranje više trajnih veza između napadačkog računala i sustava žrtve. Ciljevi faze održavanja pristupa su: kretanje dublje u sustav, eksfiltracija podataka i stvaranje *backdoor-a* i trajnih veza.

Posljednji korak u penetracijskom testiranju je prikrivanje tragova odnosno brisanje svih tragova upada u sustav. To je nužno kako bi se izbjeglo otkrivanje upada od strane sigurnosnih sustava poput antivirusa i vatrozida, te timova za odgovor na incidente, u skladu s uvjetima ugovora o testiranju penetracije. Prikrivanje tragova osigurava da ne ostaju tragovi prisutnosti na mreži, budući da je sama ideja penetracijskog testa osmišljena tako da bude skriven i simulira napad kakav bi se mogao dogoditi nad organizacijom.

1.4. Zaštita od probijanja sustava pomoću Metasploit-a

Napadački okviri (*frameworks*) pružaju mnogo ugrađenih modula koji obuhvaćaju različite mrežne protokole, operacijske sustave i pružaju mnogo *shellcode* i NOP (no operation) instrukcija. To omogućava hakerima brzo razvijanje ranih napadačkih skripti za eksploataciju različitih meta. Napredni hakeri mogu stvoriti i složenije napadačke skripte koje mogu imati mnoge putanje izvršenja, različite napadačka ponašanja i različite vrste napadačkih paketa (*payload*-a). Neki od njih se mogu pokrenuti samo u jedinstvenim uvjetima napada. Zbog toga, Gupta i Kumar u svom radu (Gupta & Kumar, 2015) predlažu sistem u kojem se pruža rješenje za sustave koji su ranjivi na eksploatacijske pakete od Metasploita. Ideja je da sustav bude prva pomoć za ranjivosti čije sigurnosne zakrpe nisu ažurirane ili dostupne, a već postoji napadačka skripta za njih. Sustav bi evaluirao i identificirao potpise napadačkih skripti s web stranice „exploit-db.com“, odredio koji je izvorni i odredišni port napadačkog paketa. Također bi u potpisima uključivao i očekivani kod koji se može izvršiti nad metom. Kombinacijom tih informacija, pripremila bi se defenzivna skripta. Sustav bi u principu bio mrežni monitor koji koristi Metasploit za identifikaciju ranjivosti i potpisa napadačkih skripti koje se pokušavaju izvršiti.

Iako sustav koliko je trenutno poznato nije još u komercijalnoj funkciji, napretkom tehnologije, specifično umjetne inteligencije (A.I.) zadnjih nekoliko godina, vrlo je vjerojatno da već postoji neka instanca ovog sustava za vojne svrhe. Na to upućuje razvoj umjetne inteligencije za kiber obranu od strane američke vlade u suradnji s Europskom unijom pod imenom AICA (*Autonomous Intelligent Cyber Defense Agent*), s toga je samo pitanje vremena kada će takav jedan sustav biti pušten u globalnu upotrebu.

1.5. Pravni i etički aspekti

Iako je penetracijsko testiranje izuzetno korisno za poboljšanje sigurnosti, važno je pridržavati se pravnih i etičkih smjernica kako bi se osiguralo da se testiranje provodi odgovorno i zakonito, jer se za samo testiranje koriste isti alati koje bi koristili i napadači, odnosno hakeri, da je riječ o pravom napadu.

Prije početka bilo kakvog testiranja potrebno je osigurati pisani sporazum s vlasnikom sustava o samom provođenju testiranja. Sami sporazum treba sadržavati definirane testne opsege, metode koje će se koristiti i očekivano trajanje testa. Bilo kakvo penetracijsko testiranje mreže ili sustava se nikada ne bi smjelo provoditi bez izričite dozvole vlasnika. Provođenje testiranja bez dozvole može se smatrati nezakonitim pristupom računalnim sustavima (hakiranje) i može biti kažnjivo, kako novčano tako i zatvorski. Postoji šansa da penetracijsko testiranje uzrokuje nenamjernu štetu u sustavu ili računalu na kojem se provodi. Zbog toga bi sporazum trebao sadržavati i jasno definiranu odgovornost za bilo kakvu nastalu štetu koja može potencijalno nastati tijekom testiranja. Neke tvrtke koje nude usluge penetracijskog testiranja nude osiguranje koje pokriva štetu koja bi mogla nastati tijekom testiranja.

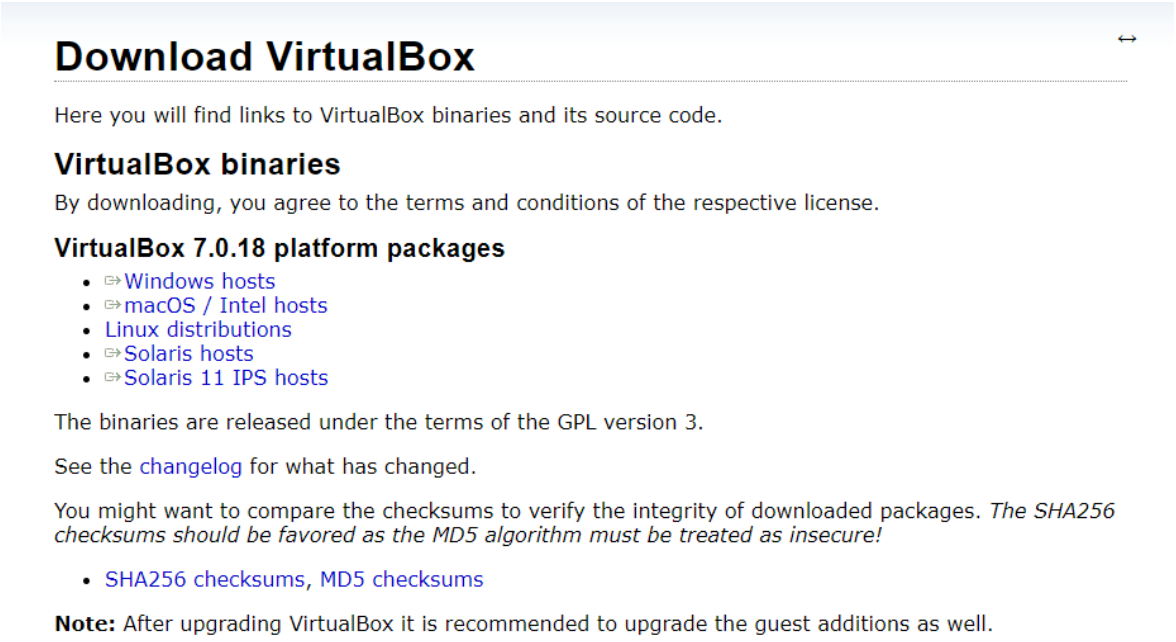
Različite zemlje i regije imaju različite propise i zakone vezane za računalnu sigurnost i privatnosti, stoga je važno poznavati i pridržavati se tih zakona jurisdikciji u kojoj se provodi testiranje. U nekim slučajevima, penetracijsko testiranje može sadržavati obradu nečijih osobnih podataka. Zbog toga je potrebno držati se i zakona o zaštiti podataka, u slučaju Europske unije, držati se GDPR-a.

Penetracijski testeri moraju slijediti etički kodeks ponašanja, poput onih koje definiraju profesionalne organizacije kao EC-Council i ISACA. Povjerljivost i diskrecija su ključni prilikom provođenja testiranja. Sami testeri trebaju biti svjesni povjerljivih informacija koje bi mogli susresti prilikom testiranja i osigurati da te informacije ostanu zaštićene. Redovita i jasna komunikacija s klijentom je ključna. Klijent mora biti redovito obaviješten o napretku i bilo kakvim problemima koji se pojave. Nakon završetka testiranja, potrebno je izraditi detaljan izvještaj koji uključuje otkrivene ranjivosti, metode koje su bile korištene za otkrivanje tih ranjivosti i preporuke za njihovo otklanjanje. Testeri trebaju biti nepristrani i objektivni u svom radu i ne smiju iskoristiti svoj pristup sustavima za neetičke aktivnosti ili osobnu korist.

2. Metodologija penetracijskog testiranja

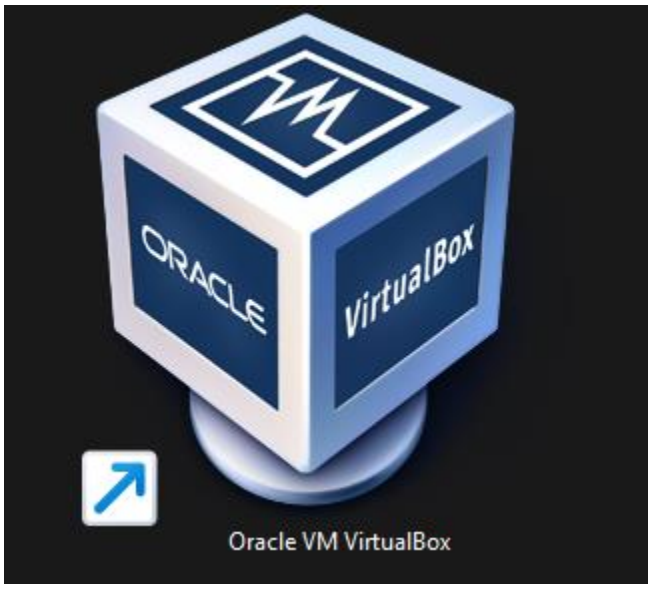
2.1. Priprema okruženja

Kako bi mogli provesti bilo kakva testiranja, prvo što je potrebno je pripremiti okruženje. Za to će nam biti potrebno nekoliko stvari. Prije svega, želimo da testovi budu provedeni u sigurnom okruženju. Za to će nam trebati softver za virtualizaciju u čijem se okruženju mogu sigurno provoditi testiranja. U svrhu izrade ovog rada bit će korišten Oracle VM VirtualBox. Program se može besplatno preuzeti sa službene stranice https://www.virtualbox.org/ gdje se također nalazi i sva dokumentacija vezana za VirtualBox i njegovo korištenje.

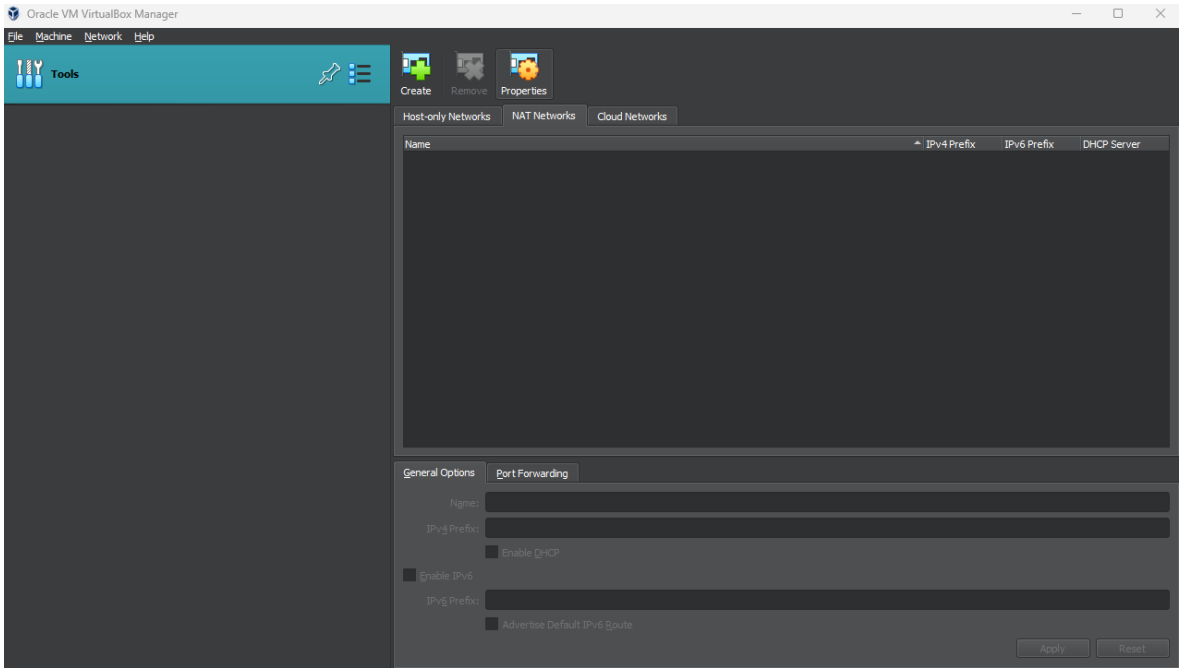


Slika 2.1 Preuzimanje Virtualboxa-a

Na stranici, klikom na „Downloads“ odvodi nas na stranicu za preuzimanje prikazanu na Slika 2.1. Potrebno je odabrati pripadajući operacijski sustav za instalaciju. U ovom slučaju, pošto se koristi Windows 11 operacijski sustav, odabire se „Windows host“ i automatski se preuzima instalacijski paket. Dvostrukim klikom na preuzeti „.exe“ započinje se instalacija VirtualBoxa. Za uspješnu instalaciju potrebno je samo pratiti upute koje su prikazane. Završetkom instalacije VirtualBox se pokreće dvostrukim klikom na ikonu prikazanu na Slika 2.2.



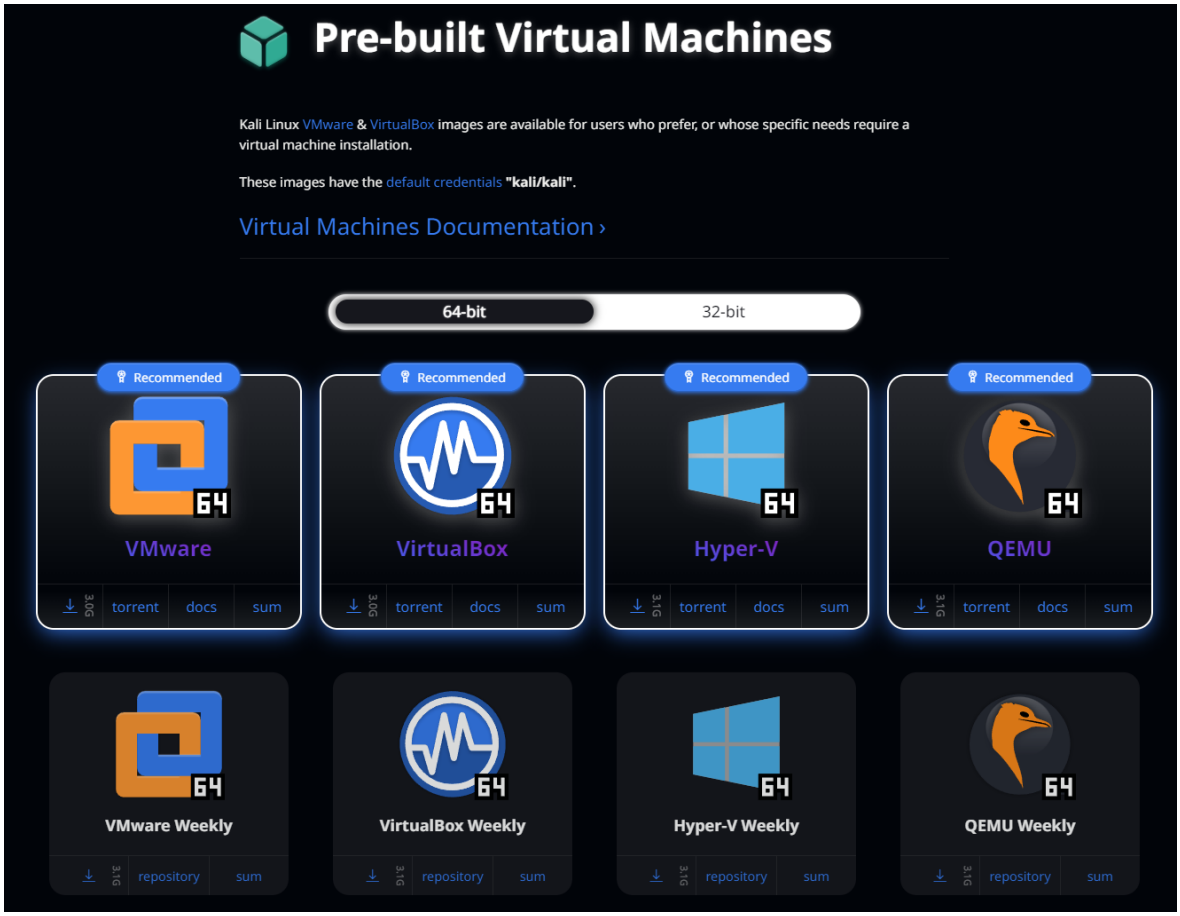
Slika 2.2 Ikona VirtualBoxa



Slika 2.3 Sučelje VirtualBox-a

Sučelje VirtuaBoxa koje vidimo na Slika 2.3, je podijeljeno na dvije strane. Na lijevoj strani će se nalaziti lista svih virtualnih računala, dok na desnoj će se nalaziti postavke i opcije za svaki od tih virtualnih računala. Trenutno na desnoj strani se nalaze samo postavke virtualne mreže s obzirom da nije dodano niti jedno virtualno računalo. Da bi dodali neko virtualno računalo treba odabrati „Machine“ i zatim dodati već postojeći sa „Add“ ili kreirati novi sa „New“.

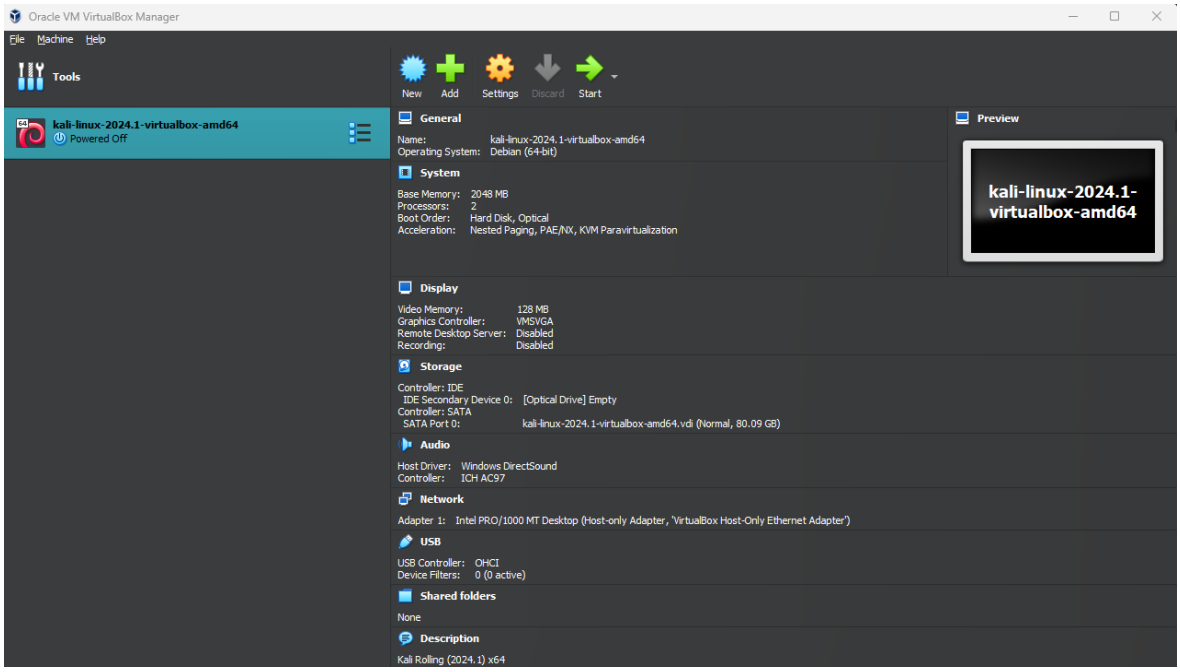
Prvo virtualno računalo je ono koje ćemo koristiti za samo provođenje napada i testiranje. To će biti računalo koje na sebi ima Kali Linux distribuciju. Za preuzimanje virtualnog računala dovoljno je preuzeti je sa službene stranice Kali Linux-a. Kao i velika većina Linux distribucija, Kali je također besplatan i može ga se preuzeti sa službene stranice https://www.kali.org/. Odlaskom na službenu stranicu i pritiskom na „Download“ dolazimo do izbora za koju točno platformu preuzimamo Kali. S obzirom da radimo s VirtualBox-om, izbor je „Virtual machines“



Slika 2.4 Odabir virtualnog računala

Klikom na „Virtual machine“ dobivamo nekoliko opcija za preuzimanje već gotovih virtualnih računala za svaki od popularnijih virtualnih okruženja, kao što je prikazano na Slika 2.4. S obzirom da koristimo VirtualBox, odabiremo njega kao opciju klikom na „download“ ikonu ili preuzimanje putem torrenta. Završetkom preuzimanja dobiva se „.rar“ datoteka koju je potrebno raspakirati. Sada koristeći VirtualBox moramo otvoriti Kali Linux. To radimo na naći da u VirtualBox-u odaberemo „Machine“ pa „Add“. Sada treba pronaći

raspakirano virtualno računalo koje bi trebalo imati naziv poput „kali-linux-2024.1-virtualbox-amd64“. Odabirom datoteke virtualno računalo se dodaje na listu unutar VirtualBoxa.

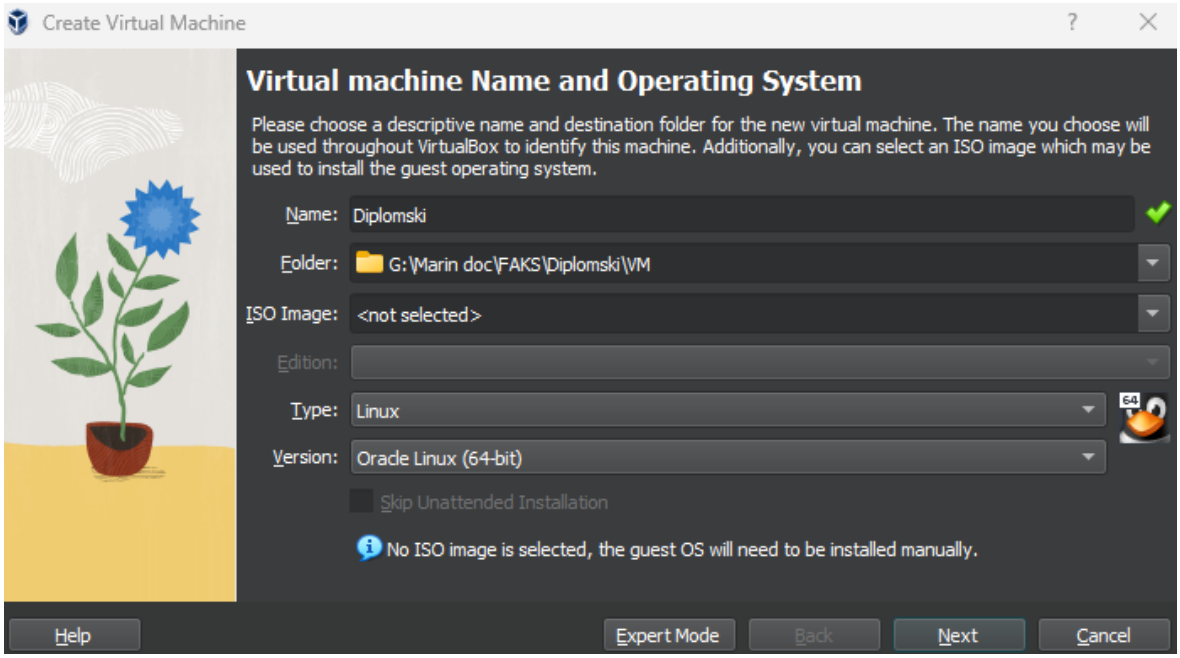


Slika 2.5 VirtualBox s dodanim Kali Linux-om

Sada na desnoj strani VirtualBoxa možemo vidjeti podatke koje do sada nismo mogli vidjeti. To uključuje koliko VM može koristiti RAM memorije, koliko jezgri procesora, količina video memorije, mrežni adapter i prostor na tvrdom disku. Uz to imamo pristup postavkama s opcijom „Options“ i gumb za pokretanje virtualnog računala.

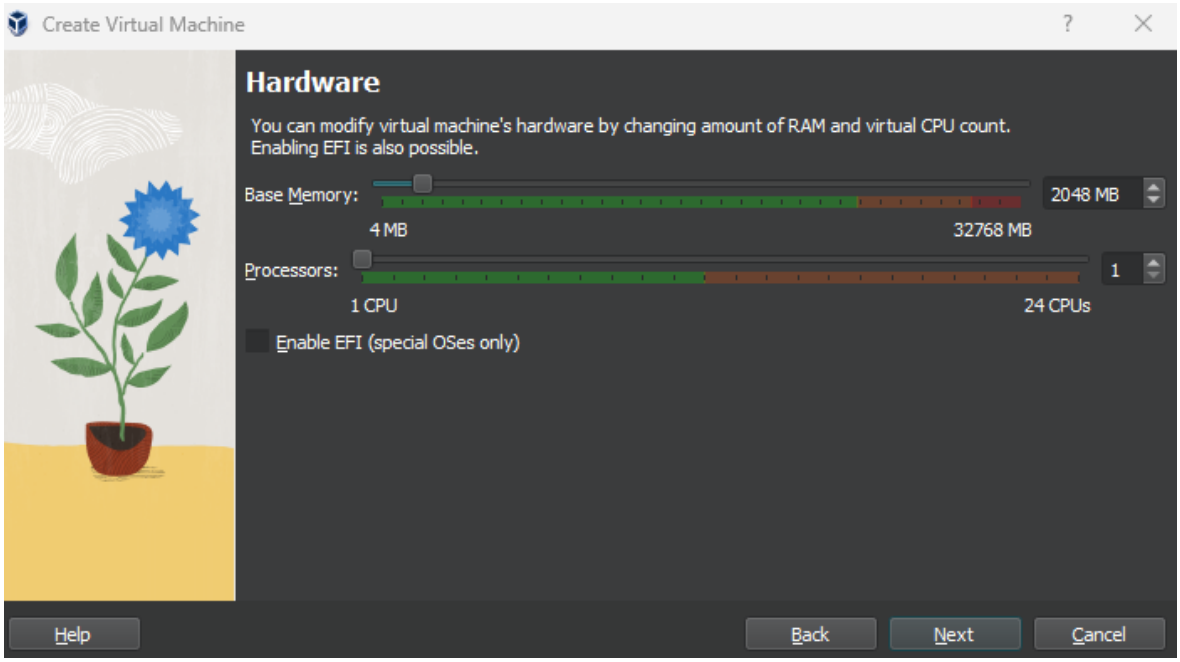
Želimo imati i računala koje će biti mete napada, odnosno testiranja. Za to će se koristiti specijalna distribucija Linuxa pod imenom Metasploitable. Metasploitable je testno okruženje koje pruža sigurno mjesto za provođenje penetracijskog testiranja i sigurnosnog istraživanja. Drugim riječima, to je Linux distribucija koja je namjerno napravljena da bude ranjiva i koji ima već otvorene neke mrežne portove kako bi se provodili napadi. Iako je moguće testove i napade provoditi na normalnim operacijskim sustavima, Metasploitable je odabran zbog lakšeg rada i provođenja samih testiranja. Metasploitable je dostupan za besplatno preuzimanje sa službene stranice https://docs.rapid7.com/metasploit/metasploitable-2/. Procedura za njegovo namještanje je ista kao i za Kali Linux. Potrebno ga je preuzeti sa službene stranice, raspakirati arhivu, a zatim unutar VirtualBox-a dodati Metasploitable na listu virtualnih računala. Taj proces će

sada biti malo drugačiji jer nije već pripremljeno virtualno okruženje. Potrebno je unutar VirtualBoxa otići na „File“ pa zatim na „New“ kako bi kreirali novo virtualno okruženje.



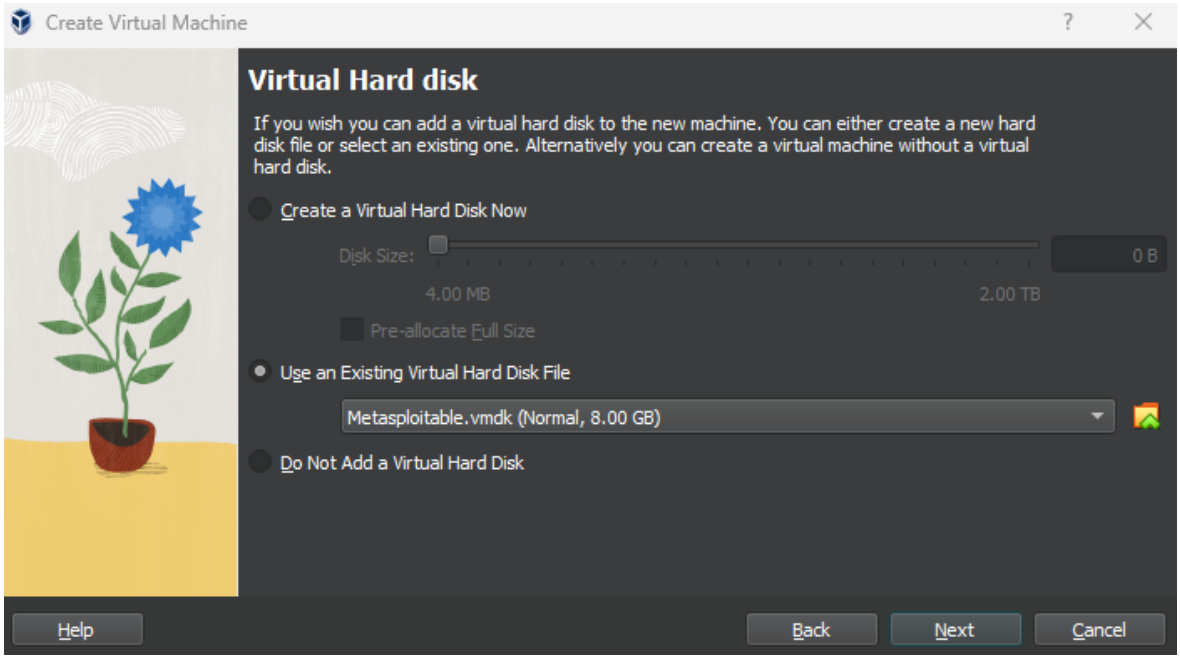
Slika 2.6 Kreiranje VM-a

Slika 2.6 prikazuje prozor koji se prikaže klikom na „New“ unutar VirtualBox-a. Potrebno je odabrati na koju ćemo lokaciju spremiti VM i na kojem operacijskom sustavu se bazira, a to u našem slučaju je Linux.



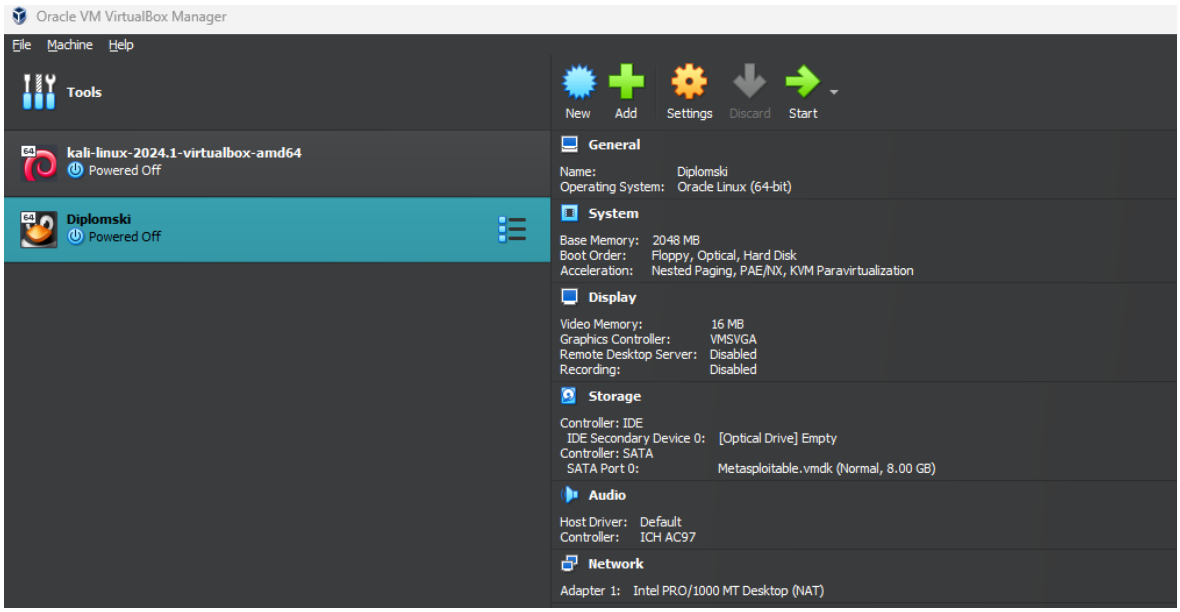
Slika 2.7 Dodjela Resursa VM-u

Klikom „Next“ dolazimo do dodjele resursa VM-u, kao što se može vidjeti na Slika 2.7. S obzirom da Metasploitable sam po sebi ne zahtjeva puno memorije ni jak procesor kako bi radio, dovoljno je ostaviti početnu dodjelu memorije i procesorskih jezgri i kliknuti „Next“. Idući korak je kreiranje virtualnog tvrdog diska i tu dolazimo do onog što smo preuzeli sa stranice.



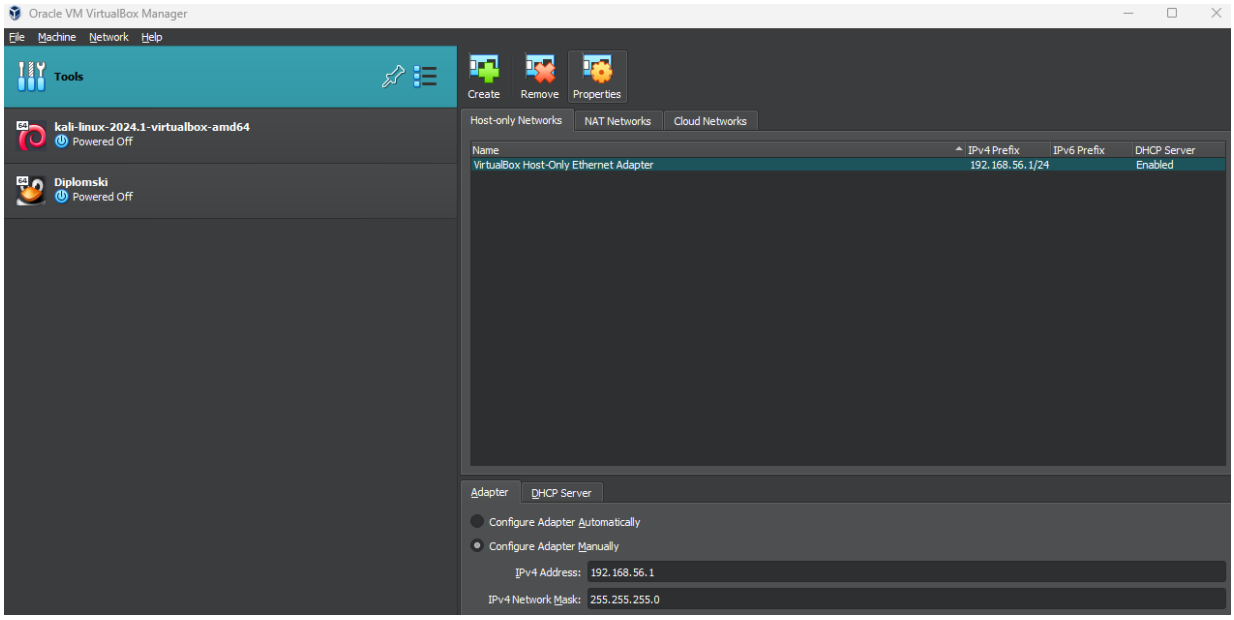
Slika 2.8 Kreiranje virtualnog diska

Želimo odabrati već postojeći virtualni disk, onaj koji smo preuzeli, a ne kreirati novi. Stoga moramo odabrati opciju „Use an Existing Virtual Hard Disk File“ kao što možemo vidjeti na Slika 2.8 poviše. Za kraj nam se prikaže sažetak VM-a kojeg kreiramo i sami postupak je gotov.



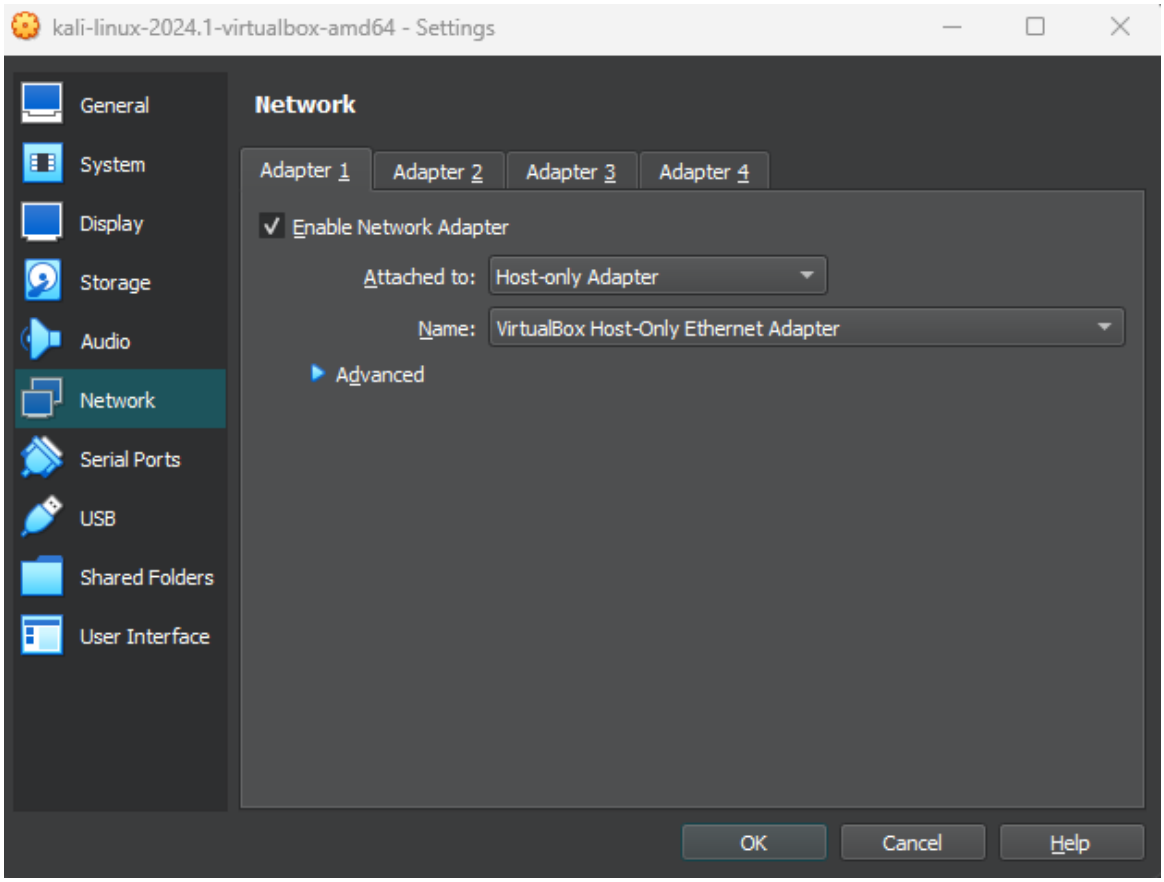
Slika 2.9 VirtualBox nakon dodavanja VM-a

Zadnja stavka koja nam je potrebna je postavljanje mreže na kojoj će se nalaziti Kali Linux i Metasploitable. S obzirom da bavimo s nešto manje legalnim radnjama, ne želimo da virtualna računala, točnije Kali Linux ima pristup vanjskom internetu, dok Metaspoitable je već laka meta i njegovo spajanje na vanjski Internet može ugroziti osobnu LAN mrežu kao i sve povezane uređaje na njoj. Stoga moramo kreirati mrežu samo za virtualne uređaje. To ćemo napraviti tako da unutar VirtualBox-a pod „Tools“ opcijom odaberemo „Host-only Networks“ i kreiramo novi ako već ne postoji.



Slika 2.10 Host-only mreža

Sada je potrebno otići na svaki od virtualnih uređaja i pod „Settings“ i „Network“ za „Attached to:“ postaviti da bude na „Host-only Adapter“, odnosno na mrežu koju smo prethodno kreirali. Treba napomenuti kako postoji mogućnost ovisno o verziji Kali Linuxa da je potrebna Internet veza kako bi preuzeo potrebne pakete za pokretanje određenih funkcionalnosti. Za to je dovoljno prije promijene na Host-only mrežu ostaviti ga na početnoj vrijednosti i pokrenuti virtualno računalo i po potrebi Metasploit.



Slika 2.11 Host-only Adapter

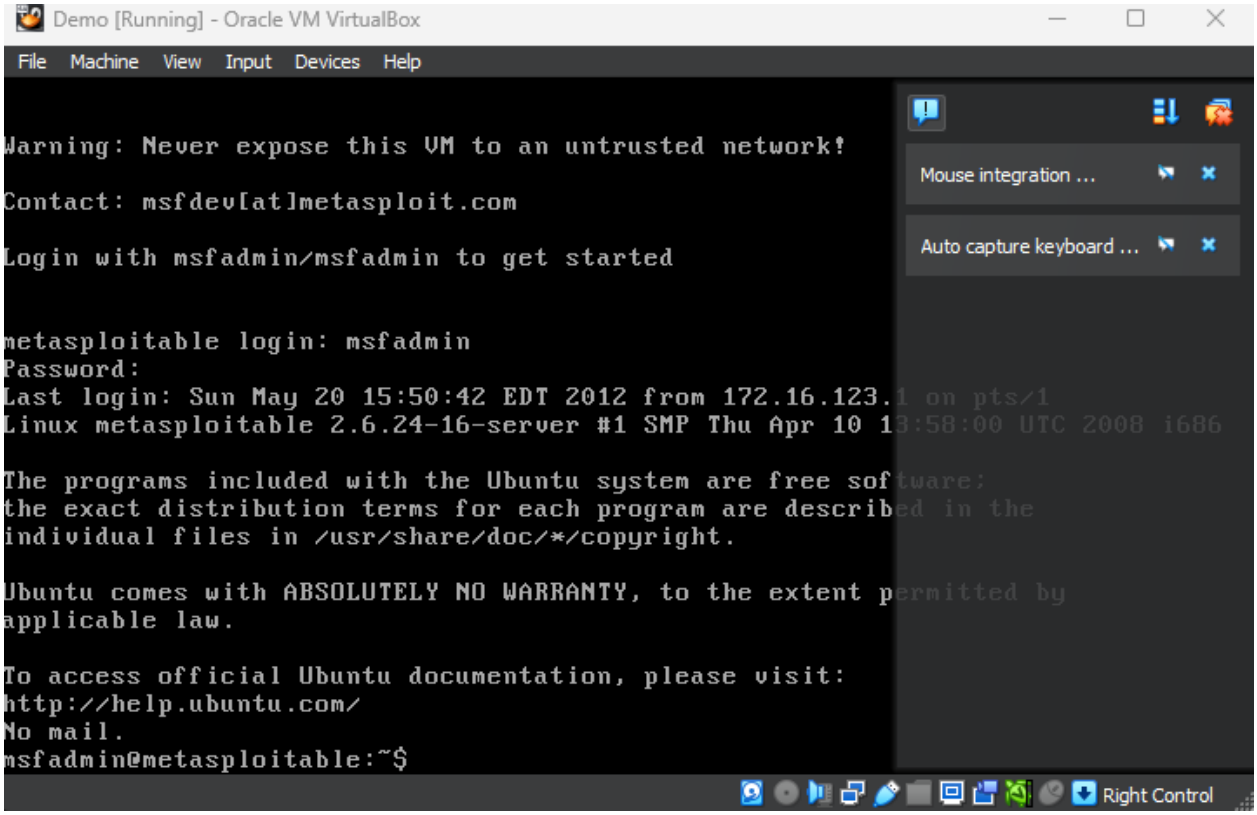
Namještanjem mreže za virtualne uređaje, postupak postavljanja okoline je gotov i možemo prijeći na fazu skeniranja i izviđanja računala na mreži.

2.2. Skeniranje i izviđanje

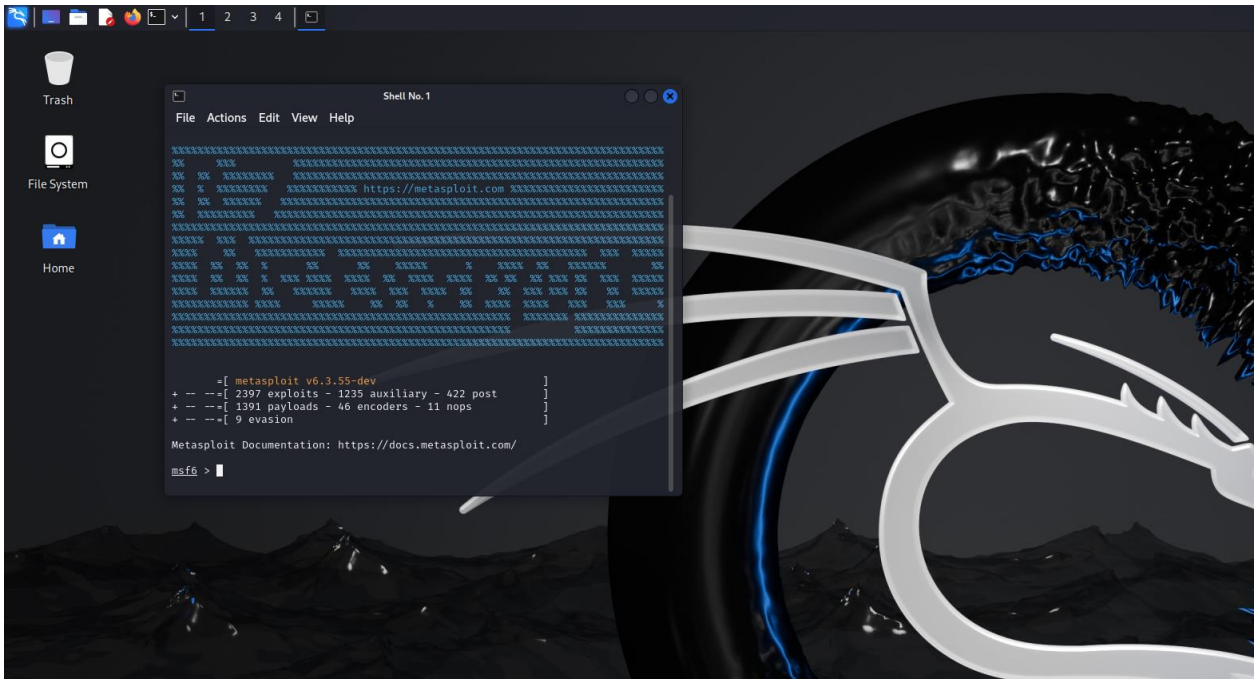
Prvi korak prije bilo kakvog testiranja ili napada treba biti skeniranje i izviđanje. Priprema i poznavanje mete čini veliku većinu posla kada je u pitanju hakiranje bilo kakvog tipa. Želimo saznati koji su sve uređaji povezani na mrežu, koje sve usluge koriste i koje su verzije tih usluga te statuse portova tih uređaja, odnosno jesu li otvoreni ili zatvoreni. Cilj ove faze je pronaći moguće ranjivosti uređaja koji su spojeni na mrežu kako bi mogli iskoristiti primjerene alate za eksploataciju ranjivosti. Alati se biraju ovisno o uslugama koje ti uređaji koriste, verzijama tih usluga i statusima portova. Na primjer ako neka usluga koristi uslugu „ftp“ (engl. *file transfer protocol*), tražit ćemo alate koji su napravljeni za eksploataciju ftp usluge pod istom verzijom kao što je i potencijalna ranjivost na uređaju koji je povezan na mrežu.

Samo skeniranje i izviđanje se vrši alatom zvanim „nmap“ koji dolazi instaliran s Kali Linux operacijskim sustavom. Nmap je softver odnosno aplikacija za skeniranje i pronalazak otvorenih portova na računalu. Napravljen je pomoću C, C++ i Python programskih jezika, te je više-platformska aplikacija što znači da ga je moguće instalirati i na drugim operacijskim sustavima, ne samo na Linux. Nmap također ima mogućnost prepoznati koji operacijski sustav koristi ciljano računalo, tehnika poznata kao „fingerprinting“. Svojim izlaskom 2020. godine postao je neizbježan alat svih administratora informatičkih sustava upravo za testiranje sigurnosti računala i mreže, ali i hakerima zbog mogućnosti provaljivanja u računala i mrežne sustave.

Sada možemo upaliti sva virtualna računala koja se nalaze u VirtualBox-u klikom na „Start“ za svako pojedinačno računalo sa liste. Za svako računalo otvara se novi prozor u kojem je ono pokrenuto. Linux obično zahtjeva neku vrstu prijave u računalo. Za Kali to bi bilo korisničko ime „kali“ i lozinka „kali“ dok za Metasploitable korisničko ime je „msfadmin“ te lozinka „msfadmin“.



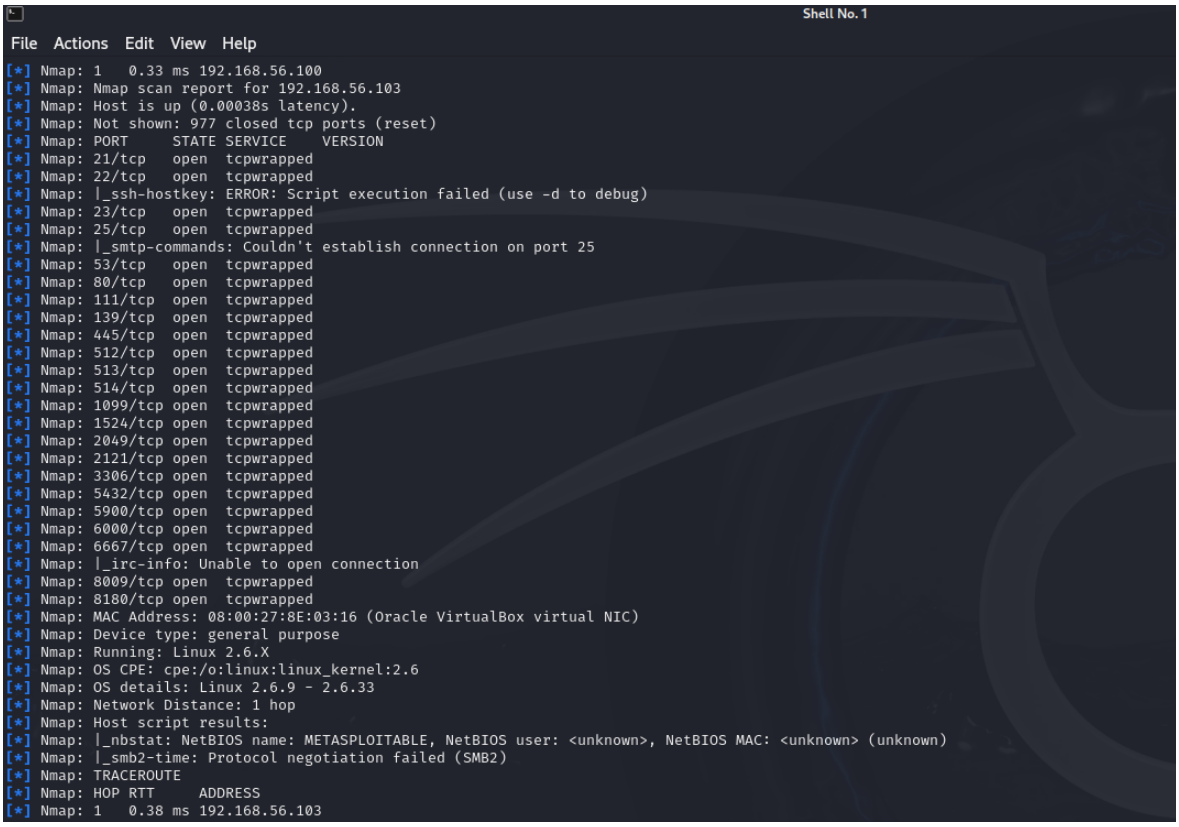
Slika 2.12 Metasploitable Linux



Slika 2.13 Kali Linux i Metasploit

Na Slika 2.12 i Slika 2.13 možemo vidjeti radna okruženja za Metasploitable i Kali Linux s otvorenim Metasploitom. Metasploitable nema grafičko korisničko sučelje kao „normalno“ računalo, već samo terminal. Kali Linux ima standardno grafičko korisničko sučelje, ali

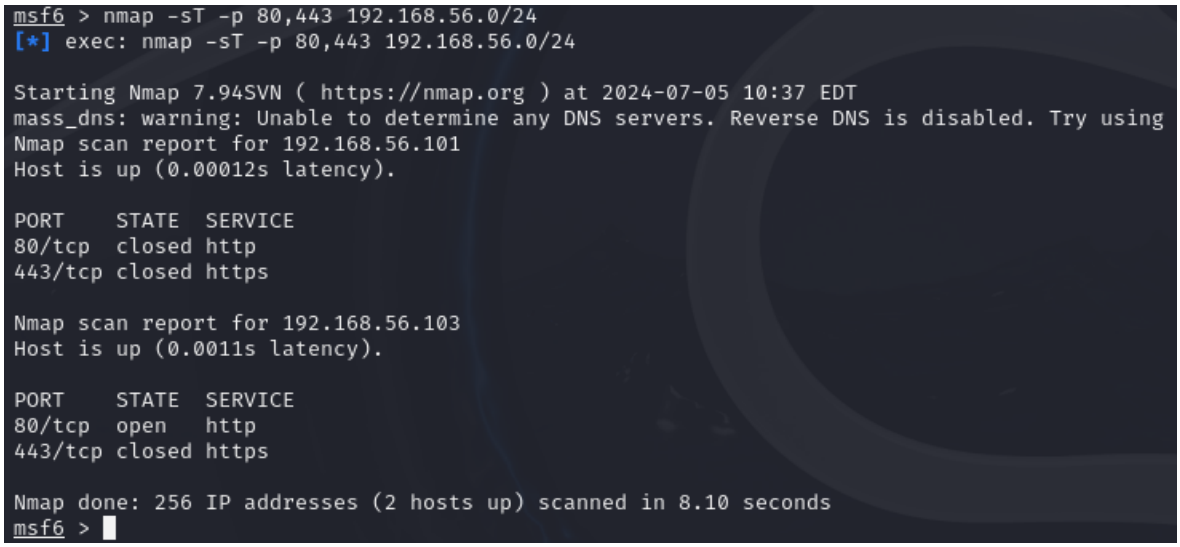
velika većina alata koji se na njemu nalaze se koriste isključivo u terminalu. Metasploit je primjer jednog takvog alata. U tražilicu Kali Linux-a je dovoljno upisati metasploit i klikom na ikonu te potvrdom administratorke lozinke s prijave, otvara se prozor vidljiv na Slika 2.13. Izvršavanjem naredbe „help“ unutar metasploit terminala dobiva se lista svih mogućnosti. Ono što nas trenutno zanima je skeniranje računala povezanih na mrežu u kojoj. Izvršavanjem naredbe „db_nmap -sS -A 192.168.1.0/24“ izvršit će se SYN skeniranje i time pokušati odrediti identificirati operacijski sustavi i verzije servisa na svim hostovima koji se nalaze u mreži 192.168.1.0/24. SYN pretraživanje je taktika koju koriste hakeri kako bi odredili status komunikacijskog porta bez da ima punu povezanost. Ovaj pristup je jedan od najstarijih i ponekad se koristi za izvođenje denial-of-service (DoS) napada. Radi tako da pomoću TCP/IP protokola šalje sinkronizacijski paket (SYN) na sve moguće portove mete pokušavajući inicijalizirati lažni „handshake“. Ako meta odgovori sa ACK (acknowledgement) ili SYN/ACK (synchronization acknowledged) paketom sa specifičnog porta, to znači da je port otvoren. Zatim napadač šalje RST(reset) paket kako bi meta mislila da je došlo do greške u komunikaciji. U tom slučaju, nije došlo do greške, port je otvoren i moguće ga je eksploatirati. Ako meta odgovori sa RST paketom, to znači da je port zatvoren i nije moguće izvršiti eksploataciju preko njega.



Slika 2.14 Izvršeno nmap skeniranje

Izvršavanjem naredbe „db_nmap -sS -A 192.168.56.0/24“ dobiva se rezultat vidljiv na Slika 2.14 poviše. Sama naredba provodi prethodno opisanu akciju skeniranja pomoću SYN pretraživanja po IP adresama dometa. Taj domet je uzet specifično zato što se sva virtualna računala u VirtualBox-u nalaze u tom dometu. Informacije koje možemo saznati su prethodno spomenute. Vidimo da je operacijski sustav Linux verzije 2.6.X, da je računalo aktivno, kolika je latencija između dva računala (u ovom slučaju to je 0.00038s), udaljenost na mreži (1 hop), MAC adresa (koja je vidljivo ona VirtualBox-a) i nekoliko otvorenih portova koji koriste „tcp“ protokol. To je dovoljno informacija za bilo kojeg hakera da pokrene pokušaje eksploatacije na pojedinim portovima određenih uređaja. Naravno u ovom slučaju smo znali u kojem dometu će se nalaziti IP adrese, jer je mreža unutar VirtualBox-a bila namještena na njega. Iako je bilo namješteno, i dalje je trebalo 73.56 sekunda da se skenira tih 254 mogućih krajnjih točaka, točnije potencijalnih meta.

Skeniranja možemo specificirati tako da na primjer tražimo specifično portove poput portova 80 i 443 koji se često koriste za web stranice.



Slika 2.15 Nmap skeniranje portova

Na Slika 2.15 vidimo rezultate nmap skeniranja portova filtrirano po portovima 80 i 443. Imamo rezultate za dva računala, od kojeg je jedno Kali Linux s adresom 101, a drugi Metasploitable sa adresom 103. Vidimo da za adresu 101 oba porta su zatvorena, što ima smisla s obzirom da je to napadačko računalo. Metasploitable s druge strane ima otvoren port 80 tcp protokolom i na njemu se koristi http servis. Taj port je otvoren možemo ga iskoristiti za eksploataciju.

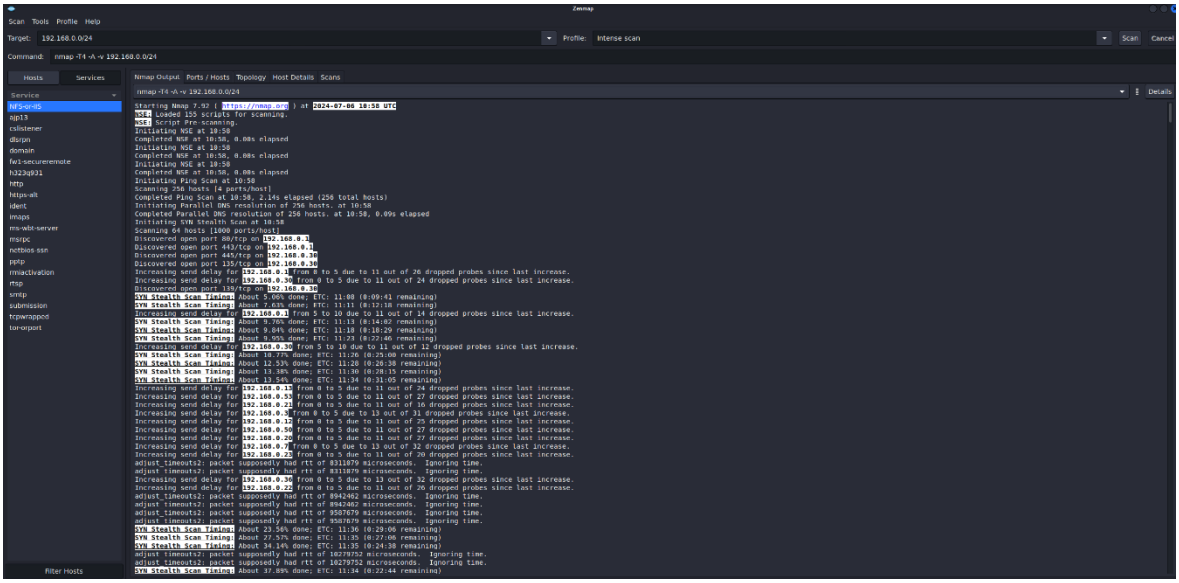
2.2.1. Zenmap i Nmap

Zenmap je službena verzija Nmap-a s grafičkim korisničkim sučeljem (GUI). Besplatan je program otvorenog koda za više platformi (Linux, Windows, Mac OS X…) čiji je cilj olakšati korištenje Nmap-a početnicima pružajući neke naprednije značajke za iskusne korisnike Nmap-a. Skeniranja koja se često koriste mogu biti spremljena kao profili za lakšu ponovnu upotrebu, rezultati skeniranja mogu biti spremljena u bazu podataka i pregledani kasnije te se mogu međusobno uspoređivati kako bi se vidjele razlike. Zenmap često dolazi upakiran zajedno s instalacijom Nmap-a. Doduše to u Kali-u nije situacija, jer u novijim verzijama Kali Linuxa dolazi samo Nmap. Za instalaciju je potrebno u terminalu unijeti naredbu „apt install zenmap -kbx“ i Kali preuzme sve datoteke potrebne za instalaciju. Za pokretanje programa, dovoljno je u tražilicu upisati ime i odabrati Zenmap ikonu.

Tablica 1 Usporedba Nmap-a i Zenmap-a

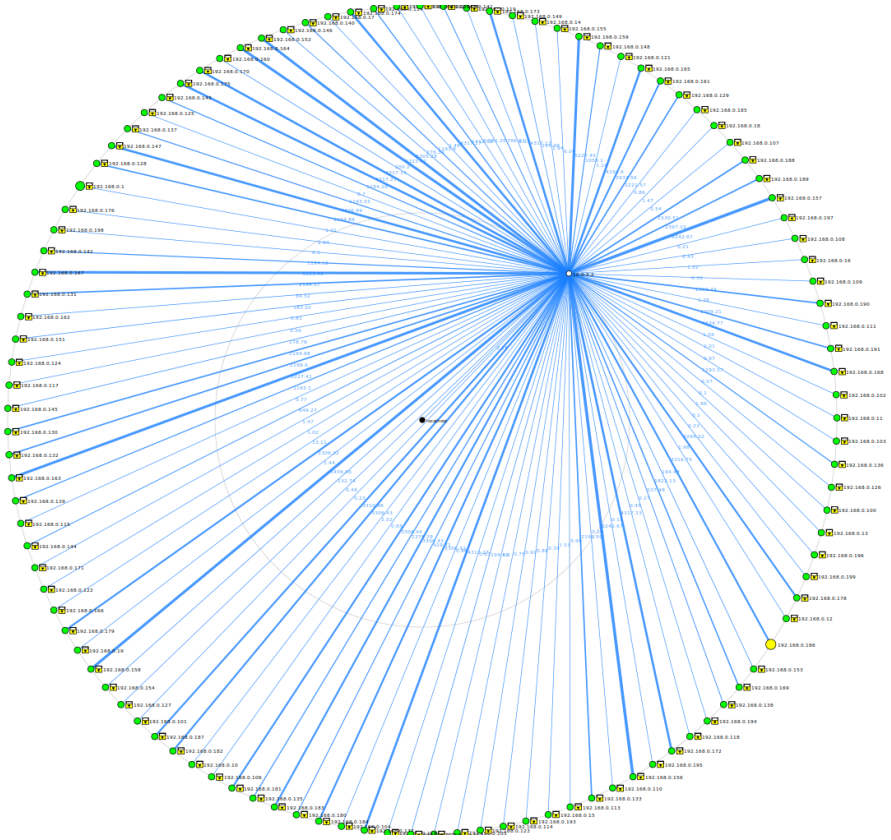
	Nmap	**Zenmap**
Korisničko sučelje	Koristi terminal kao sučelje	Verzija Nmapa sa grafičkim sučeljem, ima i terminalske komponente
Otisak	Lagan i prenosiv, dolazi instaliran na nekim distribucijama Linuxa (Kali)	Ima velik „otisak“ s obzirom na grafičko sučelje
Lakoća korištenja	Zahtjeva korištenje i komandnih linija	Lagan i jednostavan za korištenje, pogotovo početnicima
Namijenjen za	Naprednije korisnike	Za početnike
Izvještaji	Rezultati skeniranja su tekstualni	Rezultati skeniranja mogu biti spremljeni kao topologijske slike

Tablica 1 prikazuje usporedbu Nmap i Zenmap alata za skeniranje. Grafičko korisničko sučelje zenmap-a u kombinaciji s terminalnim komponentama koje ima, omogućava lakše korištenje i istu funkcionalnost kao kod Nmap-a, što definitivno je veliki plus. Nmap zahtjeva poznavanje naredbenih linija Linux-a i snalaženje u terminalu. Zenmap nudi opcije korištenja grafičkog sučelja i terminala. Korištenjem grafičkog sučelja također se generiraju naredbe za terminal, time se zapravo može učiti korištenje samog Nmapa i terminalskih naredbi što čini Zenmap idealnim za početnike. Također Zenmap nudi spremanje rezultata skeniranja u bazu podataka te paralelnu usporedbu tih skeniranja. Uz to također nudi i izradu topologije mreže koja se skenira (Slika 2.17). No ono što čini Nmap i dalje popularnim i korištenim je to što je lagan i prenosiv. To znači da ima mali digitalni otisak i koristi samo ono najpotrebnije za rad, što znači da ga je teže detektirati. Naravno u penetracijskim testovima želimo ostati ne primijećeni stoga je Nmap bolji izbor, uz to što skeniranja izvršava brže od Zenmapa upravo zbog manjka opcija, odnosno dodatnih mogućnosti ne vezanih za samo skeniranje.



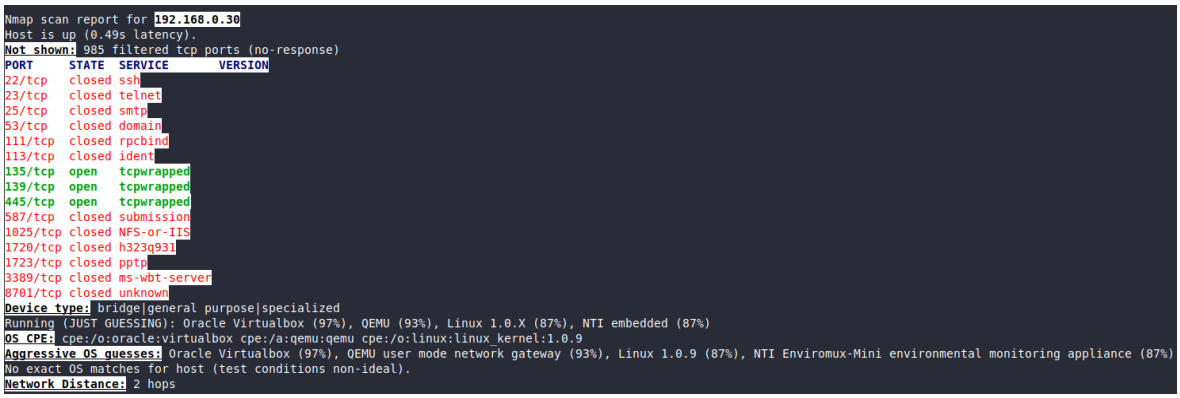
Slika 2.16 Skeniranje kućne mreže pomoću Zenmap-a

Na Slika 2.16 vidimo proces skeniranja prave kućne LAN mreže u dometu IP adresa 192.168.0.0/24. Zenmap prolazi kroz svaku moguću krajnju točku u tom dometu i radi takozvani „intense scan“. Ta vrsta skeniranja pokušava otkriti sve što može o krajnjim točkama poput otvorenih portova, servisa i operacijskih sustava. Sve informacije koje mogu bitni bitne označi bojama. Također daje i sliku topologije skenirane mreže koju možemo vidjeti na Slika 2.17 ispod.



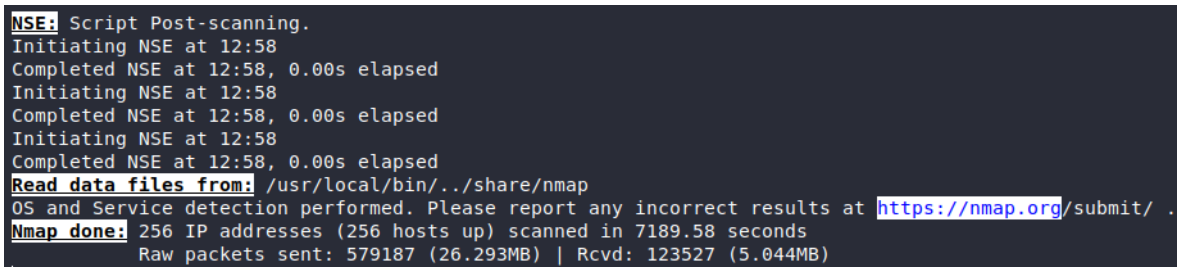
Slika 2.17 Topologija kućne mreže

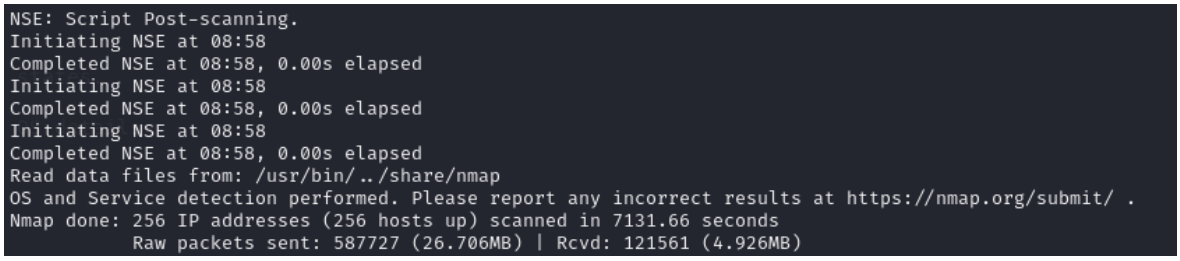
Rezultate skeniranja možemo detaljno filtrirati po domaćinima (engl. *Hosts*) i servisima. Klikom na nekog od domaćina dobivamo sve informacije koje su pronađene o njemu. Rezultati skeniranja jednog takvog računala na adresi 192.168.0.30 nalaze se na Slika 2.18 ispod gdje je naznačeno zelenom bojom koji portovi se mogu eksploatirati, a crvenom oni koji ne mogu.



Slika 2.18 Zenmap i Nmap skeniranje jednog hosta

Kada bi pokrenuli isto skeniranje, s istom naredbom (nmap -T4 -A -v 192.168.0.0/24) u običnom nmap-u dobili bi iste rezultate kao i u zenmap-u. Razlika je između dvoje je što u nmap-u dobivamo čisti tekst bez ikakvih naglašavanja bitnih stavki, nije moguća filtracija niti uspoređivanje rezultata s nekim drugim. Ali ono u čemu je nmap bolji je u vremenu izvođenja iako je razlika gotovo zanemariva. Skeniranje zenmap-om trajalo je 7189.58 sekundi, dok je skeniranje nmap-om trajalo 7131.66 sekundi.





Slika 2.19 Rezultati skeniranja Zenmap-om i Nmap-om

2.3. Eksploatacija

Kada je gotova faza izviđanja i skeniranja i imamo potrebne informacije o određenim portovima, njihovim statusima i servisima koje koriste, možemo početi tražiti odgovarajuće eksploatacijske alate, odnosno skripte, koji bi bili primjereni za napad preko određenog porta i servisa. U metasploitu unesemo naredbu „nmap -F -sV 192.168.56.102“. Ta naredba koristi nmap brzo skeniranje (-F) uključujući ispitivanje otvorenih portova kako bi se odredili servisi i verzije servisa portova na adresi 192.168.56.102 ciljnog računala kojeg smo odabrali. Izvršavanjem te naredbe dobivamo listu otvorenih portova koji su potencijalno podložni nekoj vrsti eksploatacije, kao i informacije o servisima koji se koriste na tim portovima i njihove verzije, što možemo vidjeti na Slika 2.20 ispod.

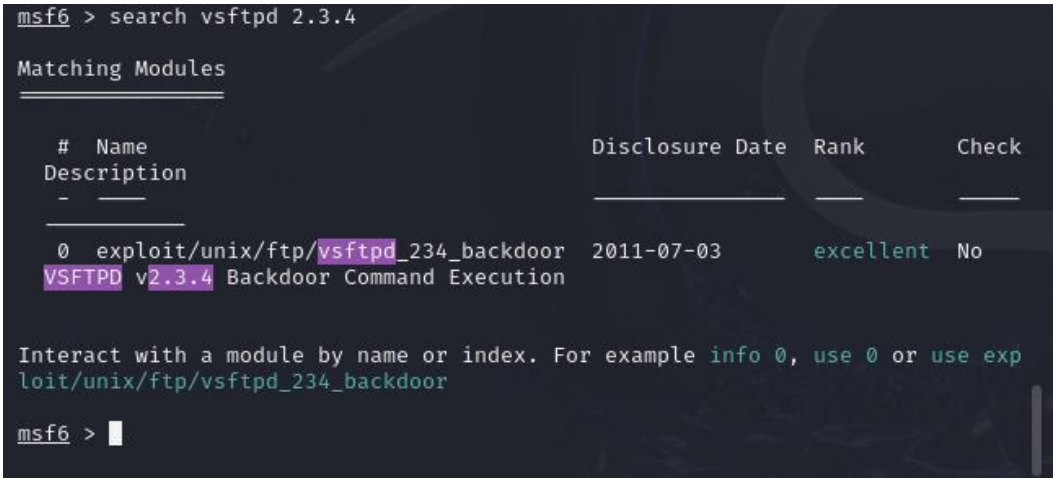
```
msf6 > nmap  -F -sV 192.168.56.102
[*] exec: nmap  -F -sV 192.168.56.102

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-14 06:55 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabl
ed. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.102
Host is up (0.0015s latency).
Not shown: 82 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
513/tcp  open  login
514/tcp  open  shell       Netkit rshd
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
```

Slika 2.20 Rezultat nmap skeniranja mete

2.3.1. VSFTPD eksploatacija

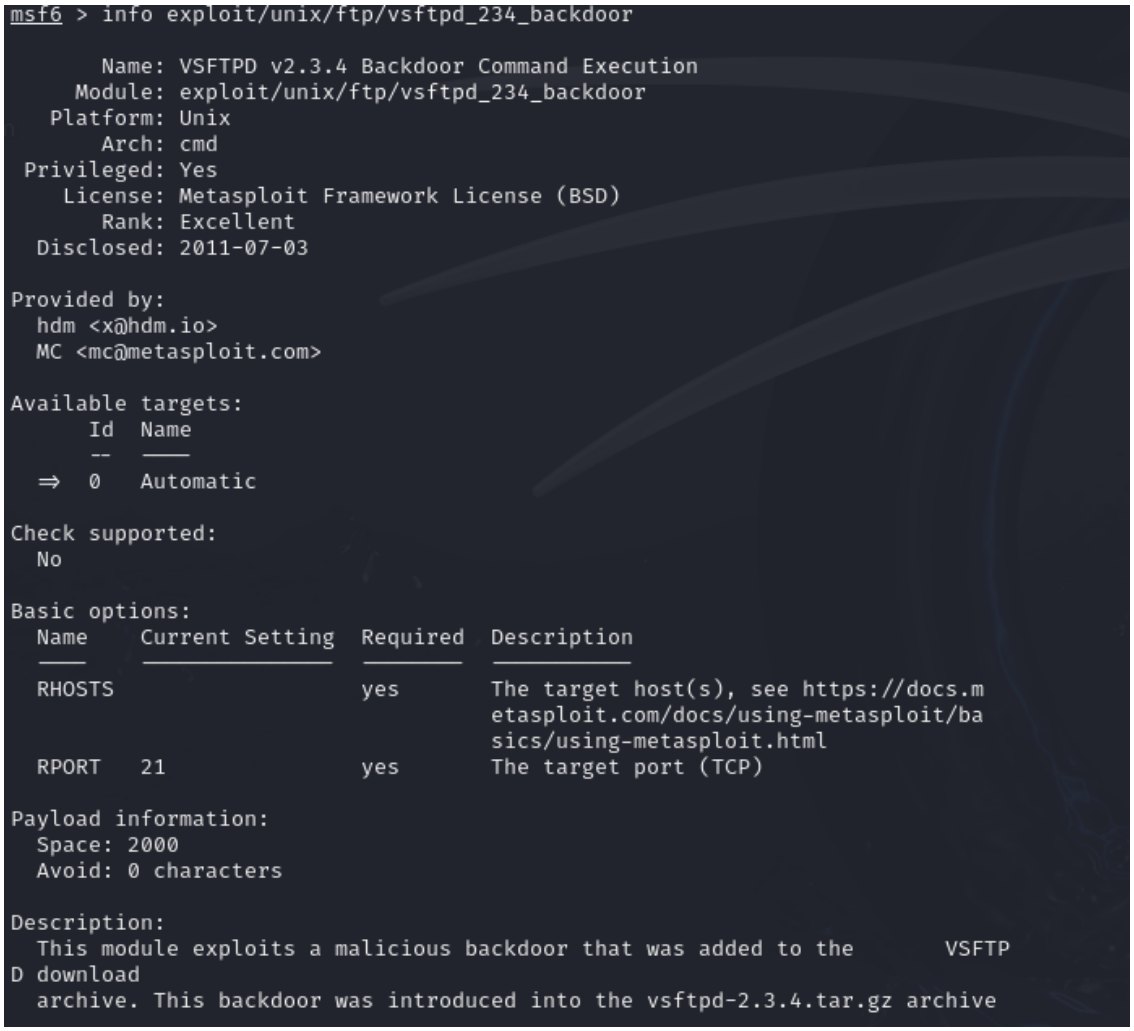
S liste otvorenih portova odabiremo jedan koji ćemo pokušati eksploatirati. Možemo uzeti port 21 koji koristi ftp servis čija je verzija vsftpd 2.3.4. Kada bi u Google tražilicu pretražili tu verziju ftp servisa, odmah bi otkrili kako postoji otkrivena ranjivost te specifične verzije. S obzirom da znamo da port ima ranjivost koja nije pokrivena, možemo je eksploatirati.

Kako bi pronašli adekvatan alat ili skriptu za eksploataciju odabranog porta, u terminalu metasploita unosimo i izvršavamo naredbu „search“ i ime verzije servisa koja bi u našem slučaju glasila „*search vsftpd 2.3.4*“. Izvršavanjem te naredbe pretrage, dobivamo listu mogućih eksploatacija koje možemo iskoristiti za napad preko tog porta i servisa.



Slika 2.21 Rezultat search naredbe

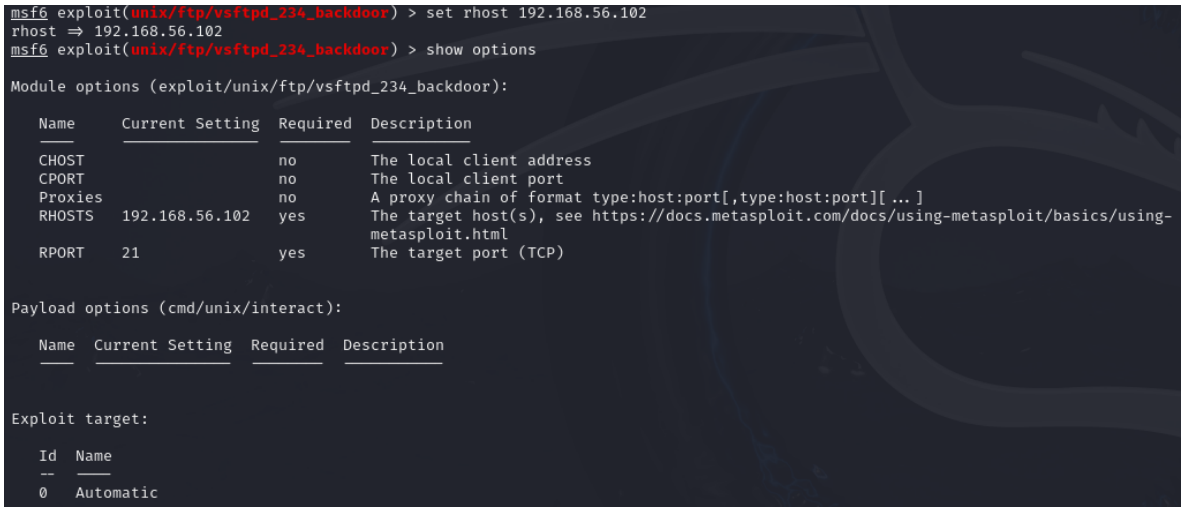
Pretraživanje prikazano na Slika 2.21 je pronašlo jednu moguću eksploataciju koja bi odgovarala u ovom slučaju. Ta skripta pod imenom „vsftpd_234_backdoor“ omogućava nam pristup nad ciljanim računalom. Točnije, ideja napada na vsftpd 2.3.4 je izvršavanje maliciozne funkcije slanjem specifičnih bajtova na port 21, koja, na uspješnom izvođenju, otvara „stražnja vrata“ (backdoor) u ciljanom računalu i daje pristup računalu napadaču. Da bi znali što je točno potrebno za korištenje same eksploatacije, možemo tražiti informacije o njoj. Unosom naredbe „info“ s putanjom do eksploatacijske skripte dobivamo osnovne informacije o skripti poput njenog opisa, postavki i što je potrebno za njeno korištenje.



Slika 2.22 Informacije o skirpti

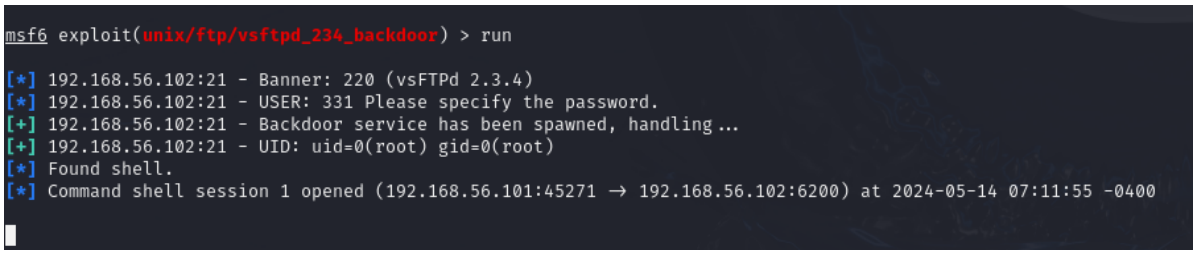
Slika 2.22 prikazuje informacije o skripti koju ćemo koristiti. Sami izvorni kod skripte je javno dostupan i može se pronaći na službenim stranicama Metasploit Framework-a: https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor/.

U ispisanim informacijama možemo vidjeti naziv skripte, odnosno modula, platformu, arhitekturu, licence, opis i što je potrebno za korištenje. Pod „Basic options“ možemo vidjeti RHOST i RPORT. RHOST je ono što želimo napasti, točnije to bi bila IP adresa računala koje napadamo i trenutno na njemu nije zadana nikakva adresa. RPORT predstavlja port po kojem će ići napad. On je već namješten na vrijednost 21 s obzirom da je već od prije poznato kako napadamo preko porta 21. Ono što treba napraviti je postaviti IP adresu na ciljno računalo. To ćemo napraviti naredbom „set rhost 192.168.56.102“. Nakon izvršavanja te naredbe, možemo pregledati ponovno sve postavljene opcije s naredbom „show options“ kako bi provjerili da su sve potrebne postavke namještene.



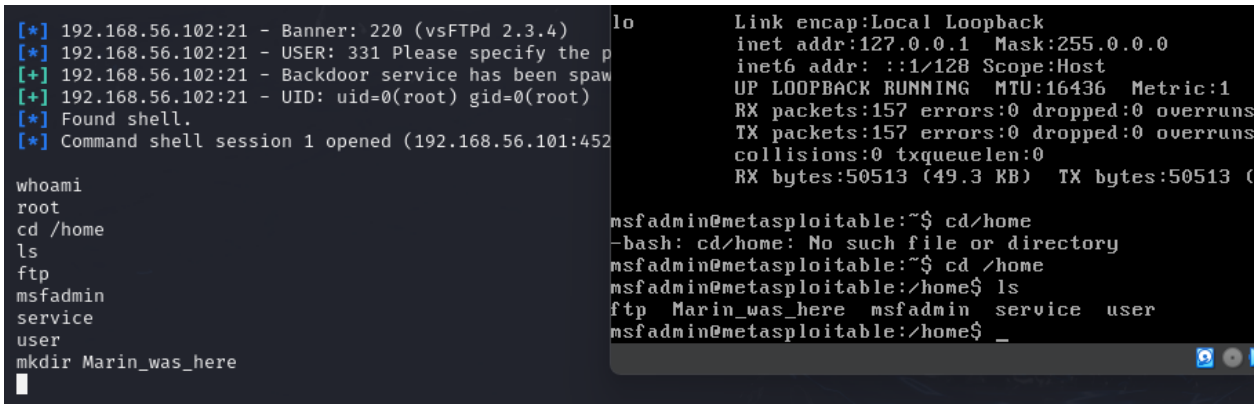
Slika 2.23 Naredba "show options"

Kada su sve postavke uspješno namještene, za pokretanje same eksploatacijske skripte potrebno je samo unijeti naredbu „run“ i to će pokrenuti izvršavanje skripte.



Slika 2.24 Izvršavanje skripte

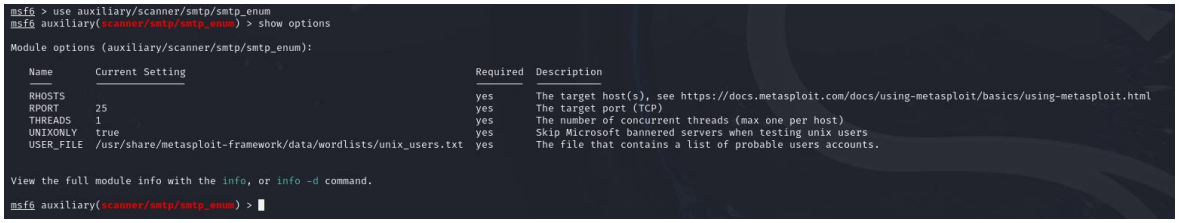
Izvršavanje skripte je gotovo kada se pojavi trepereći bijeli pravokutnik kraj kojeg nema nikakvog teksta. To je također i indikator kako je i sama eksploatacija bila uspješna i da sada imamo pristup ciljanom računalu. Kako bi provjerili da smo stvarno dobili pristup ciljnom računalu možemo izvršiti nekoliko naredbi. Pomoću naredbe u terminalu metasploita „whoami“ možemo vidjeti da ne piše „kali“ nego „root“. Zatim ako se pozicioniramo u „home“ direktorij naredbom „cd /home“. Unutar tog direktorija možemo pokušati napraviti novu datoteku ili direktorij. Izvršit ćemo naredbu „mkdir Marin_was_here“ i prikazati što se sve nalazi u „home“ direktoriju s naredbom „ls“. Ako bi se prebacili na računalo koja je meta i pozicionirali se u njegov „home“ direktorij s naredbom „cd /home“ i zatim napravili ispis sa „ls“, možemo vidjeti direktorij koji smo napravili s „mkdir“ naredbom u napadačkom računalu.



Slika 2.25 Usporedba home direktorija

2.3.2. SMTP eksploatacija

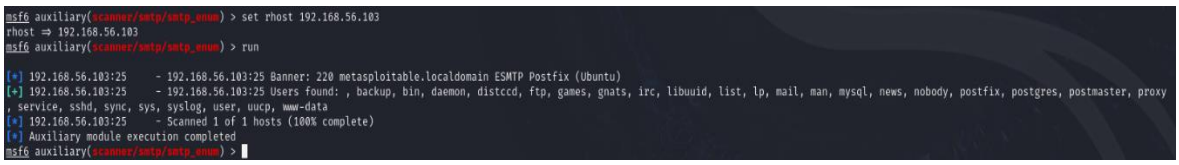
Također možemo pokušati provesti eksploataciju putem SMTP-a (Simple Mail Transfer Protocol). Taj protokol je standard za prijenos električne pošte koji ne zahtjeva nikakvu autentifikaciju što može biti problem ako postoji rupa u SMTP-u. S eksploatacijom SMTP-a nećemo dobiti pristup nekom računalu, ali ono što nam to omogućuje je nekakvu bazu za daljnje napade. Pomoću SMTP servera se provjerava tko šalje e-mailove, šalje odlaznu poštu sa servera i u slučaju da odlazna pošta ne može biti poslana, šalje pošiljatelju obavijest o nemogućnosti slanja. Pošto želimo eksploatirati SMTP koji se nalazi na portu 25 u što se možemo uvjeriti izvršavanjem naredbe skeniranja s „nmap -sV 192.168.56.103“. Ono šta želimo napraviti je saznati imena svih korisničkih računa koji se nalaze na serveru. Za to će nam trebati jedna poseban vrsta skeniranja, takozvana smtp enumeracija. U metasploit terminalu treba izvršiti naredbu „use/auxiliary/smtp/smtp_enum“ kako bi odabrali potrebnu skriptu. Izvršavanjem „show options“ dobit ćemo potrebne informacije za korištenje skripte.



Slika 2.26 Opcije smtp_enum

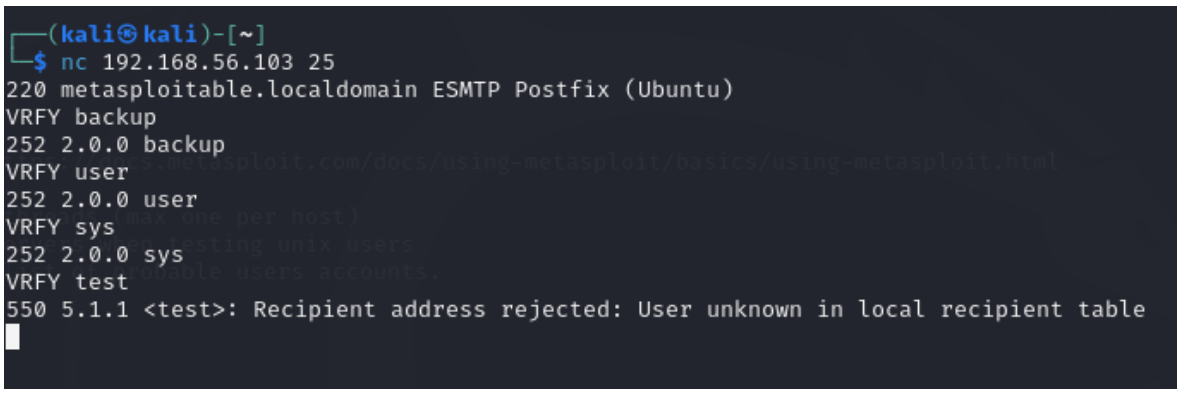
Pod opcijama i informacijama o smtp_enum možemo vidjeti neke već poznate stavke poput RHOST i RPORT. Ono što je novo je USER_FILE. To je tekstualna datoteka koja se kreira i u koju se spremaju potencijalni pronađeni računi sa servera. Postavimo RHOST na IP adresu cilja s naredbom „ set rhost 192.168.56.103“ i zatim izvršimo naredbu „run“ kako bi pokrenuli skriptu, odnosno skeniranje. Samo skeniranje može potrajati ovisno o više faktora,

uključujući računala s kojeg se pokreće. Završenim skeniranjem trebali bi dobiti listu potencijalnih korisničkih imena na tom računalu, odnosno serveru.



Slika 2.27 Rezultat smtp _enum skeniranja

Kao što je bilo za očekivati, dobili smo listu potencijalnih korisničkih imena koje možemo vidjeti na Slika 2.27, ali to nisu sve prava korisnička imena. Ono što nam sada preostaje je provjeriti koji su stvarni a koji ne. To ćemo napraviti tako da otvorimo novi Linux terminal i izvršimo naredbu „nc 192.168.56.103 25“. Naredba „nc“, to jest Netcat, je mrežni alat koji služi za čitanje i pisanje na mrežnim vezama, koristeći TCP ili UDP protokole. Spajamo se Netcatom preko porta 25 na računalo odnosno server i vršimo provjeru. Provjeru ćemo provesti tako da u netcat terminalu napišemo naredbu „VRFY *ime korisnika*“. Ako korisnik ne postoji dobit ćemo poruku da je korisnik nepoznat.

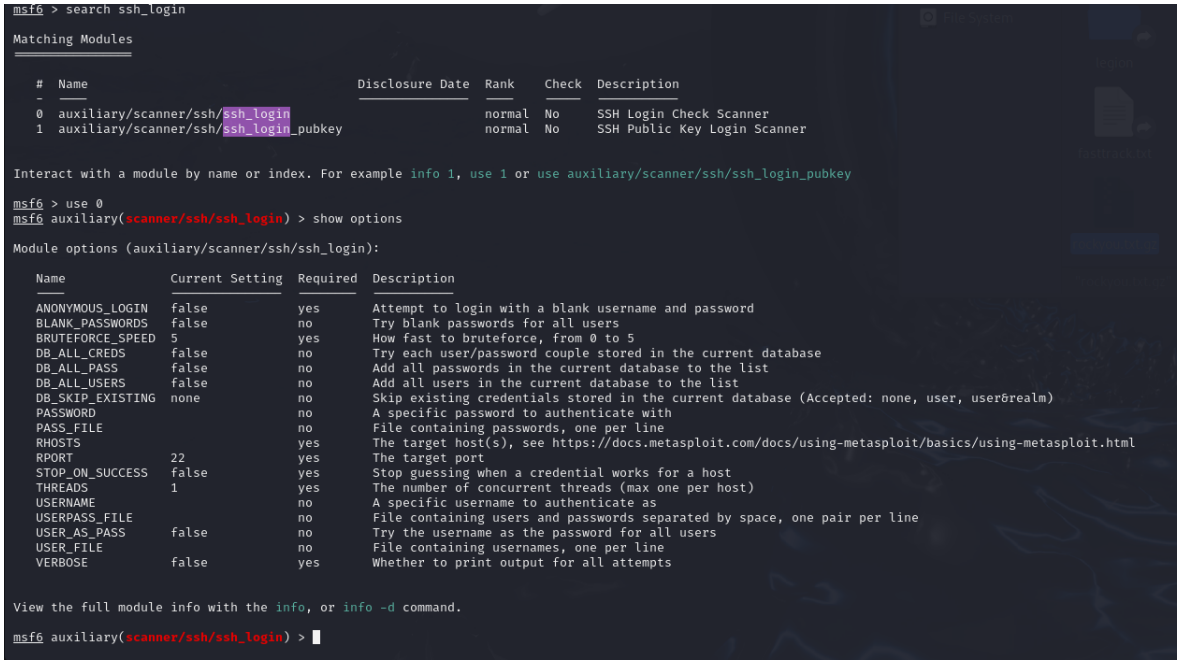


Slika 2.28 Netcat provjera korisnika

Kada korisnički računi budu provjereni i poznato je koji su pravi a koji ne, to možemo iskoristiti za pokušaj dobivanja pristupa na to računalo, odnosno server.

Ono što zapravo možemo napraviti je iskoristiti ta korisnička imena kako bi dobili pristup računalo i to koristeći ssh (Secure Shell) protokol. Ssh protokol je metoda za sigurno slanje naredbi računalu preko nezaštićene mreže. Protokol koristi kriptografiju za provjeru autentičnosti i šifriranje veze između udaljenih uređaja. Često se koristi za udaljeni pristup serverima, upravljanje infrastrukturom i prijenos podataka. S obzirom na osjetljive uvjete u kojima ssh radi, vidimo da probijanje njega i dobivanje pristupa računalu putem ssh protokola može biti pogubno za bilo koju žrtvu.

Eksploatacija ssh protokola se vrši izvođenjem „brute force“ ili „dictionary“ napada na pokušaje prijave. Za to Kali Linux ima pregršt alata poput Hydra. U ovom slučaju ćemo pokušati provesti „dictionary“ napad sa listom korisnika koje smo dohvatili preko eksploatacija SMTP protokola. U terminalu metasploita treba napraviti pretragu na ssh login naredbom „search ssh_login“. Odabiremo opciju 0 s naredbom „use 0“ i zatim „show options“ kao u dosadašnjim primjerima.



Slika 2.29 Opcije ssh_login

Pod „show options“ imamo puno više opcija nego u dosadašnjim primjerima. Ono što nas zanima je RHOST, PASS_FILE, STOP_ON_SUCCESS, USER_FILE i VERBOSE. Naredbom „set rhost“ namještamo IP adresu mete na 192.168.56.103. PASS_FILE će biti takozvani rječnik, odnosno kolekcija najkorištenijih lozinki. Za to ćemo koristiti tekstualni dokument s najpopularnijim lozinkama za unix sisteme koji se već nalazi spremljen u dokumentima na Kali Linux operacijskom sustavu. Naredba za postavljanje tog rječnika je „set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt“. U istom direktoriju se nalazi i lista najčešćih korisničkih imena na unix sistemima, uključujući one koji su izvučeni preko SMTP eksploatacije. Tu listu imena, odnosno rječnik postavljamo s naredbom „set USER_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt“. VERBOSE nije toliko ključan, ali s njim možemo vidjeti trenutne i prethodne pokušaje. Ako je postavljen na true vrijednost s naredbom „VERBOSE true“, onda se ispisuje svaki prošli i trenutni pokušaj kombinacija lozinki i korisničkih imena. Zadnje što nam preostaje je STOP_ON_SUCCESS opcija. Ako

tu opciju postavimo na true vrijednost s naredbom „set STOP_ON_SUCCESS true“, kada dobijemo pozitivnu kombinaciju lozinke i korisničkog imena, prestaje se sa izvršavanjem jer nam više nije ni potrebno.

Slika 2.30 Postavke nakon postavljanja ssh_login

Kada su postavke namještene i postavljene možemo krenuti sa probijanjem. Sve što treba je izvršiti naredbu „exploit“ i program krene s izvršavanjem. Ovo će biti dugotrajan proces jer program pokušava svaku stavku iz jednog rječnika kombinirati sa svakom stavkom iz drugog rječnika. Uz to sami proces ovisi i jačini računala s kojeg se pokreće napad, te koliko brzo može proći kroz pojedinu riječ i kombinaciju riječi.

Slika 2.31 Proces probijanja lozinke

Izvršavanje skripte staje kada se pronađe odgovarajući par korisničkog imena i lozinke što će biti naznačeno zelenom bojom, ili dok skripta ne prođe kroz oba rječnika, bez pronađene kombinacije što znači da je lozinka vjerojatno puno snažnija.

2.3.3. Usporedba eksploatacijskih alata

Metasploit, moćan i široko korišten okvir za penetracijsko testiranje, nudi razne metode eksploatacije koje omogućuju stručnjacima za sigurnost testiranje sigurnosti sustava i mreža. Svaka metoda služi različitim svrhama i ima svoje prednosti i nedostatke. Najčešće korištene eksploatacijske module dijelimo na: daljinske eksploatacije, lokalne eksploatacije i eksploatacije na strani klijenta. Svaki od eksploatacijskih alata unutar svoje kategorije koristi se ovisno o „stanju“ mete i samim uvjetima pristupa meti. Stoga se ne može reći da je neki od alata bolji od drugog, jer svaki se koristi za svoju specifičnu svrhu. Naravno postoje prednosti i nedostaci za svaki. Neki su lakši za upotrebu i zahtijevaju minimalan trud za korištenje, ali zahtijevaju da meta bude u nekom lako iskoristivom stanju (na primjer neažurirani mrežni servisi). Drugi mogu biti zahtjevniji, dugotrajniji pa čak i zahtijevati interakciju s metom poput socijalnog inženjeringa (engl. *Social engineering*), ali „nagrada“ može biti puno veća.

Tablica 2 Usporedba eksploatacijskih alata

Metoda Eksploatacije	**Opis**	**Primjeri**	**Prednosti**	**Nedostaci**
Daljinske Eksploatacije	Eksploatacije koje ciljaju ranjivosti u mrežnim servisima dostupnim preko mreže.	SMB ranjivosti (npr. EternalBlue), ranjivosti web poslužitelja (npr. Apache Struts)	-Mogu se pokrenuti s bilo kojeg mjesta s mrežnim pristupom. -Mogu utjecati na više sustava.	-Zahtijeva izloženost ciljane usluge mreži. -Lako detektabilno od strane IDS/IPS. -Ovisi o mrežnim uvjetima i pravilima vatrozida.
Lokalne Eksploatacije	Zahtijeva početni pristup sustavu za eskalaciju privilegija ili daljnji pristup	Ranjivosti za eskalaciju privilegija (npr. kernel, SUID binarni fajlovi)	-Zaobilazi mrežno-bazirane sigurnosne mjere. -Korisno za eskalaciju privilegija.	-Zahtijeva početni pristup sustavu. -Ograničeno trenutnom razinom pristupa.
Eksploatacije Na Strani Klijenta	Ciljaju ranjivosti u klijentskim aplikacijama.	Ranjivosti preglednika (npr. Flash, Java), eksploatacije dokumenata (npr. zlonamjerni PDF-ovi)	-Zaobilazi obranu na strani poslužitelja. -Cilja izravno korisnikovo računalo.	-Zahtijeva korisničku interakciju. - Ovisi o socijalnom inženjeringu.

Na Tablica 2 vidimo usporedbu triju kategorija eksploatacijskih alata unutar Metasploit platforme. Daljinske eksploatacije ciljaju ranjivosti u mrežnim servisima i aplikacijama koje su dostupne preko mreže. Ove eksploatacije omogućuju napadačima da dobiju neovlašteni pristup sustavima bez potrebe za fizičkim pristupom. Prednosti ovih eksploatacija su što se mogu pokrenuti s bilo kojeg mjesta koje ima mrežni pristup cilju, što više-manje teoretski

pokriva cijeli planet koristeći virtualne privatne mreže ili *proxychain* (Maravić, 2021) i što potencijalno omogućava utjecaj eksploatacije na širu mrežu, odnosno može utjecati na više sustava ovisno o tome je li eksploatirana usluga široko korištena. Nedostaci ovih alatu su to što zahtijevaju da usluga koja se želi eksploatirati bude izložena mreži, jer inače nije moguće koristiti alate iz ove kategorije. Također ovi tipovi eksploatacijskih alata mogu se lako detektirati od strane IDS/IPS sustava, te ovise čak i o mrežnim uvjetima i pravilima vatrozida.

Lokalne eksploatacije zahtijevaju da napadač ima neku razinu pristupa ciljnom sustavu (npr. korisnički račun). Ove eksploatacije se koriste za eskalaciju privilegija ili dobivanje daljnjeg pristupa nakon što je početni pristup dobiven. Prednost alata iz ove kategorije je što mogu zaobići mrežno-bazirane sigurnosne mjere jer se izvršavaju izravno na ciljnom sustavu. Ovi alati zapravo omogućavaju prodiranje dublje u sustav, točnije korisni su za eskalaciju privilegija nakon upada u kompromitirani sustav. Problem ovih alata je taj što za iskoristiti ih već mora postojati pristup sustavu, ali čak i uspješnim eksploatiranjem, ograničeni su trenutnom razinom pristupa napadnutog računala, sve dok se ne provedu post-eksploatacijski moduli.

Eksploatacije na strani klijenta ciljaju ranjivosti u klijentskim aplikacijama, poput web preglednika, medijskih playera ili email klijenata. Napadač obično mora prevariti korisnika da izvrši eksploataciju, poput posjete zlonamjernoj web stranici ili otvaranja zlonamjerne datoteke. Ovo je čak i najčešći oblik koji vidimo u stvarnom životu, poput lažnih mailove o dobitku na lutriji, poruke o nedostavljenom paketu na mobitelima, lažni računalni pomoćni centri i slično. Ovi alati „žive i umiru“ na tome hoće li napadač uspjeti prevariti korisnika, odnosno na ljudskom faktoru i naivnosti korisnika, što je glavna i zapravo jedina mana ovih alata. Stoga napadači, odnosno testeri moraju biti jako kreativni, pa čak i do te mjere da izrade dovoljno uvjerljive web stranice kako bi prevarili korisnike. Ali prednost ovih alata, ako je socijalni inženjering uspješan, je zaobilaženje velike većine obrambenih mjera na strani poslužitelja i dobivanje pristupa korisnikovom računalom iskorištavajući neispravan softver ili ubacujući vlastiti lažni poput modificiranih dokumenata Office paketa.

2.4. Post-eksploatacija

Faza post-eksploatacije se odvija nakon što je uspješno došlo do proboja u sustav, odnosno neko računalo. Iako je primarni cilj bilo kakvog penetracijskog testiranja pronaći ranjivosti i eksploatirati ih kako bi se dobio pristup računalu, faza post-eksploatacije se fokusira na što se sve može napraviti nakon što se dobije pristup računalu ili sustavu. Najčešće to uključuje održavanje pristupa računalu ili sistemu, prikupljanje osjetljivih podataka i potencijalno prelaženje na ostala računala na mreži.

Najvažniji od ovih bi bio održavanje pristupa, jer samo održavanje pristupa nam omogućuje da možemo kopati dublje po računalu, a i dalje u mrežu. Primarni cilj održavanja pristupa je osiguranje da napadač može ostati unutar kompromitiranog sustava bez da bude otkriven. Održanim pristupom napadač može dalje „istraživati" sustav, identificirati dodatne ranjivosti koje može eksploatirati i prikupljati vrijedne informacije poput imena korisnika.

Iskoristit ćemo situaciju prijašnje eksploatacije preko FTP protokola preko koje smo dobili pristup računalu. Ono što želimo napraviti kada imamo uspostavljenu sesiju s ciljanim računalom je staviti tu sesiju da radi u pozadini tako da napravimo CTRL + Z na tipkovnici i prihvatimo sa „y".

```
[*] Command shell session 1 opened (192.168.56.101:35963 → 192.168.56.103:6200) at 2024-07-12 04:47:45 -0400

^Z
Background session 1? [y/N]  y
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions

Active sessions
===============

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               192.168.56.101:35963 → 192.168.56.103:6200 (192.168.56.103)
```

Slika 2.32 Sesija u pozadini

Kada imamo sesiju upaljenu u pozadini, možemo koristiti post-eksploatacijske alate, jer ti alati zahtijevaju aktivnu sesiju za korištenje. Jedan od prvih post-eksploatacijskih alata koje možemo testirati je „*hashdump*". Ideja *hashdump*-a je sakupljanje svih mogućih lozinki koje se mogu pronaći na ciljanom računalo. Te lozinke su najčešće lozinke prijave u računalo ili sustav i enkriptirane su. S obzirom da imamo sesiju sa Linux sistemom, želimo pronaći *hashdump* skriptu za Linux sisteme s naredbom u metasploit terminalu „search hashdump". Ono što nam je potrebno se nalazi na „post/linux/gather/hashdump" i odabire se sa „use 5", gdje je broj 5 broj u listi koju smo dobili pretraživanjem ili „use

post/linux/gather/hashdump“. Možemo prikazati opcije modula, odnosno skripte sa „show options“.

```
msf6 post(linux/gather/enum_system) > use 5
msf6 post(linux/gather/hashdump) > show options

Module options (post/linux/gather/hashdump):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on

View the full module info with the info, or info -d command.
```

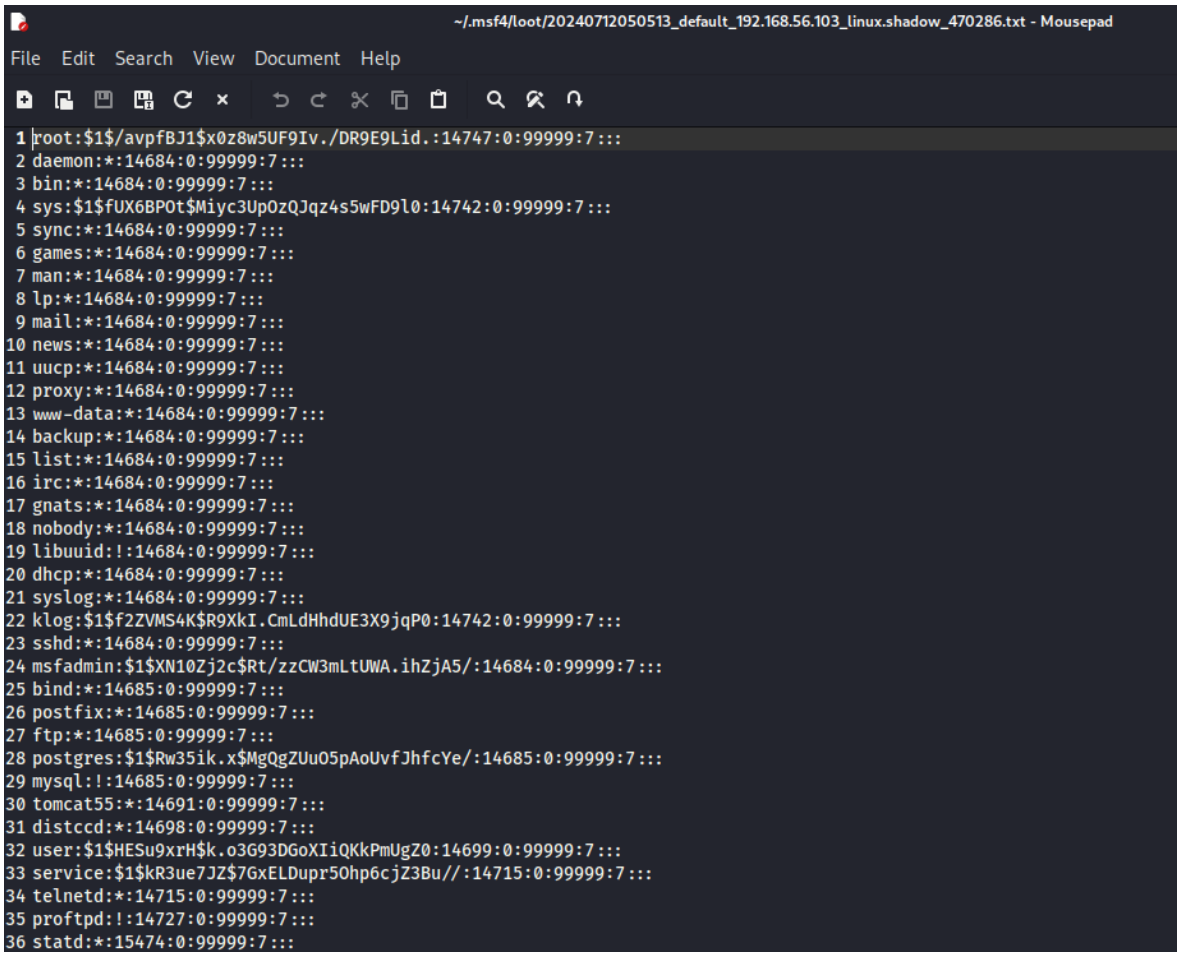
Slika 2.33 Hashdump opcije

Pod opcijama imamo samo SESSION, odnosno ID sesije na kojoj želimo provesti *hashdump*. Na Slika 2.32 vidimo da je ID sesije 1, stoga SESSION postavljamo na vrijednost 1 s naredbom „set SESSION 1“. Za pokretanje *hashdump* modula izvršavamo naredbu „exploit“.

```
msf6 post(linux/gather/hashdump) > exploit

[!] SESSION may not be compatible with this module:
[!]  * incompatible session platform: unix
[+] root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
[+] sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
[+] klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
[+] msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
[+] postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
[+] user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/home/user:/bin/bash
[+] service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/bash
[+] Unshadowed Password File: /home/kali/.msf4/loot/20240712050514_default_192.168.56.103_linux.hashes_316177.txt
[*] Post module execution completed
msf6 post(linux/gather/hashdump) >
```

Slika 2.34 Izvršavanje hashdump-a

Na Slika 2.34 poviše vidimo sam proces izvršavanja *hashdump* modula. Možemo vidjeti kako je pronađeno nekoliko lozinki za prijavu korisnika, odnosno nekoliko kriptiranih lozinki. Možemo i otići u navedeni direktorij gdje su spremljeni svi pribavljene hash vrijednosti, a ne samo korisnički.



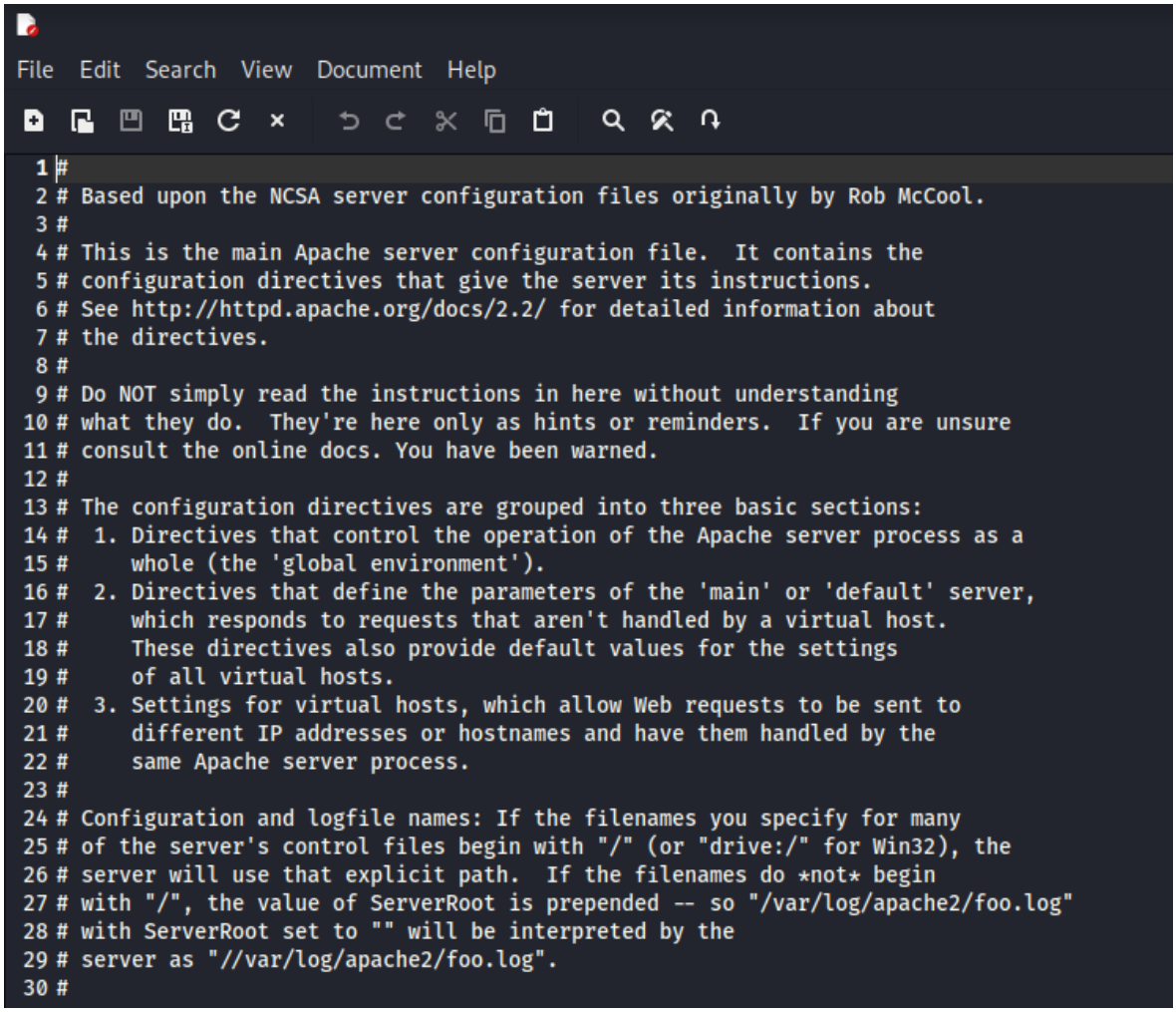
Slika 2.35 Prikupljene hash vrijednosti

Ono što možemo napraviti s ovim hash vrijednostima je pokušati s nekim drugim modulom dekriptirati prikupljene hash vrijednosti, poput *Hydra* koja je namijenjena za *brute force* napade i dekriptiranje ili *PSExec Pass the Hash* koji omogućuje prijavu na sustav bez dekriptiranja lozinke, samo korištenjem pribavljene hash vrijednosti.

Još jedan zanimljivi post eksploatacijski modul u Metasploit-u bi bio „enum_configs“ koji prikuplja sve važne konfiguracijske datoteke i sprema ih na napadačko računalo. Tom modulu pristupamo naredbom „use post/linux/gather/enum_configs“. Situacija je ista kao i kod hashdump-a. Pod „show options“ imamo samo opciju SESSION koji postavljamo na vrijednost 1 s naredbom „set SESSION 1“, i naredbom „exploit“ pokrećemo modul.

Slika 2.36 Izvršavanje enum_configs

Završetkom izvršavanja modula u home direktoriju Kali Linux-a se spremaju izvučeni konfiguracijske datoteke raznih servisa, poput „apache“ ili „mySQL“ za baze podataka.



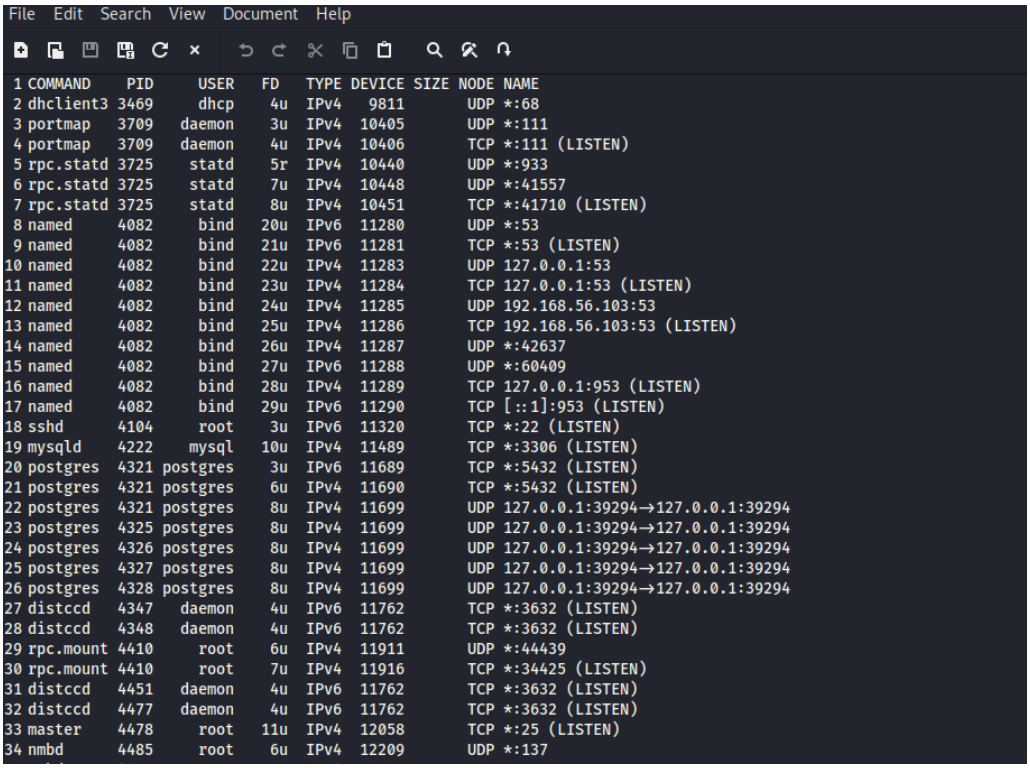
Slika 2.37 Primjer Apache konfiguracijske datoteke

Također jedan od interesantnijih modula za post eksploataciju je „enum_network“. Taj modul prikuplja informacije o mreži ciljanog računala. Te informacije uključuju sučelja,

informacije bežične mreže, otvoreni portovi, aktivne mrežne konekcije, DNS informacije i SSH informacije. S obzirom da smo već prije upoznali osjetljivost nekih od tih mrežnih informacija, poznato nam je koliko je zapravo kritično za metu posjedovanje tih informacija. Da bi koristili taj alat potrebno je odabrati ga s naredbom „use post/linux/gather/enum_network“. Postavke su također već od prije poznate. Potrebno je samo postaviti vrijednost SESSION na 1 naredbom „set SESSION 1“ i pokrenuti modul s naredbom „exploit“.

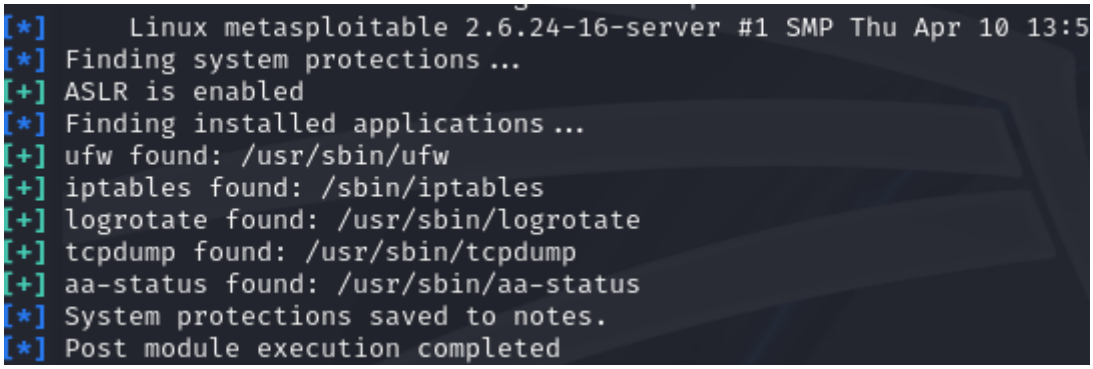
Slika 2.38 Izvršavanje enum_network modula

U terminalu gdje se izvršavao modul možemo vidjeti kako su pronađene informacije za neke od ključnih mrežnih sustava. Osim općenite mrežne konfiguracije sada imamo pristup informacijama o DNS-u, SSH ključevima, portovima pa čak i samog vatrozida.



Slika 2.39Informacije o aktivnim vezama

Po istom principu možemo otkriti i još neke stavke. Na primjer s „enum_protection“ možemo saznati koju vrstu zaštite, odnosno antivirusa se koristi na računalu. To omogućuje napadaču da pronađe način kako najbolje zaobići ga.



Slika 2.40 Pretraga zaštite

Ili „enum_users_history“ pomoću kojeg se mogu izvući informacije o prijašnje prijavljenim korisnicima, uključujući one koji su koristili „sudo“ naredbe za administratorske privilegije. Poznavanjem koji korisnici imaju administratorske privilegije mogu omogućiti „brute force“ napada na prijavu što bi vjerojatno jako dugo potrajalo, ili pokušavanje izvlačenja administratorske lozinke preko hashdump-a i PSExec-a.

Slika 2.41 Povijest korisnika

Poznavanje svih ovih alata i prikupljenih informacija spomenutih u post eksploatacijskoj fazi, omogućili bi daljnje napada na ciljani sustav. Čak i ako uspijemo upasti na računalo koje nije nužno toliko visoko u hijerarhiji sustava, to nam je dovoljno da s vremenom i znanjem kompromitiramo cijeli sustav i napravimo veliku štetu za osobu, osobe ili kompaniju koja nam je meta.

2.4.1. Usporedba post eksploatacijskih modula

Post-eksploatacijski moduli su ključni za daljnje istraživanje i eksploataciju sustava nakon uspješnog napada. Naravno kako je to bio slučaj i za eksploatacijske module, ne postoje moduli koji su bolji ili lošiji jedni od drugih. Korištenje pojedinih modula ovisi o više faktora, uključujući operacijskom sustavu mete i na koji način je dobiven pristup meti. Postoji nekoliko često korištenih post eksploatacijskih modula, a to su Meterpreter, Shell, VNC Injection, Postgres/Postgres_enum i Windows Gather.

Meterpreter je napredni, modularni post-eksploatacijski agent koji je uključen u Metasploit platformu. Razvijen od strane H.D. Moorea, Meterpreter nudi širok spektar funkcionalnosti za daljnje iskorištavanje i istraživanje sustava nakon što je postignut početni pristup. Meterpreter koristi modularni dizajn, što omogućava da se dodatni moduli učitaju po potrebi, bez potrebe za ponovnim uspostavljanjem veze. To znači da se dodatne funkcionalnosti mogu dodati u stvarnom vremenu. Dizajniran je da bude težak za otkrivanje. Meterpreter se izvršava u memoriji, što minimizira tragove na disku i otežava detekciju od strane antivirusnog softvera. Dostupan je za razne platforme, uključujući Windows, Linux, MacOS i Android. To ga čini korisnim alatom za penetracijsko testiranje različitih sustava. Ono što ovaj modul nudi od funkcionalnosti je komandna linija, manipulacija datotekama, sakupljanje informacija poput (prijava i korisničkih računa), *keylogging*, izvršavanje skripti i migraciju procesa što omogućuje stabilnost veze i smanjuje šansu za detekciju. Mana ovog modula je njegova kompleksnost. Meterpreter zahtijeva određeno znanje za učinkovito

korištenje svih mogućnosti upravo zbog velikog broja mogućnosti. Također može biti resursno zahtjevan na ciljanom sustavu što može dovesti do njegovog otkrivanja.

Shell je jedan od najosnovnijih i najjednostavnijih post-eksploatacijskih modula unutar Metasploit platforme. Omogućuje penetracijskim testerima da dobiju pristup komandnoj liniji na ciljanom sustavu nakon uspješnog eksploatacijskog napada. Shell modul omogućuje korisniku pristup komandnoj liniji ciljanog sustava. Na Windows sustavima to je CMD (*Command Prompt*), dok je na Unix sustavima (kao što su Linux i macOS) to *Bash* ili slična ljuska. Penetracijski tester može izvršavati bilo koju naredbu dostupnu na ciljanom sustavu. Ovo uključuje osnovne operacije s datotekama, mrežne naredbe, manipulaciju korisnicima i druge sistemske zadatke. Korisnici mogu pregledavati datotečni sustav, kreirati, brisati, premještati i kopirati datoteke i direktorije. Shell omogućuje prikupljanje osnovnih sistemskih informacija kao što su verzija operativnog sustava, mrežne konfiguracije, aktivni procesi i korisnički računi. Izvršavanje mrežnih naredbi kao što su *ping, netstat, ipconfig*/*ifconfig, tracert*/*traceroute* omogućava testerima da istraže mrežnu okolinu ciljanog sustava. Ali u sporedbi s naprednim post-eksploatacijskim modulima kao što je Meterpreter, Shell nudi ograničene mogućnosti. Nedostaju mu napredne funkcionalnosti kao što su migracija procesa, *keylogging*, *sniffing* i *pivoting*. Izvršavanje naredbi putem Shell-a može biti lakše otkriveno od strane sigurnosnih alata i administratorskih nadzora, jer svaka naredba može ostaviti trag u sistemskim logovima. Za razliku od Meterpretera, Shell ne podržava napredno skriptiranje i automatizaciju zadataka, što može otežati kompleksnije operacije.

VNC Injection je specifičan post-eksploatacijski modul unutar Metasploit platforme koji omogućava penetracijskim testerima daljinsko upravljanje grafičkim korisničkim sučeljem ciljanog sustava. Ovaj modul je posebno koristan kada je potreban vizualni pristup ciljanom sustavu, što omogućava testerima interakciju s GUI aplikacijama koje se ne mogu kontrolirati putem komandne linije. Samim time se dosta razlikuje od predhodno navedenih modula koji koriste terminal kao korisničko sučelje. Omogućava testerima daljinski pristup grafičkom korisničkom sučelju (GUI) ciljanog sustava. To uključuje kontrolu miša, tipkovnice i praćenje zaslona. Koristi VNC (*Virtual Network Computing*) protokol za komunikaciju između ciljanog sustava i napadača. VNC je široko korišten protokol za daljinski pristup GUI-ju. Omogućuje testerima pokretanje i upravljanje GUI aplikacijama koje se ne mogu kontrolirati putem komandne linije, što može biti korisno za određene vrste eksploatacija. Može se koristiti za preuzimanje aktivnih korisničkih sesija na ciljanom

sustavu, omogućavajući testerima da nastave rad gdje je korisnik stao. Također omogućuje testerima snimanje zaslona ciljanog sustava, što može biti korisno za dokumentiranje eksploatacije ili prikupljanje vizualnih dokaza. Mane ovog modula leže u tome da VNC Injection generira mrežni promet koji može biti otkriven od strane mrežnih sigurnosnih alata. Također, aktivni VNC sesija može biti vidljiva korisnicima ili administratorima na ciljanom sustavu. Uz to Budući da VNC Injection omogućava vizualni pristup, aktivnosti penetracijskog testera mogu biti vidljive korisnicima ili administratorima, povećavajući rizik od otkrivanja.

Postgres_enum je post-eksploatacijski modul unutar Metasploit platforme koji je dizajniran za sakupljanje informacija iz PostgreSQL baza podataka na ciljanom sustavu. Ovaj modul je vrlo koristan za penetracijske testere koji žele prikupiti detaljne informacije o konfiguraciji i stanju PostgreSQL baza nakon što su stekli pristup ciljanom sustavu. Modul prikuplja informacije o verziji PostgreSQL servera koji je instaliran na ciljanom sustavu, kao i popis svih korisnika baze podataka, uključujući njihove uloge i privilegije. Modul omogućuje dubinsko prikupljanje informacija o PostgreSQL bazama podataka, što je korisno za daljnju analizu i planiranje napada. Postgres_enum je specijaliziran za rad s PostgreSQL bazama podataka, pružajući detaljne i relevantne informacije koje generički alati možda ne bi prikupili. Također nudi brzu analizu konfiguracije i sigurnosnog stanja PostgreSQL servera, pomažući testerima da identificiraju potencijalne ranjivosti. Nedostaci ovog modula kreću od njegove ograničenosti, odnosno modul je specifičan za PostgreSQL baze podataka i nije primjenjiv na druge sustave za upravljanje bazama podataka (DBMS). Također modul zahtijeva autentifikacijske podatke za pristup PostgreSQL serveru. Ako tester nema odgovarajuće vjerodajnice, modul neće biti koristan, te aktivnosti samog modula mogu biti lako otkrivene on strane administratorskih alata ili sigurnosnih mjera.

Windows Gather je kategorija post-eksploatacijskih modula unutar Metasploit platforme koja je specijalizirana za prikupljanje informacija s ciljanih Windows sustava nakon što je postignut pristup. Ovi moduli omogućuju penetracijskim testerima da prikupe širok raspon podataka koji mogu biti korisni za daljnju analizu, eksploataciju ili planiranje napada. Prethodno je demonstriran rad ovih modula ali nad Linux sustavom. Moduli su identični u funkcionalnosti, samo što su specijalizirani za svoj određeni operacijski sustav, ali prednosti, mane i korištenja su identična. Jedina prava razlika je ta što je Windows enumeracija češće korištena jer veliku većinu korištenih računala u svijetu koristi isključivo Windows operacijski sustav. Moduli prikupljaju osnovne informacije o ciljanom Windows sustavu,

uključujući verziju operativnog sustava, arhitekturu, instalirane zakrpe, pokrenute procese i konfiguraciju mreže. Također prikupljaju podatke o korisničkim računima, njihovim privilegijama i članstvu u grupama. Moduli poput hashdump omogućuju prikupljanje NTLM hash vrijednosti lozinki, dok moduli poput kiwi (Mimikatz) mogu izvući lozinke u čistom tekstu iz memorije. Osim informacija o korisnicima i sustavu, moduli prikupljaju podatke o konfiguraciji mreže, uključujući IP adrese, aktivna mrežna sučelja, DNS postavke i dijeljene mrežnih resursa. Nedostaci ovih modula su to što su specifični za Windows sustave i nisu primjenjivi na druge operativne sustave kao što su Linux ili macOS, što znači da su optimizirani za prikupljanje relevantnih informacija iz ovog sustava. Neki od modula koji prikupljaju osjetljive informacije (poput lozinki) mogu povećati rizik od otkrivanja zbog njihovih operacija na sustavu. Korištenje više modula može zahtijevati napredno poznavanje Metasploit platforme i Windows sustava kako bi se iskoristile sve mogućnosti.

Tablica 3 Usporedba po kriterijima

Modul	**Funkcionalnost**	**Jednostavnost korištenja**	**Diskretnost**	**Specifičnost**
Meterpreter	Vrlo visoka	Srednja	Visoka	Sveobuhvatan
Shell	Osnovna	Visoka	Niska	Sveobuhvatan
VNC Injection	Srednja	Visoka	Niska	Specifičan za GUI
Postgres_enum	Srednja	Srednja	Srednja	Specifičan za DB
Windows Gather	Visoka	Srednja	Srednja	Specifičan za Win

Tablica 3 prikazuje usporedbu prethodno navedenih modula prema kriterijima. Ono što možemo zaključiti je da Svaki od ovih post-eksploatacijskih modula ima svoje specifične primjene i prednosti:

- **Meterpreter** je najmoćniji i najfleksibilniji alat za napredne korisnike koji žele maksimalne mogućnosti.
- **Shell** je idealan za brzo i jednostavno izvršavanje osnovnih naredbi.
- **VNC Injection** je koristan za situacije gdje je potreban vizualni pristup GUI-u ciljanog sustava.
- **Postgres_enum** je specijaliziran za rad s PostgreSQL bazama podataka.
- **Windows Gather** je odličan za dubinsko prikupljanje informacija na Windows sustavima.

Izbor modula ovisi o specifičnim potrebama penetracijskog testiranja i karakteristikama ciljanog sustava.

Zaključak

U ovom radu istražene su različite metode penetracijskog testiranja koristeći Metasploit platformu. U uvodnom dijelu smo predstavili osnovne koncepte penetracijskog testiranja i važnost ovog procesa u današnjem digitalnom okruženju. Nakon toga, detaljno smo opisali Metasploit kao jedno od najmoćnijih i najpopularnijih alata u oblasti sigurnosti informacija.

Tokom ovog rada, demonstrirano je nekoliko metoda penetracijskog testiranja, uključujući skeniranje ranjivosti, eksploataciju, post-eksploatacijske aktivnosti, te usporedbe tih alata. Također, obratili smo pažnju na etičke aspekte korištenja ovih tehnika, naglašavajući važnost dobivanja dozvole i pridržavanja zakonskih okvira jer provođenje ovih aktivnosti nad nečijim sustavom bez dozvole, može biti kažnjavan novčanom i zatvorskom kaznom. Kroz ovaj rad pokazalo se da Metasploit, zahvaljujući svojoj fleksibilnosti i širokom spektru modula, može efikasno identificirati i iskoristiti ranjivosti u različitim sistemima. Korištenje Metasploit platforme omogućava stručnjacima za sigurnost da realno procijene sigurnosnu situaciju i pruže preporuke za poboljšanje zaštite sustava. Metasploit platforma ključan je alat u arsenalu svakog stručnjaka za informacijsku sigurnost. Korištenjem ovog alata, organizacije mogu bolje razumjeti svoje sigurnosne slabosti i implementirati odgovarajuće mjere zaštite.

Penetracijsko testiranje je važna tema o kojoj bi svi povezani s informatičkom industrijom trebali biti svjesni i imati bar nekog znanja. S upotrebom interneta koja stalno raste, područje računalne sigurnosti postalo je vrlo izazovna tema, ne samo za tvrtke i organizacije, već i za obične korisnike pa čak i javne i medicinske službe, te nismo sigurni samo s antivirusnim programom. Danas postoji veća šansa da će osoba biti hakirana nego opljačkana, a alati za probijanje dobivaju puno više pažnje, budući da u njihovoj „proizvodnji“ nema ograničenja i svakim danom nastaju novi alati. Otvoreni alati mogu se prilagoditi individualnim potrebama, kao na primjer alat za probijanje koji može hakirati satelite i mijenjati prognoze vremenskih prilika, ili aktivirati nuklearno oružje, sve su to moguće opcije ili opcije koje će tek biti moguće s obzirom na ubrzan napredak tehnologije, pogotovo umjetne inteligencije. Korištenjem ovih alata, se mogu hakirati gotovo svi uređaji, od računala i servera do pametnih termostata, do medicinskih uređaja ili čak automobila. Nekadašnja znanstvena fantastika je današnja stvarnost.

Literatura

Aslan, A. O.-O. (2024). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks and Solutions. *Electronics*.

David Kennedy, J. O. (2011). *METASPLOIT The Penetration Tester's Guide.* San Francisco: No Starch Press.

Denis, M., Zena, C., & Hayajneh, T. (2016). Penetration testing: Concepts, attack methods, and defense strategies. *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT).* Farmingdale, NY, USA: IEEE.

Gupta, H., & Kumar, R. (2015). Protection against penetration attacks using Metasploit. *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions).* Noida, India: IEEE.

ITU. (2008). *ITU Council* . Geneva: ITU.

Kott, A. (2023). *Autonomous Intelligent Cyber Defense Agent (AICA).* Adelphi, MD, USA: Springer.

Maravić, M. (2021). Uređaj za mrežnu sigurnost s Raspberry Pi 4 i Kali Lunux OS. Slpit, Hrvatska: AAI@EduHR.

Merriam-Webster. (24. 7 2024). *Merriam-Webster*. Dohvaćeno iz Merriam-Webster.com dictionary: https://www.merriam-webster.com/dictionary/cybersecurity

Offsec Services. (n.d.). Dohvaćeno iz Kali Linux: https://www.kali.org/docs/

Rahalkar, S. (2017). *Metasploit for Beginners.* Packt.

Raphaël Hertzog, J. O. (2017). *Kali Linux Revealed - Mastering the Penetration Testing Distribution.* Cornelius NC: Offsec Press.

Rapid7. (n.d.). Dohvaćeno iz Metasploit: https://www.metasploit.com/

Rapid7. (n.d.). Dohvaćeno iz Metasploitable: https://docs.rapid7.com/metasploit/metasploitable-2/

Rossouw von Solms, J. v. (2013). From information security to cyber security. *Computers & Security*, 97-102.

Sagar Rahalkar, N. J. (2019). *The Complete Metasploit Guide.* Birmingham, UK: Packt Publishing.

Singh, G. D. (2022). *The Ultimate Kali Linux Book: Perform advanced penetration testing using Nmap, Metasploit, Aircrack-ng, and Empire.* Packt Publishing Ltd.

Stanger, J. (2021). *CompTIA*. Dohvaćeno iz Open-Source Tools for Pen Testing: Putting Your Technique to Work: https://www.comptia.org/blog/open-source-tools-for-pen-testing

Wenye Wang, Z. L. (2013). Cyber security in the Smart Grid: Survey and challenges. *Computer Networks*, 1344-1371.

Skraćenice

API - Application Programming Interface.
AI – Artificial inteligence
DNS – Domain Name System
AICA - Autonomous Intelligent Cyber Defense Agent
VM – Virtual Machine
LAN – Local Area Network
IP – Internet Protocol
FTP – File Transfer Protocol
VNC - Virtual Network Computing