Napadi tipa Man-in-the-middle na HTTPS i njihovo prepoznavanje

Varelija, Dominik

Sveučilište u Rijeci – Odjel za informatiku

Jednopredmetni preddiplomski studij informatike

Dominik Varelija

Napadi tipa Man-in-the-middle na HTTPS i njihovo prepoznavanje

Završni rad

Mentor: dr. sc. Vedran Miletić

Rijeka, rujan 2018

Sadržaj

1. Uvod

Ovaj rad osvrće se na problem zaštite podataka koji se šalju putem interneta. Do problema dolazi u slučaju kada se šifrirana poruka koju klijent šalje u nekom trenutku između klijenta i servera presreće, bilo u slučaju da se presretanje odvija na računalu klijenta (npr. antivirusi) ili negdje na putu prema serveru (npr. Unutar lokalne mreže). Ovaj rad također se bavi i načinima otkrivanja napada i posrednika u cilju zaštite klijenta i servera. Velik problem ovog napada je nedovoljna upoznatost ljudi sa potencijalnim opasnostima krađe identiteta ili senzitivnih podataka. Velike kompanije koje koriste programe za blokiranje sadržaja unutar tvrtke nenamjerno svojim zaposlenicima mogu komunikaciju preko mreže ostaviti ranjivom na MITM zbog toga što ti programi često koriste lošu enkripciju ili imaju ranjivosti preko kojih napadač može doći do čistog teksta poruke. Cilj ovog seminara je upoznavanje korisnika interneta sa načinima na koje napadači mogu doći do njihovih podataka kako bi mogli izbjeći takve situacije i zaštititi svoje podatke i komunikacije zadržati privatnima.

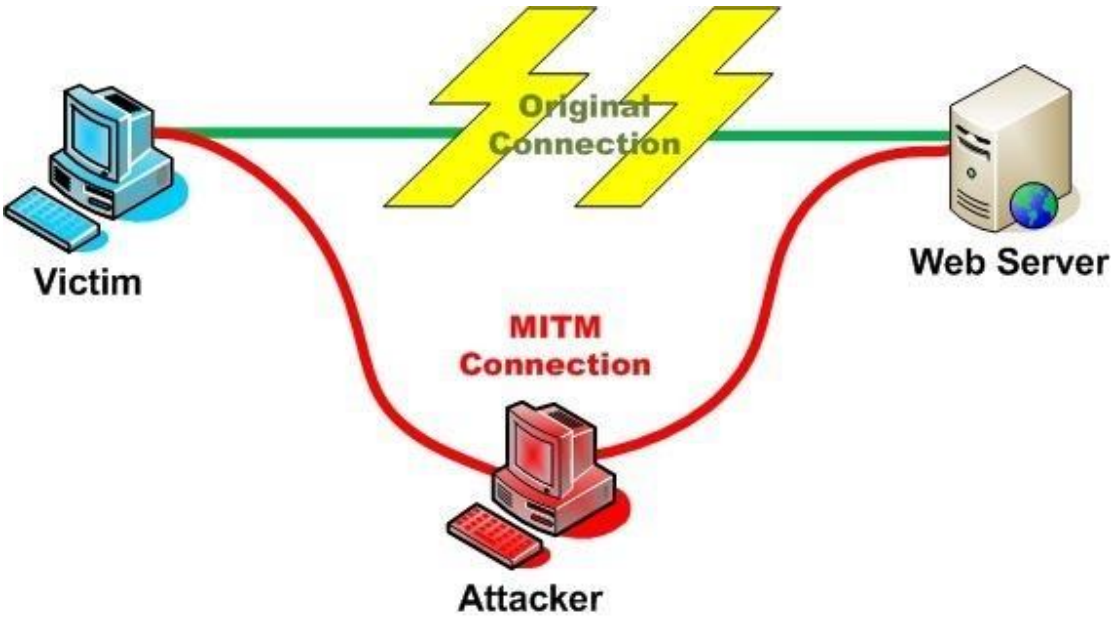
2. Man-in-the-middle

2.1 Što je MITM?

Man in the middle je vrsta napada u kojem napadač upada u komunikaciju između klijenta i servera tako da ih uvjeri da klijent i server komuniciraju direktno dok napadač u stvari preuzima cijelu komunikaciju bez znanja ostalih sudionika komunikacije. Napad može jedino biti uspješan ako napadač uvjeri obje strane komunikacije da je druga strana upravo ona kojoj ta strana želi pričati. Većina sigurnosnih protokola koristi neku vrstu provjere autentičnosti na rubovima komunikacije kako bi spriječila MITM napade. SSL (eng. Secure Sockets Layer) protokol koristi se upravo kako bi obje strane komunikacije mogle provjeriti autentičnost tako da obje strane budu provjerene od pouzdane treće koja ih ovjeri certifikatom.

Objasnimo to sa tri sudionika: Bob, Mark i Tim. Bob i Mark pokušavaju komunicirati dok Tim pokušava presresti komunikaciju. Bob šalje Marku zahtjev za Markovim javnim ključem. Ako Mark pošalje svoj ključ Bobu ali Tim presretne poruku, Tim može započeti sa napadom. Tim šalje krivotvorenu poruku prema Bobu koji zaključuje da je dobio poruku od Marka ali poruka sadrži Timov javni ključ. Bob koji vjeruje da je ključ koji je dobio bio od Marka, kriptira poruku sa Timovim ključem i šalje nazad prema Marku. Tim presreće i tu poruku, izmjenjuje ju i ponovo šifrira, ovaj put sa Markovim javnim ključem koji je originalno Mark slao Bobu. Kad Mark dobije poruku on vjeruje da je poruka od Boba.

Ovakva vrsta napada je najjednostavniji oblik MITM napada i može se izbjeći tako da Bob i Mark svoje identitete potvrde uz pomoć treće strane kojoj vjeruju ili provjerama da nije došlo do izmjene poruke između Boba i Marka.



Slika 1: MITM napad, U našem slučaju: Tim-Attacker, Mark i Bob – Victim i web server (https://null-byte.wonderhowto.com/forum/do-mitm-attack-with-websploit-0180442/)

2.2. Tipovi man in the middle napada

RAP (eng. Rouge Access Point) - Uređaji koji se spajaju preko bežične veze često se automatski spajaju na pristupnu točku koja ima najjači signal što znači da ako napadač postavi svoju pristupnu točku, koja ima signal jači od prave pristupne točke, može prevariti ostale uređaje da se spoje na tu pristupnu točku nakon čega sav promet koji ide preko te lažne pristupne točke može biti izmijenjen ili pročitan od strane napadača. Za ovu vrstu napada, napadač samo treba fizički blisku pristupnu točku meti svog napada.

ARP (Address Resolution Protocol) Spoofing – ARP se koristi unutar lokalne mreže kako bi pristupna točka znala kome poslati promet koji dolazi u lokalnu mrežu. Kad klijent želi nekome poslati nešto preko mreže tad putem ARP pogleda ip adressu servera kojem želi poslati podatke. Napadač koji želi presresti podatke ovdje se predstavlja kao drugi klijent i odgovara sa svojom MAC adresom na podatke koji nisu bili za njega. U slučaju da uspije poslati podatke u pravo vrijeme, napadač dobiva pristup svim podatcima koji se razmjenjuju između legitimnog servera i klijenta te tako može doći i do tokena te sesije i dobiti potpuni pristup aplikaciji klijenta.

MDNS spoofing – mDNS je sličan kao dns ali se izvodi na pouzdanim mrežama kao na lokalnim mrežama. Na takvim mrežama najčešće se nalaze printeri, televizori. Napadač u tom slučaju odgovara sa lažnim podatcima na taj multicast i pri tome napadač postaje pouzdan dok god se nalazi u lokalnoj memoriji od pouzdanih adresa.

2.3. Tehnike MITM napada

Njuškanje (eng. Sniffing) – napadač koristi alate za hvatanje paketa koji se šalju unutar mreže i pregledava iste na niskom nivou. Koristeći specifične bežične uređaje napadač može vidjeti pakete koji nisu bili namijenjeni njemu.

Ubacivanje paketa (eng. Packet injection) – napadač koristeći uređaj koji može osluškivati pakete koji se šalju unutar mreže osluškuje promet u koji se želi ubaciti. Pokušava odrediti koliko često i kada se paketi šalju te ubacuje svoj maliciozni paket u legitiman tok podataka. Nakon toga dobiva pristup komunikaciji i mogućnost izmjene ili čitanja poruka.

Preuzimanje sesije (eng. Session hijacking) – Napadač dolazi do tokena sesije od legitimnog korisnika i na taj način dobiva pristup web aplikaciji kao da je taj korisnik. Najčešće do tokena sesije dolazi metodom ubacivanja paketa.

Skidanje enkripcije (eng. SSL stripping) – napadač se ubacuje u sigurnu vezu i preuzima pakete koji dolaze sa HTTPS protokolom te ih mijenja u HTTP protokol tako da korisnik šalje svoje podatke sada nezaštićenom vezom u tekstualnom obliku bez ikakve enkripcije koji je lako čitati.

2.4. Presretanje TLS protokola

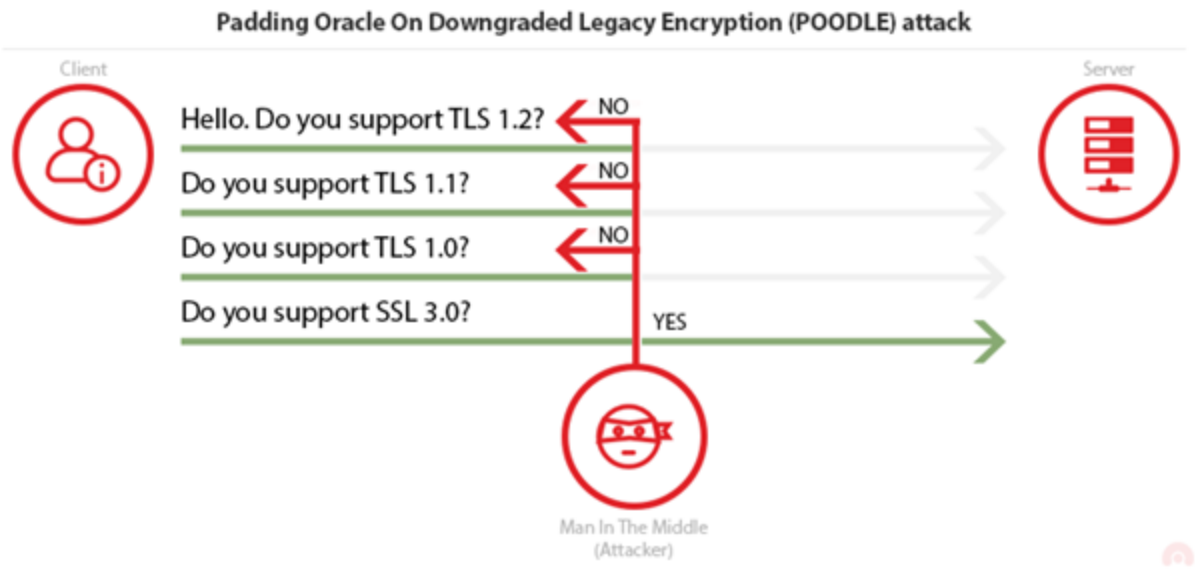
TLS (Transport Layer Security) protokol je protokol koji bi zamjenjuje SSL (Secure Sockets Layer) protokol. Koristi se za očuvanje privatnosti komunikacije između dva računala na mreži. Kako bi veza bila sigurna koristi se simetrična kriptografija kojom se sav promet na mreži kriptira, a ključ za svaku vezu se jedinstveno generira. Identiteti sudionika mogu se provjeriti putem provjere javnog ključa i kao dodatna sigurnost koristi se provjera integriteta poruke putem čega je moguće otkriti ako je došlo do gubitka dijela poruke ili namjerne izmjene poruke za vrijeme prijenosa. Velika većina presretanja veze odvija se na računalu klijenta kada softver na strani klijenta prilikom instalacije potpiše svoj certifikat te na taj način dobiva pristup podatcima koji se šalju putem HTTPS protokola uzimajući ulogu transparentnog proxy-a. Takvi programi prekidaju i dešifriraju TLS sesiju od strane korisnika, analiziraju poruku u tekstualnom obliku te započinju novu TLS vezu sa odredišnim klijentom. Kako bi zaobišli validaciju protokola koriste certifikate koji su kreirani od strane programa za vrijeme instalacije. Certifikati koji se nalaze na računalu korisnika i koji su instalirani sa administratorskim ovlastima se prihvaćaju kao takvi.

Programi koji presreću HTTPS veze najčešće su middleboxes, antivirusni programi i računalni virusi. Middleboxes su programi koji se koriste za prikupljanje podataka o mreži, filtriranje sadržaja na mreži i

detekciju neželjenih programa. Primjeri middleboxeva su vatrozidi, sustavi za detekciju upada, NAT (Network Address Translator) i WAN optimizatori. U istraživanju sigurnosti TLS veze pokazano je da skoro svi middleboxevi smanjuju sigurnost veze, a da 58% njih sadrži ranjivosti koje kasnije mogu biti iskorištene za MITM napad. Prema istraživanju dvanaest proizvođača antivirusnih programa prilikom instalacije programa ubacuje novi certifikat potpisan od strane proizvođača kako bi taj program mogao presretati vezu [1]. Nadalje od 20 testiranih antivirusa koji su bili posrednici u vezi njih 18 je na neki način smanjilo sigurnost veze, bilo korištenjem RC4 kriptografije, nepravilnim provjeravanjem certifikata ili korištenjem slabih šifriranja (eng. export ciphers) koja su ranjiva na FREAK napad [2]. Osim middleboxeva i antivirusnih programa podatke na mreži presreću i neželjeni i maliciozni softver. U ovom slučaju maliciozni softveri koriste programe koji preusmjeravaju podatke na mreži putem dobro poznatih softvera za preusmjeravanje i filtriranje prometa poput Komodia ili NetFiltera [3].

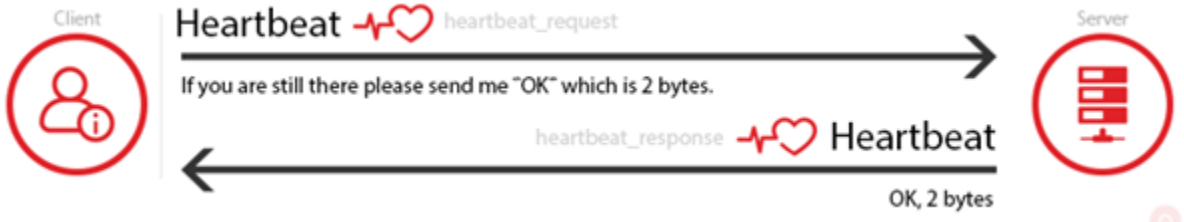
2.5. Napadi na TLS

POODLE (Padding Oracle On Downgraded Legacy Encryption) napad koristi dvije ranjivosti. Kada klijent pošalje „Client Hello“ i pri tome podržane verzije SSL i TLS protokola napadače presretne promet koristeći jednu od MITM tehnika i pokušava oponašati server dok klijent ne prihvati ranjivu SSL verziju protokola za komunikaciju. Nakon što klijent i server ostvare nesigurnu vezu, napadač započinje s POODLE napadom. Problem nastaje kad server ne može provjeriti vrijednost padding-a poruke nego samo njezinu dužinu. To znači da server ne može provjeriti da li je padding poruke bio izmijenjen. Napadač može dešifrirati tekst poruke šifriranog bloka tako da mijenja vrijednost padding i gleda odgovore servera. Treba najviše 256 zahtjeva prema serveru kako bi napadač dobio vrijednost jednog padding bajta. Napadač ne mora znati koja metoda šifriranja se koristi nego može koristiti automatske alate kako bi došao do teksta poruke slovo po slovo i na taj način doći do lozinke, sesije ili drugih podataka. POODLE napad se može spriječiti sa server strane tako da server ne prihvaća SSL (bilo koju verziju) te da koristi samo TLS od verzije 1.2. Sa strane klijenta moguće je isključiti SSL protokole kako bi pretraživač koristio samo TLS i kako se veza ne bi mogla uspostaviti preko nesigurnog protokola.



Slika 2 Vizualizacija POODLE napada (https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/)

Heartbleed je vrsta napada na popularnu kriptografsku biblioteku otvorenog tipa OpenSSL. Napad je koristio metodu „heartbeat“ u TLS protokolu koja inače služi za održavanje veze između klijenta i servera. U normalnoj komunikaciji metoda „heartbeat“ radi ovako: klijent serveru pošalje poruku koja sadržava podatak i veličinu podatka i na tu poruku server mora odgovoriti sa istom heartbeat porukom koja sadrži podatke i veličinu podataka koje je klijent poslao. Do problema dolazi kad klijent pošalje lažnu veličinu podataka a server pokušava odgovoriti na poruku i onda kako bi uspješno odgovorio šalje podatak koji je dobio od klijenta i slučajne podatke koje u tom trenutku ima kako bi zadovoljio veličinu poruke. Na taj način moguće je od servera dobiti nešifrirane podatke poput privatnog ključa servera koji bi kasnije služio za dešifriranje svih podataka koji idu prema serveru. Prevencija je korištenje zadnje verzije OpenSSL-a u kojoj je propust zakrpan.



Slika 3 Standardna heartbeat poruka

Slika 4 Maliciozna Heartbeat poruka (Heartbleed) (https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/)

CRIME (eng. Compression Ratio Info-leak Made Easy) je ranjivost pronađena u TLS kompresiji. Kompresiju u TLS protokolu je moguće odabrati, a najčešće su poruke slane bez kompresije. Kompresija se koristi kako bi se smanjila količina podataka slanih preko veze. Najčešća verzija kompresije je DEFLATE koja koristi metodu zamjene alfanumeričkih znakova koji se ponavljaju sa pokazivačima i vrijednostima. Pretpostavimo da napadač želi kolačić (eng. Cookie) i zna da neka web stranica naziva kolačić sa „adm“. Napadač tada ubacuje u kolačić vrijednost 0. Nakon toga napadač samo mijenja alfanumeričke vrijednosti i gleda odgovor servera, ako je veličina odgovora od servera manja od početne znači da je pogodio jedu vrijednost od kolačića i nastavlja sa ostalim znakovima dok ne dobije cijeli kolačić i time sesiju korisnika. Ovo je problem unutar pretraživača na strani korisnika i kako bi se izbjegao treba koristiti zadnje verzije pretraživača.

Slika 5 Vizualizacija CRIME napada (https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/)

3. HTTPS i sigurnost na internetu

3.1 Što je HTTPS i zašto je važan?

HTTPS (Hypertext Transfer Protocol Secure) je nadogradnja HTTP protokola kako bi se ostvarila sigurna komunikacija preko računalne mreže, najčešće Interneta [4]. HTTPS protokol korištenjem SSL ili TLS protokola postiže očuvanost integriteta i privatnosti poruka slanih između klijenta i servera tako da sve poruke budu šifrirane. Takav način slanja poruka je najbolji način za sprječavanja MITM napada zbog toga što i u slučaju da napadač presretne komunikaciju i dalje ne može vidjeti tekst poruke ili mijenjati poruku. SSL i TLS protokoli koriste enkripciju javnim ključem. Poruke se šifriraju javnim ključevima sudionika u komunikaciji a mogu se pročitati samo ako su dešifrirane korištenjem privatnog ključa osobe za koju je poruka namijenjena. Na ovaj način riješen je problem poruke u prijenosu, ali kako osigurati da zbilja komuniciramo sa osobom s kojom želimo a ne sa napadačem koji nas pokušava prevariti? To je riješeno korištenjem certifikata kojim se potvrđuju identiteti osoba (ili servera) u komunikaciji tako da se zatraži potvrda identiteta od treće strane, najčešće institucije kojoj obje strane vjeruju.

Slika 6 Poruka bez HTTPS enkripcije čitljiva napadaču

Slika 7 Poruka sa dobrom enkripcijom, beskorisna napadaču (https://www.acunetix.com/blog/articles/tls-security-what-is-tls-ssl-part-1/)

3.2 Povijest TLS/SSL protokola

SSL protokol predstavljen je 1994. godine za pretraživač „Netscape Navigator“ kao način za sigurnu kupovinu preko interneta jer prije toga nije bilo moguće sigurno poslati podatke putem HTTP protokola, također nije bilo moguće detektirati MITM napade. HTTP protokol bio je učahuren u SSL protokol kako bi se postigla sigurna veza. SSL 1.0 verzija protokola nikad nije zaživjela zbog velikih sigurnosnih propusta. SSL 2.0 verzija izdana 1995. godine bila je korištena sve do 2011. godine kada je IETF (Internet Engineering Task Force) zabranila korištenje protokola zbog brojnih nedostataka. SSL 2.0 poruke koristile su MD5 algoritam za autentifikaciju poruka, rukovanje nije bilo zaštićeno što je omogućavalo MITM napade tako što bi klijent odabrao slabiju verziju šifriranja poruke i sesije su mogle biti lako prekinute ubacivanjem TCP FIN (Transfer Control Protocol sa zastavicom FIN za prekid veze) paketa. TLS protokol napravljen je 1999. kao evolucija SSL protokola. Prva verzija TLS-a (TLS 1.0) nije bila znatno drugačija od SSL-a (SSL 3.0) jer je bila napravljena kako bi omogućila komunikaciju između SSL i TLS protokola, no to nije bilo dobro rješenje jer je moglo doći do oslabljene sigurnosti ako se koristi SSL 3.0 protokol umjesto TLS. SSL 3.0 protokol izbacuje se iz upotrebe nakon što je 2014. pokazana ranjivost na POODLE napad te je 2015. godine preporučljivo izbjegavati ga prema RFC 7568 [5]. Osim ranjivosti na POODLE SSL 3.0 koristio je SHA-1 i MD5 koji se smatraju nesigurnima te nije imao sve mogućnosti od TLS protokola te nije mogao biti nadograđen kako bi se te mogućnosti dopunile. TLS 1.0 protokol sadrži načine na koje je moguće smanjiti sigurnost veze na SSL protokol te zbog toga se koriste samo protokoli verzija 1.1 i 1.2 [6]. TLS verzija 1.3 definirana je u kolovozu 2018. godine prema RFC 8446 [7]

3.3 TLS certifikati i CA (eng. Cetrificate Authority)

CA je pravna osoba koja dodjeljuje digitalne certifikate za TLS i SSL, također ta pravna osoba ima vlastiti certifikat koji koristi njihov privatni ključ kako bi potpisao bilo koji izdani certifikat od njihove strane. Takav certifikat se naziva izvorišni certifikat (eng. Root Certificate). Izvorišni cerifikat zajedno sa javnim ključem je instaliran i smatra se vjerodostojnim od strane preglednika. Certifikati se najčešće koriste za dokazivanje sigurne veze (HTTPS) i za potpisivanje digitalnih dokumenata. Certifikati služe kako bi zaobišli problem MITM napada. Klijent prije uspostave sigurne veze koristi CA certifikat kako bi potvrdio identitet servera s kojim želi komunicirati. Uobičajeno je da preglednici koriste certifikate instalirane na računalu korisnika i da ih smatraju vjerodostojnima, no tu dolazi do problema jer maliciozni klijent može preskočiti sve provjere

certifikata i tako prevariti korisnike. Upravo izvorišnim certifikatima instaliranim na računalu korisnika većina antivirusnih programa uspijeva prekinuti i započeti TLS vezu kako bi provjerili ima li malicioznih podataka. Uz antivirusne programe programi za analiziranje mreže mogu koristi iste metode kao i maliciozni softver poput Superfish programa.

3.4 Kako prepoznati TLS presretanje?

Sa strane servera jedan od načina prepoznavanja presretnute veze je analiziranje paketa Client Hello, TLS rukovanja i Korisničkog zaglavlja (eng. User-agent Header). Preglednici koriste određene vrste šifriranja i obično imaju standardiziran Client Hello koji se razlikuje za razne preglednike. Na primjer Mozilla Firefox ima gotovo identičan Client Hello bez obzira na to sa kojeg operacijskog sustava ili platforme šalje paket. Svi parametri unutar Client Hello uključujući šifriranja i kompresije su zapisani unutar preglednika i nisu izmjenjivi. Korisnici mogu samo isključiti pojedina šifriranja ali ne mogu dodati svoja. Zbog svega navedenoga prilikom analiziranja paketa koji je navodno došao od strane Firefox preglednika ukoliko se jedan od dijelova ne poklapa sa redoslijedom metoda koji je zapisan od strane Firefox-a ili se koristi šifriranje koje nije podržano preglednikom znamo da je došlo do presretanja podataka [8]. Za preglednik Google Chrome situacija je malo kompliciranija kod otkrivanja da li je došlo do presretanja zbog toga što Chrome koristi različite Client Hello poruke, različite vrste šifriranja i poredak ekstenzija na različitom hardveru i u različitim verzijama. U tom slučaju možemo analizirati i sa sigurnošću reći da je došlo do presretanja samo u slučaju da se u Client Hello poruci pojavi vrsta šifriranja za koju znamo da Chrome ne podržava. Ovom metodom možemo saznati da veza nije sigurna zbog presretanja, ali ne znamo tko je presretnuo vezu. Kako bi saznali tko presreće vezu moramo analizirati različite Client Hello poruke od strane raznih programa kako bi ih mogli usporediti. Na temelju različitih Client Hello poruka možemo identificirati programe zaslužene za presretanje veze. Procijenjeno je da je 5-10% svih veza presretnuto.

Tablica 1 Presretanje veze [9]

Promatrana Stranica	Postotak presretnutih veza		
	Bez presretanja	Moguće presretanje	Presretnuto
Cloudflare	88.6%	0.5%	10.9%
Firefox	96.0%	0.0%	4.0%
E-commerce	92.9%	0.9%	6.2%

3.5 Kako osigurati vezu prema internetu

HTTPS Everywhere je dodatak za preglednike (Firefox, Chrome, Opera) koji je razvijen sa svrhom da sve stranice kojima pristupamo koriste HTTPS vezu. Mnogo stranica na internetu podržavaju neku verziju HTTPS enkripcije ali primarno koriste nezaštićenu HTTP verziju stranice. HTTPS Everywhere u tom slučaju traži od stranice HTTPS verziju i zahtjeve prepisuje u HTTPS. Važno je napomenuti da HTTPS Everywhere može zaštititi samo veze ako je moguće ostvariti HTTPS vezu sa serverom, jer neki serveri ne podržavaju nikakvu verziju HTTPS-a. Moguće je i blokirati sve HTTP veze kako bi bili sigurni u to da je sav naš promet zaštićen.

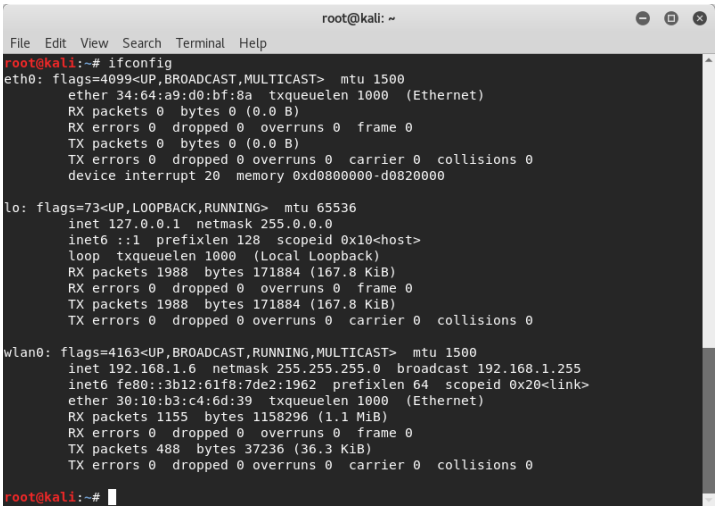
Let's Encrypt je besplatna i automatizirana CA otvorenog koda napravljena od strance Internet Security Research Group. Za razliku od ostalih CA ova organizacija je besplatna i zbog toga mnogo lakša i brža za korištenje jer ne moramo čekati potvrdu servera, potvrdni e-mail i paziti na obnovu certifikata. Projekt podržavaju velike kompanije poput Googla i Mozille i cilj je da HTTPS postane primarni protokol na internetu. Let's encrypt podržava samo certifikate koji se potvrđuju domenom jer je to jedini certifikat koji se može potpuno automatizirati, a svi izdani ili povučeni certifikati su javno dostupni svima.

4. Primjer MITM napada

4.1. Priprema

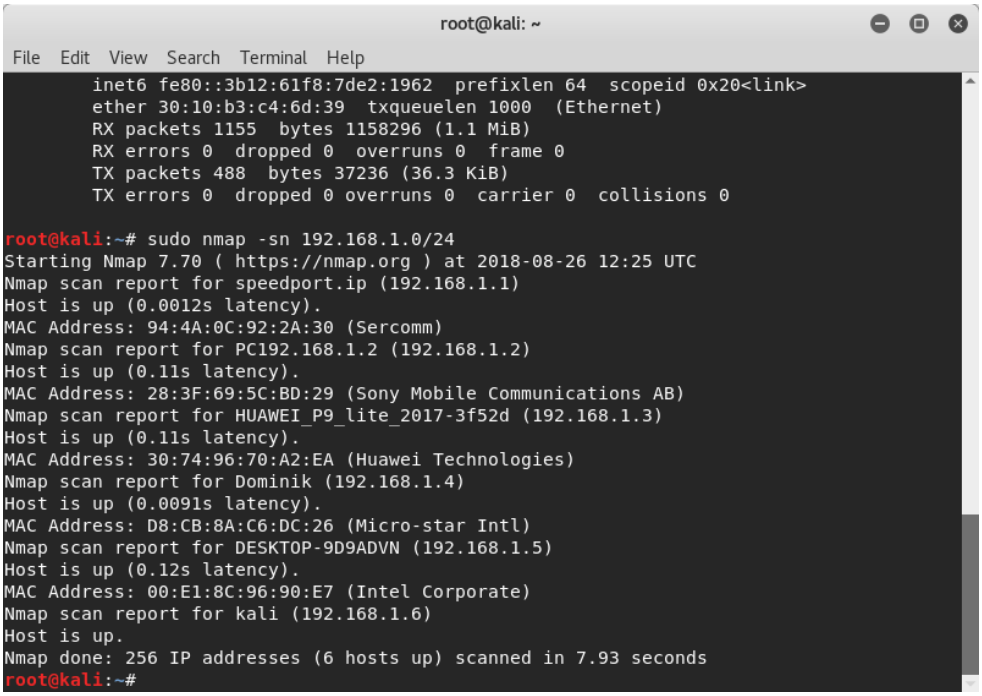
Za izvedbu MITM napada korišten je operacijski sistem Kali [11] namjenjen za testiranje sigurnosti sustava, program Websploit [12] za izvedbu MITM napada i automatsko postavljanje postavki prosljeđivanja paketa unutar mreže i program Nmap [13] za pronalazak uređaja na lokalnoj mreži.

Prvi korak je dobiti pristup mreži na kojoj želimo izvesti napad, u ovom slučaju lokalna mreža. Pretpostavimo da smo došli do pristupa mreži i želimo saznati sve uređaje koji se nalaze na istoj mreži. Pomoću *ifconfig* naredbe dobivamo informacije o našem računalu poput naše ip adrese unutar LAN mreže i naziva našeg wi-fi uređaja (wlan0).



Slika 8 ifconfig

Nmap program otkriva ostale uređaje unutar mreže, njihove ip i MAC adrese te naziv uređaja kako bi lakše pronašli našu metu i njihovu ip adresu koja će nam trebati kako bi znali gdje trebamo slati presretnute podatke.

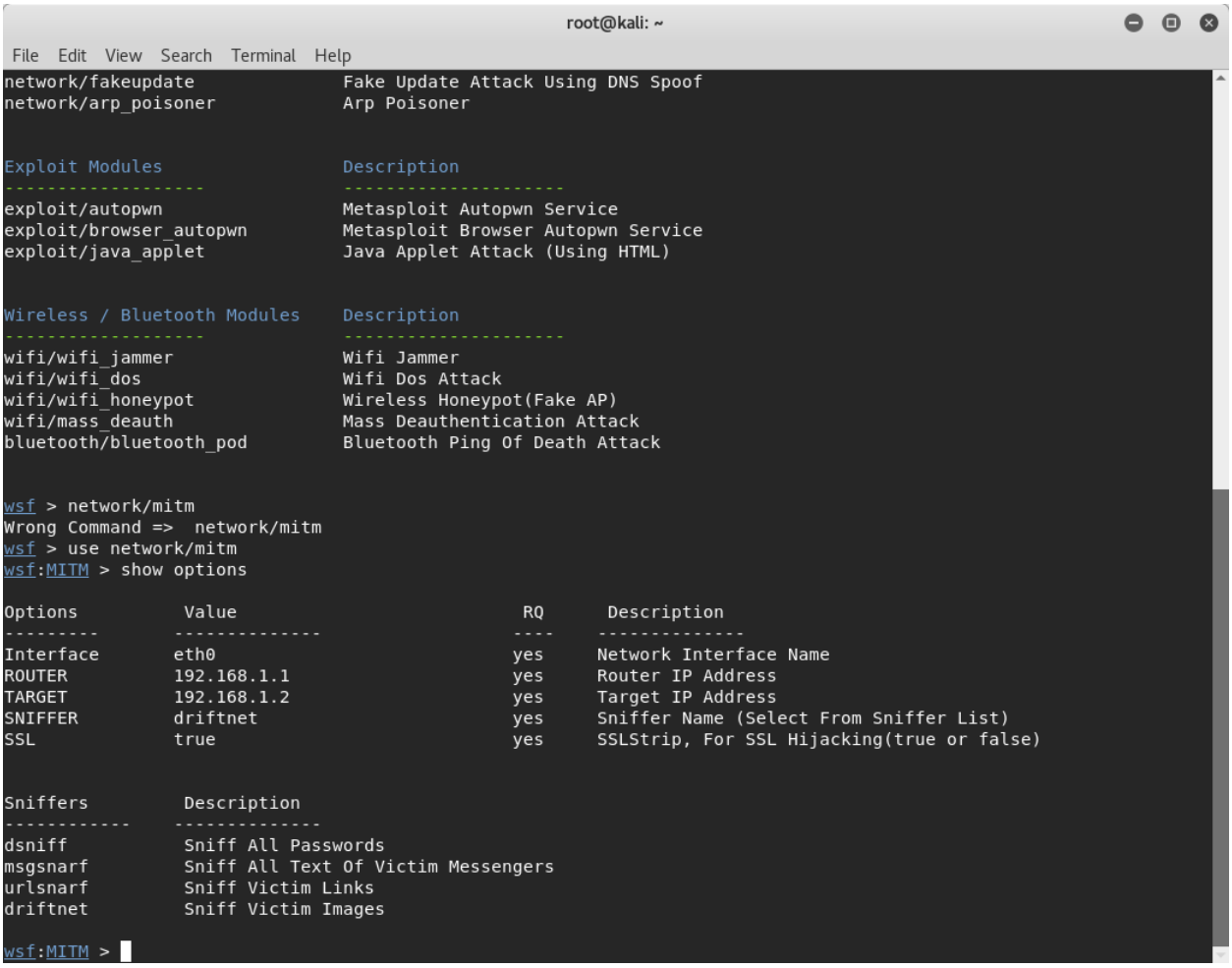


Slika 9 pronalazak routera i mete

Nakon što smo identificirali našu metu (ovdje Sony mobilni uređaj) započinjemo s ubacivanjem u komunikaciju između naše mete i router-a.

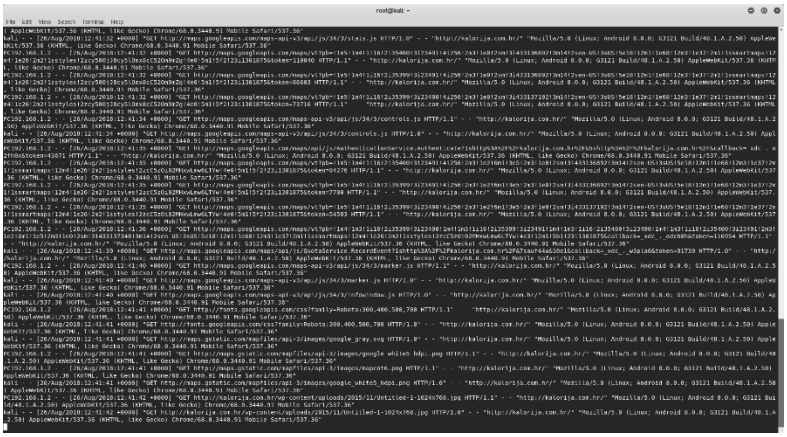
4.2. Websploit

Nakon pokretanja Websploit-a korsitimo naredbu *show modules* kako bi vidjeli sve njegove mogućnosti. Kako bi izveli MITM napad koristimo *network/mitm* modul. Zatim postavljamo opcije napada tako da se ubacimo između routera i mete.

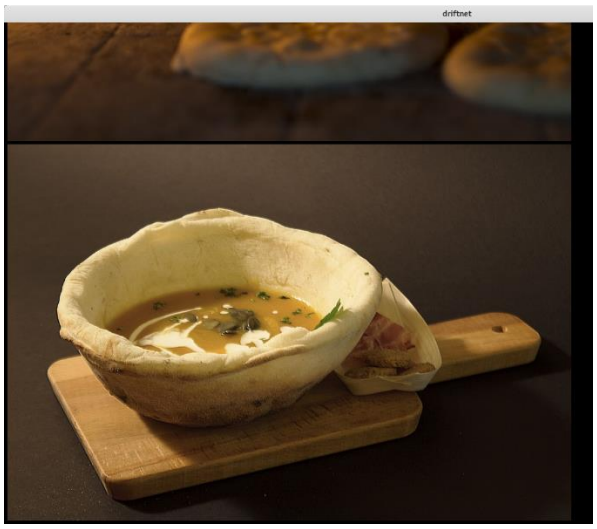


Slika 10 sučelje websploita

Websploit koristi ARP spoofing tehniku i ubacuje nas u komunikaciju između mete i routera i dobivamo pristup svim paketima koji su poslani. Ukoliko meta ne koristi TLS protokol moguće je izvući podatke iz paketa poput teksta, lozinki ili slika.



Slika 11 svi url-ovi kojima je meta pristupila



Slika 12 Lijevo: primjer pogleda na stranicu od strane mete; Desno: prikupljanje slika poslanih mrežom od strane napadača

5. Zaključak

Cilj ovog seminara je obrazovanje ljudi i upućivanje na sve veći problem zaštite podataka na internetu. Broj korisnika interneta raste iz godine u godinu i internetu pristupamo sa svih uređaja koje koristimo i bilo gdje da se nalazimo, često zaboravljajući na to koliko svojih intimnih podataka šaljemo upravo preko veza u čiju sigurnost nismo sigurni. Problem leži među ostalim i u tome što velike tvrtke ne ulažu dovoljno truda u to da zaštite svoje potrošače ili idu protiv potrošača (npr. Antivirusi koji koriste loše enkripcije) i na taj način prodaju lažnu sigurnost. Ipak nije sve tako negativno, svi novi pretraživači koriste različite načine kako bi obavijestili korisnika ako se nalazi na nesigurnoj mreži ili ako neka stranica ne koristi certifikat kako bi upozorili korisnika da ne šalje svoje podatke toj stranici ili preko takve veze. Primjer MITM napada trebao bi upozoriti na to koliko je lako doći do podataka ili u ovom slučaju slika ako se nalazimo na istoj mreži kao i žrtva napada. Primjer iz seminara moguće je zaobići korištenjem TLS protokola, kao što vidimo u URL dijelu pretraživača, obaviješteni smo da veza zbog nekog razloga nije sigurna.

[1],[8],[9] Durumeric, Zakir, et al. “The Security Impact of HTTPS Interception.” Network and Distributed Systems Symposium (NDSS’17). 2017.

[2] D. Fisher, „New FREAK Attack Threatens Many SSL Clients“ , 3. ožujak 2015. Dostupno 22. 8. 2018: https://threatpost.com/new-freak-attack-threatens-many-ssl-clients/111390/ .

[3] Filippo Valsorda, „Komodia/Superfish SSL Validation is broken“, 20. veljača 2015. Dostupno 22. 8. 2018: https://blog.filippo.io/komodia-superfish-ssl-validation-is-broken/

[4] „What is HTTPS?“, Dostupno 23.8.2018: https://www.instantssl.com/ssl-certificate-products/https.html

[5] „Deprecating Secure Sockets Layer Version 3.0“, lipanj 2015. Dostupno: 24.8.2018: https://tools.ietf.org/html/rfc7568

[6] T. Polk, K. McKay, S. Chokhani, „Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations“, travanj 2014. Dostupno: 24.8.2018: https://web.archive.org/web/20140508025330/http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r1.pdf

[7] E. Rescorla, „The Transport Layer Security (TLS) Protocol Version 1.3“ kolovoz 2018. Dostupno 24.8.2018: https://tools.ietf.org/html/rfc8446

[11] „Kali linux“, Dostupno 26.8.2018: https://www.kali.org/

[12] „Websploit“, Dostupno: 26.8.2018: https://github.com/websploit/websploit

[13] „Nmap“, Dostupno 26.8.2018: https://www.kali.org/