UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Klemen Hribar

Novi tipi omrežnih napadov in njihovo preprečevanje

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Klemen Hribar

Novi tipi omrežnih napadov in njihovo preprečevanje

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTORICA: doc. dr. Mojca Ciglarič

Ljubljana, 2014

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Klasične tipe omrežnih napadov poznamo že dolgo, vendar pa se z novimi protokoli in novimi načini uporabe porazdeljenih sistemov vsakodnevno pojavljajo nove varnostne ranljivosti in vzporedno tudi novi omrežni napadi. V diplomskem delu najprej kategorizirajte omrežne napade, nato pa opišite izbrane sveže tipe napadov, ki so bili odkriti v zadnjih dveh letih. Razvrstite jih po tipu v ustrezne kategorije in pojasnite njihovo anatomijo; posebej bodite pozorni na napade na SSL/TLS. Enega od njih izberite in ga v testnem okolju reproducirajte. Komentirajte (ne)kompleksnost njegove izvedbe glede na posledice, ki jih lahko povzroči. Pri vseh napadih svetujte primerne ukrepe za njihovo preprečevanje in ne ovrednotite pomen ustrezne varnostne ozaveščenosti uporabnikov.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Klemen Hribar, z vpisno številko **63110097**, sem avtor diplomskega dela z naslovom:

Novi tipi omrežnih napadov in njihovo preprečevanje

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mojce Ciglarič,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 30. junija 2014 Podpis avtorja:

Zahvaljujem se družini za podporo pri izdelavi diplomske naloge in celotnega študija. Zahvaljujem se tudi mentorici za uspešno mentorstvo.

Kazalo

Povzetek

Omrežna varnost je področje, ki ga je treba neprestano spremljati. Pojavljajo se nove oblike omrežnih napadov, ki jih lahko razdelimo v več kategorij. Nove oblike ranljivosti lahko spremljamo na različne načine. Pomembno je, da čim hitreje odkrijemo načine za odpravljanje ranljivosti ali preprečevanje njihove zlorabe. V diplomskem delu smo si za cilj zadali opraviti pregled novejših omrežnih napadov in poiskati ukrepe za njihovo preprečevanje. V ta namen smo izdelali krajši pregled omrežnih napadov. Podrobneje smo analizirali napade za zavrnitev storitve in varnostne ranljivosti, ki so bile v zadnjih letih odkrite v protokolu SSL/TLS. Da bi pokazali, kako lahko je izvesti omrežni napad s sodobnimi orodji, smo izvedli napad z uporabo ranljivosti Heartbleed in pripravili navodila za odpravljanje ranljivosti.

Ključne besede: Omrežni napadi, omrežna varnost, varnostna ranljivost, zavrnitev storitve, Heartbleed, Metasploit, odpravljanje ranljivosti.

Abstract

Network security is an area that needs to be constantly monitored. New forms of network attacks are being discovered and can be divided into several categories. New forms of computer vulnerability can be monitored in various ways. It is important to rapidly discover ways to eliminate vulnerabilities and prevent their abuse. In the thesis, we have set ourselves the aim to conduct a review of recent network attacks and seek measures to prevent them. For this purpose, we have produced a short overview of network attacks. More specifically, we analyzed denial of service attacks and security vulnerabilities which have been discovered in recent years in the SSL / TLS protocol. To demonstrate how easy it is to implement a network attack with modern tools, we carried out an attack using the Heartbleed vulnerability and prepared instructions for eliminating the vulnerability.

Key words: Network attacks, network security, Denial of Service, Heartbleed, vulnerability mitigation.

Poglavje 1 Uvod

Namen dela »Novi tipi omrežnih napadov in njihovo preprečevanje« je opraviti pregled novejših omrežnih napadov in poiskati ukrepe za njihovo preprečevanje. Slovar slovenskega knjižnega jezika opredeljuje izraz napad kot *"nenadno nasilno dejanje, s katerim se hoče kaj pridobiti, doseči ali komu škodovati"* in izraz tip kot *"kar tvorijo izdelki iste vrste zaradi določenih enakih lastnosti, značilnosti, zlasti glede oblike, zgradbe"*.

Računalniki so vse bolj prisotni v naših življenjih. Skoraj vsak prenosni telefon je postal že osebni računalnik. Na svetovnem spletu je dostopnih vedno več osebnih podatkov, ki jih posredujemo sami ali pa se beležijo samodejno. Varnost na spletu zato dobiva vse večji pomen. Zaradi tega je potrebno neprestano spremljati novo odkrite varnostne grožnje in možne načine za njihovo preprečevanje. V Sloveniji je to naloga nacionalnega odzivnega centra za obravnavo incidentov s področja varnosti elektronskih omrežij in informacij (SI-CERT), ki opravlja koordinacijo razreševanja nepredvidenih dogodkov, tehnično svetovanje ob vdorih, računalniških okužbah in drugih zlorabah ter izdaja opozorila za upravitelje omrežij in širšo javnost o trenutnih grožnjah na elektronskih omrežjih. Pri izdelavi diplomske naloge smo si pomagali tudi z njihovimi usmeritvami in priporočili.

V zadnjih letih se na področju informacijske varnosti, poleg okužb z zlonamerno kodo, pogosto pojavljata dve vrsti napadov: porazdeljeni napadi za zavrnitev storitve in napadi na zaščito s protokolom SSL/TLS. Porazdeljeni napadi za zavrnitev storitve so pogosto učinkoviti, napadalca pa je težko izslediti. Pregledali bomo načine za njihovo izvedbo. V letih 2013 in 2014 sta največ škode povzročili predvsem dve kategoriji porazdeljenih napadov za zavrnitev storitve. To so napadi z odbojem preko strežnikov NTP in napadi z DNS odbojem. Te napade bomo podrobneje analizirali in poskušali poiskati načine za njihovo preprečevanje.

Protokol SSL/TLS skrbi za varnost spletnih komunikacij, nižji protokoli pogosto ne vsebujejo lastnih zaščitnih mehanizmov, zato jih ovijemo v sejo SSL/TLS. V želji, da bi odkrili varnostne ranljivosti v protokolu ali njegovi implementaciji so strokovnjaki poskušali zaobiti varnostne mehanizme na več načinov. Nekaj večjih ranljivosti je bilo odkritih v preteklih treh letih. Analizirali bomo njihovo delovanje in načine za odpravljanje.

Eno izmed večjih ranljivosti v implementaciji protokola, ki je bila odkrita pred kratkim, so poimenovali Heartbleed. Pripravili bomo okolje in prikazali primer napada z uporabo te ranljivosti. Poskusili bomo prebrati podatke, ki jih protokol ščiti. Poiskali bomo kako se ranljivost odpravlja in pripravili navodila za odpravljanje ranljivosti na paketu strežnikov XAMPP.

Poglavje 2 Pregled in razdelitev omrežnih napadov

2.1 Različni načini kategorizacije

Omrežne napade lahko razdelimo na več načinov. Možnosti za kategorizacijo je veliko, zato bodo predstavljene najbolj pogoste. Napade lahko razdelimo v grobem na: aktivne ali pasivne, vrsto motiva, napako, ki jo izkoriščajo ali tip omrežja, ki ga napadajo. Tehnične napade lahko razdelimo tudi po plasteh v modelu ISO/OSI glede na protokol v katerem izkoriščajo ranljivost oziroma na aplikacijsko plast, če je ranljivost v programski kodi. Lahko pa jih delimo tudi na tehnične napade ter prevare in goljufije, kot jih obravnava SI-CERT.

2.2 Osnovne kategorije omrežnih napadov

Osnovne kategorije omrežnih napadov so velike družine napadov, ki so sorodni glede na način njihove izvedbe.To so: napadi s pregledovanjem (ang. Scanning), napadi s ponarejanjem (ang. Spoofing), napadi z odjemanjem prometa (ang. Network sniffing), napadi za zavrnitev storitve (ang. DoS), napadi s posrednikom (ang. Man in the middle), XSS[1] napadi, Ribarjenje (ang. Phishing) in zlonamerni programi. Vsaka izmed naštetih kategorij je obširno poglavje zase, zato je namen tega poglavja predstaviti kratek pregled osnovnih kategorij za boljše razumevanje načinov kategorizacije in njihove povezave z novimi tipi napadov.

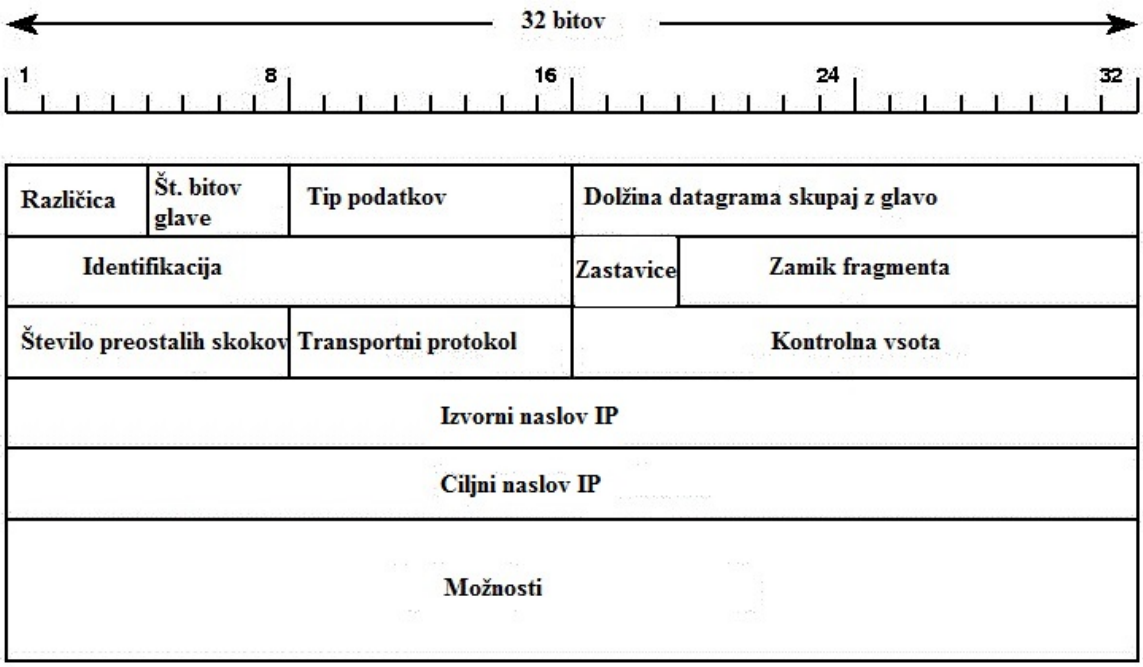
2.2.1 Napadi s pregledovanjem vrat

Preden lahko napadalec izkoristi ranljivost v aplikaciji mora dobro poznati sistem. Kadar fizični dostop do naprav ni možen, se v ta namen uporablja pregledovanje vrat (ang. Port scanning). Pregledovanje vrat je podoben pristop kot ga nepridipravi uporabljajo za vlom v stavbe. Najprej si jih dobro ogledajo in preverijo okna in vrata, če so odprta oziroma odklenjena. Obstaja več načinov pregledovanja odprtih vrat in več orodij za izvedbo pregledovanja. Možna pa je tudi izvedba brez uporabe orodja, s kodiranjem surovih paketkov. Tako kodiranje mora biti opravljeno v nizkonivojskem jeziku, ki omogoča neposreden nadzor mrežne kartice. Zelo razširjeno orodje za pregledovanje je Nmap 4.3.2.

[1] XSS (ang. cross-site scripting), napad na spletno stran z vrinjenjem zlonamerne kode, napisane v skriptnem jeziku.

2.2.2 Napadi s ponarejanjem

Med napade s ponarejanjem štejemo vse primere zlonamernega pošiljanja ponarejenih IP, naslovov MAC, paketkov TCP in sporočil ARP. Protokol ARP služi za povezavo IP in naslova MAC. Protokol ARP služi za povezavo IP in naslova MAC. Pare IP in naslovov MAC naprav na lokalnem omrežju naprava hrani v tabeli ARP. Kadar za posamezen naslov IP naprava nima shranjenega naslova MAC, pošlje ARP zahtevek vsem napravam na omrežju. Naprava, ki ima iskani naslov IP, odgovori s sporočilom unicast, ki vsebuje njen naslov MAC napravi, ki je poslala poizvedbo. Napadalec lahko to izkoristi in na tak način poveže napravo, ki je poslala zahtevo in svojo napravo. Tako doseže, da promet tarče teče preko napadalčeve naprave. Tak napad je običajno uvod v druge napade. Napadalec lahko povezavo izkoristi za prisluškovanje prometa, ki se prenaša (Sniffing), lahko promet spreminja in posreduje naprej oziroma izvede napad s posrednikom ali pa prepreči določenemu prometu, da se posreduje do tarče in tako izvede primer zavrnitve storitve. Za preprečevanje ponarejanja ARP se uporabljajo programi, ki nadzirajo sporočila ARP in obvestijo omrežnega administratorja, kadar so zaznani določeni parametri. Primer takega programa je Arpwatch. Pogosto se uporablja tudi ponarejanje naslovov datagramov . [1] Ker protokol IP ne omogoča preverjanja izvornega in ponornega naslova IP, lahko napadalec zamenja izvorni naslov in se pretvarja, da je nekdo drug ali pa povzroči, da naprave, ki jim pošilja promet odgovorijo na ponarejeni naslov IP. Na ta način lahko napadalec izvede prestrezanje prometa, ukrade sejo TCP ali pa povzroči zavrnitev storitve. Osnovno zaščito pred zlonamernimi ponarejenimi paketki nudi požarni zid, ki preprečuje prehod paketkov, ki prihajajo zunaj lokalnega omrežja in vsebujejo izvorni naslov IP, ki se nahaja znotraj omrežja, saj s tem poizkušajo posnemati lokalno napravo. Protokoli na višjih plasteh vsebujejo lastne zaščite za odkrivanje ponarejenih sporočil. Protokol TCP vsebuje polje zaporedna številka, ki se jo uporablja za sledenje toku sporočil. V primeru, da napadalec poskuša ugrabiti sejo, mora predvideti še zaporedne številke v datagramih TCP, v nasprotnem primeru bodo paketki na strani tarče zavrnjeni.



Slika 1: Format IPv4 datagrama.

2.2.3 Napadi z odjemanjem prometa

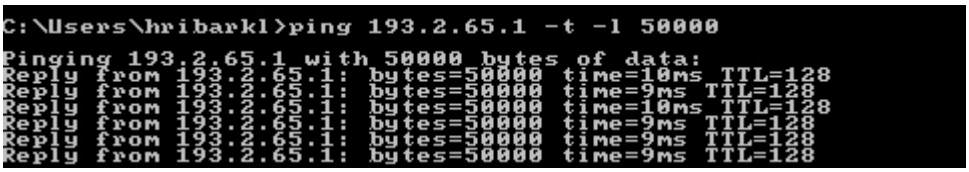
Napade z odjemanjem prometa oziroma podatkov lahko razdelimo na vse plasti referenčnega modela OSI. Njihova delitev je prikazana v kategorizaciji po plasteh v referenčnem modelu OSI v poglavju 2.6.2. Odjemanje prometa napadalec izvede, da si pridobi zaupne podatke kot so uporabniška imena in gesla, lahko pa poizkuša pridobiti podatke na nižjih nivojih, kadar ni mogoče neposredno prebrati podatkov za avtentikacijo. Takrat lahko poizkuša prebrati podatke za krajo seje SSL, seje TCP, lahko poizkuša prebrati IP tarče in vrata storitve ali pa naslov MAC iz sporočil ARP. Odjemanje prometa se lahko izvaja na lokalnem omrežju z mrežno kartico, v načinu za zajemanje vsega prometa (ang. promiscuous mode), lahko pa kot del napada s posrednikom ali kot del napada s ponarejanjem, v primeru, ko se napadalec pretvarja, da je pravi sogovornik tarče. Pred takimi napadi se lahko zaščitimo z uporabo varnega protokola HTTP, ki uporablja SSL/TLS, da kriptira sporočila in z uporabo standarda IPSec za kriptiranje povezave na omrežnem nivoju.

2.2.4 Napadi za zavrnitev storitve

Prav tako kot napade z odjemanjem prometa lahko tudi napade za zavrnitev storitve razdelimo po vseh OSI plasteh. Njihov namen je tarči preprečiti dostop do spleta ali pa do storitve na spletu. V grobem jih lahko razdelimo na dve večji kategoriji in sicer DoS, ki prihajajo iz enega vira in DDoS, pri katerih dve ali več naprav generirajo zlonamerni promet. Osnovni primeri napada vključujejo poplave paketkov ICMP, poplave paketkov SYN s katerimi napadalec izčrpa vire tarče. Naprednejša oblika pa so DDoS ali porazdeljeni napadi z zavrnitvijo storitve in vključujejo uporabo mreže računalnikov, preko katere napadalec ojači promet. V zadnjem času postajajo napadi vse bolj dovršeni. Uporabljajo jih različne skupine aktivistov za izražanje protesta in so zaradi tega tudi medijsko odmevni.

Preprost primer poplave paketkov ICMP lahko prikažemo v ukazni vrstici operacijskega sistema Windows. Za pošiljanje paketkov ICMP nam je na voljo ukaz "ping". Da bo ukaz pošiljal promet brez prekinitve dodamo stikalo "-t" s stikalom "-l" pa določimo velikost paketka. Največja dovoljena velikost je 65500. Tako lahko poženemo ukaz:

"ping ip_naslov_tarče –t –l 65500"



Slika 2: Primer poplave ICMP z uporabo ukaza "ping" v ukaznem pozivu

Vendar pa je tak napad neuspešen, saj povzroči zanemarljivo količino prometa, poleg tega pa je viden napadalčev naslov IP. Napad pa bi lahko bil uspešen, če bi na isti naslov naenkrat paketke pošiljalo več računalnikov. V tem primeru govorimo o porazdeljenem napadu za zavrnitev storitve.

2.2.5 Porazdeljeni napadi za zavrnitev storitve

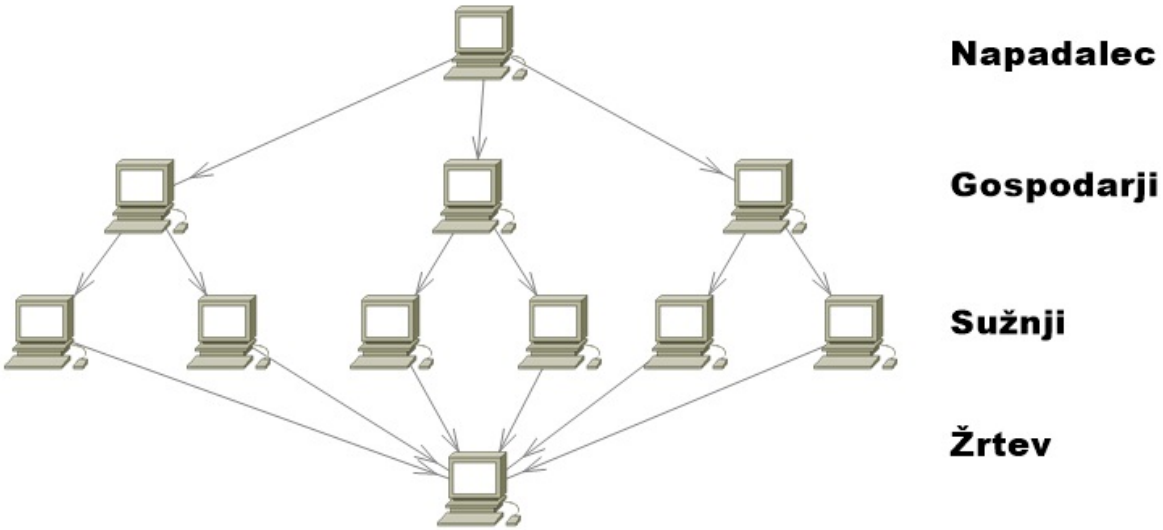
Za porazdeljene napade za zavrnitev storitve velja, da dve ali več naprav pošiljajo zlonameren promet, ki povzroči, da tarča izgubi dostop do spleta oziroma do dela spleta. V letu 2014 se zgodi povprečno 28 takih napadov vsako uro [2]. Taki napadi so običajno usmerjeni proti pomembnejšim spletnim strežnikom, bankam, plačilnim storitvam in celo korenskim domenskim strežnikom. US-CERT [3] kot simptome in posledice takih napadov navaja počasno delovanje spleta, nedostopnost določene spletne strani ali nedostopnost celotnega spleta, nenavadno povečanje neželene elektronske pošte.

Napadi želijo doseči enega od sledečih ciljev:

- Porabo računalniških virov kot so pasovna širina, pomnilnik, prostor na disku ali procesorski čas
- Poškodovanje nastavitev, kot so nastavitve za usmerjanje
- Spreminjanje informacij o stanju povezave, na primer nezaželena ponastavitev TCP sej
- Motenje fizičnih komponent omrežja
- Motenje komunikacijskega medija med uporabniki in tarčo

Za doseganje teh ciljev napadalec uporabi ojačitev prometa. Da bi popolnoma ohromil sistem mora napadalec ustvariti zelo veliko količino prometa, za kar potrebuje veliko naprav. Ojačitev lahko doseže z uporabo omrežja robotskih računalnikov (ang. Botnet) ali pa z odbojem prometa preko strežnikov. Pri uporabi omrežja robotskih računalnikov si napadalec največkrat pomaga z zlonamerno programsko kodo, s katero okuži računalnike. Računalniki nato pošiljajo zlonameren promet po napadalčevih ukazih. Pri uporabi odboja za ojačitev pa napadalec pošlje zahtevek velikemu številu strežnikov. V zahtevek kot izvorni naslov IP vstavi žrtvin naslov IP. Tako strežniki odgovorijo žrtvi in jo posledično poplavijo s prometom.

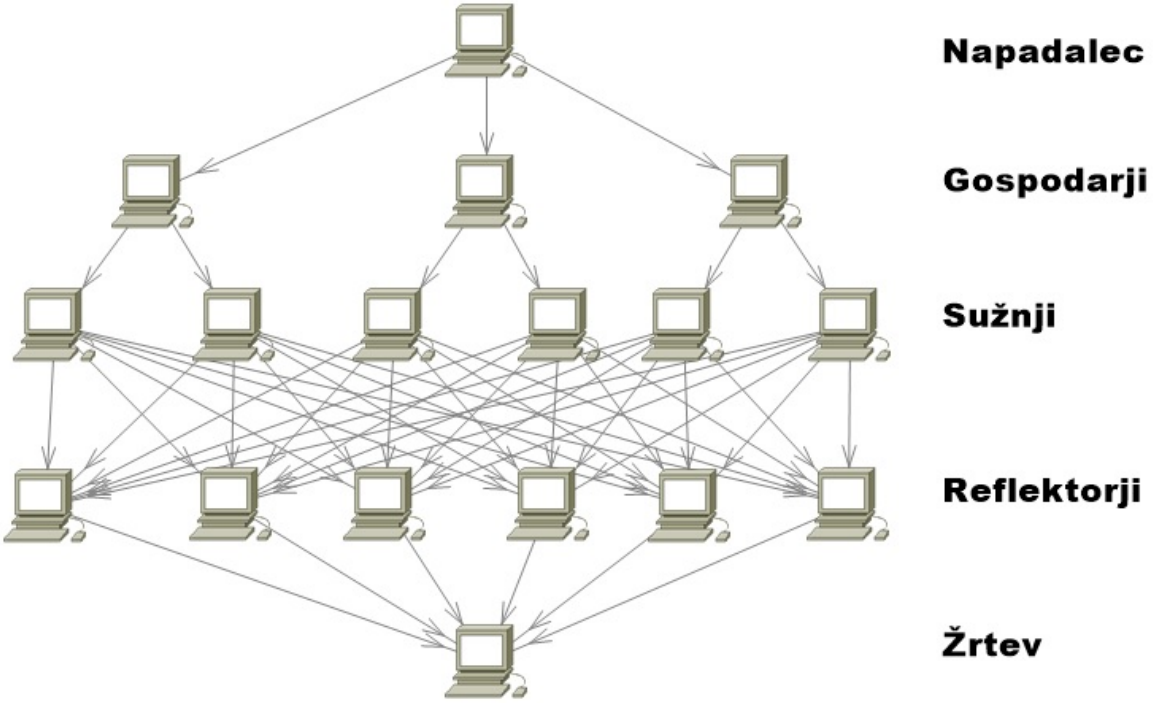
Napade razdelimo v dve večji kategoriji, to so neposredni napadi in napadi z odbojem oziroma DRDoS[2]. Da bi bil napad bolj uspešen, oba načina potrebujeta mrežo za ojačitev. V vsakem primeru tako sodelujejo vsaj tri strani, napadalec, žrtev in mreža za ojačitev. Mrežo za ojačitev sestavljajo tako imenovani "zombi" računalniki, ki so okuženi z zlonamerno kodo. Zombiji se delijo na gospodarje in sužnje. Slika 3 prikazuje tok podatkov od napadalca do žrtve. Zlonamerna koda na računalnikih gospodarjih miruje in čaka na ukaz napadalca, računalniki sužnji pa čakajo na ukaze gospodarjev. Ko napadalec pošlje ukaz računalnikom gospodarjem, se na njih izvedejo procesi, ki aktivirajo sužnje, da začnejo pošiljati velike količine paketkov in tako poplavijo žrtev z neuporabnimi zahtevami ter posledično izčrpajo njene vire.



Slika 3: Skica arhitekture neposrednega porazdeljenega napada za zavrnitev storitve.

[2] DRDoS(ang. Distributed Reflection Denial of Service) porazdeljeni napadi za zavrnitev storitve z odbojem.

Pri napadih z odbojem so v mrežo za ojačitev poleg gospodarjev in sužnjev vključeni še računalniki za odboj oziroma reflektorji. Slika 4 prikazuje tok podatkov od napadalca do žrtve preko reflektorjev. Napadalec ima nadzor nad gospodarji, ki naprej nadzirajo sužnje. Glavna razlika je, da sužnji pošljejo reflektorjem tok paketkov s ponarejenim izvornim naslovom IP, ki pripada žrtvi, ti pa jo nato zasujejo z odgovori. Tako pri napadih z odbojem sodelujejo računalniki, ki niso okuženi in težko zaznajo, da so del napada. Tako je napad še bolj porazdeljen in težje izsledljiv, poleg tega pa povzroči tudi večji tok podatkov, ker je udeleženih več naprav.



Slika 4 Skica arhitekture neposrednega porazdeljenega napada za zavrnitev storitve z odbojem.

Seznam vsebuje nekaj bolj znanih primerov napadov. Čeprav so v večini primerov odpravljeni, so možne nove različice na podlagi koncepta napada.

- Apache2: Napad je usmerjen proti spletnemu strežniku Apache. Odjemalec pošlje zahtevo za storitev, v kateri je vsebovanih mnogo glav http. Strežnik prejme tako veliko število zahtev, da jih ne more obdelati pravočasno in se sesuje.

- Back: Tudi ta napad je usmerjen proti spletnemu strežniku Apache. Napadalec pošlje zahtevke, ki vsebujejo veliko število poševnic (/) v naslovu URL. Ker strežnik poizkuša obdelati vse te zahtevke, ne odgovarja na zahtevke drugih uporabnikov in tako postane neodziven.

- Ping of Death: Napadalec ustvari paket, ki vsebuje več kot 65536 bytov, ki je največ kar dovoli protokol IP. Ker se na napravi, ki ga prejme sproži sistemska napaka, lahko napad resetira sistem. Vendar pa je napad dobro poznan in odpravljen na vseh novejših operacijskih sistemih.

- Smurf Attack: Napad temelji na zlorabi protokola ICMP. Napadalec pošlje veliko število paketkov ICMP "echo-request", ki vsebujejo žrtvin naslov IP v izvoru, na broadcast naslov[3]. Naprave se odzovejo s paketki "echo-reply" in zasujejo žrtev z odzivi.

- SSH Process Table: Napadalec vzpostavi veliko število povezav z žrtvijo preko protokola SSH, ne da bi dokončal proces prijave. Tako povzroči, da proces, ki sprejme zahtevek, sproži tako veliko število procesov SSH, da žrtvi izčrpa vire.

- TCP Reset: Napadalec pregleduje omrežje in beleži zahtevke za povezavo TCP . Takoj, ko zazna žrtvin zahtevek, ji odgovori s paketkom TCP RESET, ki povzroči, da se povezava zapre.

- Teardrop: Medtem, ko je paketek na poti med pošiljateljem in prejemnikom, se lahko razdrobi na več fragmentov. Napadalec ustvari tok fragmentov IP, ki vsebujejo prevelike ali premajhne vrednosti za zamik. Napad je povzročil sesutje operacijskih sistemov Windows 3.1, 95 in NT ter Linux pred različico 2.0.32 zaradi napake v kodi za ponovno sestavljanje paketkov.

[3] Broadcast naslov je naslov, ki naslovi vse naprave v omrežju

2.2.6 Napadi s posrednikom

Za napad s posrednikom (ang. Man in the middle) je značilno, da napadalec promet tudi spreminja ali drugače vpliva nanj in je posledica prej opisanih napadov s ponarejanjem podatkov in z odjemanjem prometa. Nevarnost takega napada je, da tarča pogosto ne zazna, da je napad v teku. Cilj napada je kraja seje in tako podatkov, ki se prenašajo. Obstaja več metod za izvedbo napada. Pogosta je uporaba zastrupljanja tabele ARP. Napadalec pošlje ponarejen odgovor ARP tako, da poveže svoj naslov MAC in žrtvin naslov IP. Tako se promet pošilja njemu, on pa modificiran promet posreduje tarči. Lahko pa poizkuša svoj naslov MAC predstaviti kot privzeti prehod omrežja. Na tak način vse naprave posredujejo promet preko napadalca. Tak napad lahko izvede na lokalnem omrežju. Podoben napad lahko napadalec izvede tudi na spletu s pomočjo lažnega strežnika DNS. Tarči spremeni naslov strežnika DNS tako, da zahteve za reševanje v naslov IP preusmeri na svoj strežnik. Napadalčev strežnik nato žrtev preusmeri na lažno spletno banko ali drugo storitev misleč, da se nahaja na avtentični strani.

2.2.7 Napadi z vrinjenjem skriptne kode

Manipulacija kode je najbolj pogosta ranljivost v spletnih tehnologijah. Ponavadi je posledica nepravilnega programiranja ali slabih programerskih praks. Napadi XSS (ang. Cross-site scripting attacks) so še vedno možni na velikem številu spletnih strani kljub temu, da je problem že dobro poznan. Ranljivost omogoča, da lahko odjemalec pošlje del izvršljive kode na strežnik ali pa spremeni del kode, ki se nahaja na strežniku. Tehniko si lahko ogledamo na sledečem primeru: [4]

index.php:

```
<?php

$name = $_GET['name'];

echo "Pozdravljeni $name<br>";

echo "<a href='http://primerXSSnapada.com/'>Pritisni za prenos</a>";

?>
```

Napadalec lahko poskusi spremeniti ciljni URL[4] v povezavi "Pritisni za prenos". Uporabnika preusmeri na zlonamerno spletno stran.

```
index.php?name=<script>window.onload = function() {var
link=document.getElementsByTagName("a");link[0].href="http://not-real-
xssattackexamples.com/";}</script>
```

Pogosto napadalec še pretvori znake ASCII v heksadecimalno obliko. Zgornja povezava bi nato izgledala tako:

```
index.php?name=%3c%73%63%72%69%70%74%3e%77%69%6e%64%6f%77%2e%6f%6e%6c%6f%61%64
%20%3d%20%66%75%6e%63%74%69%6f%6e%28%29%20%7b%76%61%72%20%6c%69%6e%6b%3d%64%6f
%63%75%6d%65%6e%74%2e%67%65%74%45%6c%65%6d%65%6e%74%73%42%79%54%61%67%4e%61%6d
%65%28%22%61%22%29%3b%6c%69%6e%6b%5b%30%5d%2e%68%72%65%66%3d%22%68%74%74%70%3a
%2f%2f%61%74%74%61%63%6b%65%72%2d%73%69%74%65%2e%63%6f%6d%2f%22%3b%7d%3c%2f%73
%63%72%69%70%74%3e
```

To stori predvsem zato, da je URL ljudem neberljiv in tako lažje skrije zlonamerno kodo v povezavo.

2.2.8 Vrivanje kode SQL

Vrivanje SQL je tehnika, ki jo zlonamerni uporabniki izkoriščajo za vrivanje svojih ukazov SQL v stavek SQL, preko obrazca za vnos na spletni strani. Tako lahko spremenijo stavek SQL in pridobijo podatke do katerih jim sicer dostop ni omogočen, zbrišejo cele tabele ali pa celo vstavijo v bazo svoje podatke, ki se nato izpišejo ostalim uporabnikom. Preprost primer vrivanja kode lahko pokažemo na stavku, ki vrne uporabnika:

```
SELECT * FROM Users WHERE UserId = 105
```

Napadalec bi lahko v stavek dodal še dodaten pogoj, ki je vedno resničen na primer "1=1". Tako stavek postane:

```
SELECT * FROM Users WHERE UserId = 105 or 1=1
```

[4] URL (ang. Unified Resource Locator) je naslov spletnih strain v svetovnem spletu.

Ker za delovanje prvotne poizvedbe uporabnik potrebuje dostop do tabele Users in do stolpca UserId, bi bila takšna poizvedba veljavna, strežnik bi napadalcu posredoval podatke o vseh uporabnikih v tabeli. Napadalec lahko svojo kodo vrine preko parametrov v URL, preko vnosnih polj v obrazcu spletne strani ali pa celo z manipulacijo kode na odjemalčevi strani.

2.2.9 Ribarjenje

Ribarjenje (ang. Phishing) je tehnika, ki se uporablja za pridobivanje zaupnih podatkov z uporabo ponarejenih spletnih strani ali samo spletnih obrazcev. Napadalec lahko preusmeri uporabnika na spletno stran, ki je na videz enaka pravi spletni strani. To lahko naredi s posredovanjem lažnega spletnega naslova v elektronski pošti, ki uporabnika preusmeri na napadalčevo spletno stran. Kadar to stori preko posredovane povezave, je tak napad redko uspešen, uporabnik namreč hitro opazi, da se naslov lažne strani ne ujema s pravim spletnim naslovom. Zato je ribarjenje pogosto uporabljeno skupaj s prej omenjenimi napadi. Napadalec lahko preusmeri promet na svoj strežnik z zastrupljanjem DNS, ponarejanjem naslova IP ali pa kot del napada s posrednikom. Ribarjenje uporabi, ker je promet pogosto kriptiran in je dekripcija prometa, ki ga sicer odjema redko uspešna ali pa skoraj nemogoča. Zaradi kraje podatkov za prijavo, se uporabljajo še dodatni mehanizmi za preverjanje avtentičnosti uporabnika, kot na primer digitalna potrdila, ki jih napadalec ne more ukrasti na tak način. Napadi z ribarjenjem so pogosto tudi uspešni.

2.2.10 Zlonamerni programi

Med zlonamerne programe štejemo računalniške črve, viruse, trojanske konje, izsiljevalske programe in vohunske programe. Začetki raziskovanja programov, ki se lahko sami razmnožujejo, segajo v leto 1949, ko jih je na univerzi v Illinoisu preučeval John von Neumann. Prvi virus za osebne računalnike se imenuje Brain. Leta 1986 sta ga napisala brata Farooq Alvi iz Pakistana, da bi preprečila piratsko kopiranje njune programske opreme. Ker v tistem času osebni računalniki še niso bili povezani v splet, se je virus širil preko disket. Danes se širijo večinoma prek spleta in USB pomnilniških naprav. Zlonamerni programi postajajo vse bolj zapleteni in vse težji za odkrivanje. Njihov namen je kraja podatkov, poškodovanje programske opreme, izsiljevanje, vohunjenje ali prisilno oglaševanje. Osnovno zaščito pred njimi nam nudijo: dober protivirusni program, požarni zid, varen operacijski sistem in posodobljena programska oprema. Pred okužbo se najlažje zaščitimo tako, da ne prenašamo programske opreme iz nezanesljivih virov.

2.3 Aktivni in pasivni napadi

Pri razdelitvi na aktivne in pasivne, štejemo napade za aktivne, kadar ti spreminjajo podatke ali izvajajo ukaze na sistemu in pasivne, kadar ne spreminjajo delovanja sistema ali njegovih podatkov [5]. Tako jih lahko razdelimo na:

- Pasivni
    - Prisluškovanje liniji (ang. Wiretaping)
    - Skeniranje vrat
- Aktivni
    - Napadi z zavrnitvijo storitve (ang. Denial of service)
    - Napadi s ponarejanjem (ang. Spoofing)
    - Napadi s posrednikom (ang. Man in the middle)
    - Zastrupljanje ARP
    - Vrivanje SQL kode
    - Zlonamerni programi
    - Napadi na gesla

2.4 Razdelitev na tehnične napade ter goljufije in prevare

Slovenski nacionalni odzivni center za obravnavo incidentov s področja varnosti elektronskih omrežij in informacij (SI-CERT) v svojih letnih poročilih o omrežni varnosti razdeli obravnavane incidente glede na to ali so tehnični napadi ali pa izkoriščajo človeški faktor. Če izkoriščajo človeški faktor, potem jih uvrščajo v kategorijo "Goljufije in prevare". [6]

- Tehnični napadi
    - Skeniranje in poskušanje
    - Botnet
    - Zavrnitev storitve(DDoS)
    - Škodljiva koda
    - Zloraba storitve
    - Vdor v sistem
    - Zloraba uporabniškega računa
    - Razobličenje (ang. Defacement)
    - Napad na aplikacijo
- Goljufije in prevare
    - Kraja identitete
    - Goljufija
    - Spam
    - Phising
    - Dialler

2.5 Razdelitev glede na motiv napada

Motivov za napad je veliko, kljub temu pa lahko večino napadov razvrstimo glede na pogoste motive za izvedbo napada. Med glavne motive lahko štejemo: zbiranje podatkov, povzročanje škode, finančna korist, izsiljevanje, izkazovanje znanja ali za zabavo.

- Zbiranje podatkov
    - Skeniranje vrat
    - Odjemanje prometa (ang. Sniffing)
    - Napadi s posrednikom (ang. Man in the middle)
    - Beleženje pritiskov (ang. Keylogging)
    - Vdori v baze podatkov
- Povzročanje škode
    - Zavrnitev storitve (DDoS)
    - Brisanje podatkov
    - Poškodovanje programske opreme
    - Poškodovanje strojne opreme
- Finančna korist
    - Napadi na avtentikacijo
    - Napadi na spletno bančništvo
    - Kraja kriptovalut
    - Napadi na borze

- Izsiljevanje
    - Virusi tipa ransomware
    - Kraja občutljivih podatkov
- Prikaz znanja ali zabava
    - Razkrivanje varnostnih lukenj
    - Potegavščine

2.6 Razdelitev glede na plast v referenčnem modelu OSI

2.6.1 Referenčni model ISO OSI

OSI je kratica za model Open Systems Interconnection organizacije ISO. To je referenčni model, ki predstavlja modulirano zgradbo protokolov. [7] Protokoli so razdeljeni na sedem abstraktnih plasti. Aplikacijska plast služi kot vmesnik med uporabnikom in programsko opremo. Tukaj so definirani protokoli za elektronsko pošto, svetovni splet, za prenašanje datotek in drugi. Predstavitvena plast zagotavlja različne načine kodiranja in sisteme pretvorb za aplikacijsko plast. Pretvarja podatke, določa sintakso in formiranje podatkov. Sejna plast nadzira komunikacijo med računalniki. Določa vrsto komunikacije, jo vzpostavlja in prekinja. Transportna plast definira način prenosa in razbija sporočila na manjše dele. Omrežna plast izbira pot in skrbi za usmerjanje in preklapljanje paketkov ali zvez. Povezavna plast nudi zanesljivo povezavo med dvema neposredno povezanima vozliščema in odpravlja napake, ki lahko nastanejo na fizični plasti. Fizična plast definira prenosni medij, nivo signala, hitrost prenosa in način zapisa podatkov. Model je bil razvit leta 1984 s strani Mednarodne organizacije za standardizacijo (ang. International Organization for Standardization).

OSI Plast	**Podatkovna enota**	**Protokoli**
Aplikacijska plast	Sporočilo/podatki	DNS, FTP, HTTP, IMAP, IRC, NNTP, POP3, SIP, SMTP, SSH[5]
Predstavitvena plast	Sporočilo/podatki	MIME, XDR
Sejna plast	Sporočilo/podatki	NetBIOS, SAP, RTP, SOCKS SPDY
Prenosna plast	Segment(TCP) ali Datagram(UDP[6])	DCCP, TCP, UDP, SCTP, RTP
Omrežna plast	Paket (datagram IP)	IPv4, IPv6, ICMP, IGMP,
Povezavna plast	Okvir	ARP(med plastema), ECP, ATM, DDCMP, BSC, LAPB, LAPD
Fizična plast	Biti/simboli	Token ring, Ethernet, FDDI, PPP, Wi-Fi.

Tabela 1: OSI model.

[5] Secure Shell Protokol
[6] User Datagram Protocol

Tako lahko posamezno kategorijo napadov naprej razdelimo glede na to, v katerem protokolu izrablja ranljivost. Na tak način pa ne moremo razvrstiti napadov, ki izkoriščajo človeški faktor, viruse in napake pri kodiranju.

2.6.2 Kategorizacija napadov z odjemanjem prometa po OSI plasteh

Napadi z prisluškovanjem prometu z omrežja (ang. Network sniffing) so družina napadovizvedenih z namenom, da prestrežejo in preberejo promet. Lahko jih razvrstimo tudi po plasteh v OSI referenčnem modelu glede na protokol, ki ga uporabljajo. [8]

OSI plast	**Tehnika za napad**
Aplikacijska plast	Odjemanje uporabniškega imena in gesla
Predstavitvena plast	Kraja SSL/TLS seje
Sejna plast	Odjemanje FTP, Telnet prometa
Prenosna plast	Kraja TCP seje, odjemanje UDP prometa
Omrežna plast	Odjemanje IP ali vrat
Povezavna plast	Odjemanje naslovov MAC in ARP
Fizična plast	Prisluškovanje mediju

Tabela 2: Razvrstitev napadov z odjemanjem po plasteh OSI.

2.6.3 Kategorizacija napadov za zavrnitev storitve po plasteh OSI

Tudi napade za zavrnitev storitve lahko razvrstimo po plasteh referenčnega modela OSI. Prikazana razvrstitev je povzeta po poročilu, ki so ga pripravili na US-CERT. [9]

OSI plast	**Tehnika za napad**
Aplikacijska	Zahtevki HTTP GET in POST, Vsiljena elektronska pošta
Predstavitvena plast	Ponarejeni zahtevki SSL
Sejna plast	DDoS z uporabo protokola Telnet
Prenosna plast	Poplava z uporabo paketkov SYN, napad Smurf
Omrežna plast	Poplava z uporabo paketkov ICMP
Povezavna plast	ARP-nevihta
Fizična plast	Fizična prekinitev ali motenje prenosnega medija

Tabela 3: Razvrstitev napadov za zavrnitev storitve po plasteh OSI.

Poglavje 3 Pomembne odkrite varnostne ranljivosti v letih 2012-2014

3.1 Organizacije ki spremljajo in obravnavajo nove odkrite ranljivosti

3.1.1 Nacionalni odzivni centri

Prvi CERT je bil ustanovljen leta 1988 v ZDA kot odgovor na prvi večji internetni incident – širjenje prvega črva, kasneje imenovanega kar "The Internet Worm" (več v A Tour of the Worm in RFC 1135). S širitvijo Interneta po svetu, so se postopoma začele podobne organizacije in servisi pojavljati tudi izven ZDA, prvotni CERT pa se je preimenoval v CERT Coordination Center (CERT/CC). SI-CERT je bil ustanovljen leta 1995. [10] SI-CERT je tudi član svetovnega združenja odzivnih centrov (FIRST Forum of Incident Response and Security Teams). V združenju FIRST je 305 ekip iz 66 držav. Nacionalni odzivni centri sprejemajo prijave o zlorabah in se ukvarjajo z njihovo analizo in preprečevanjem. Sodelujejo tudi s policijskimi oddelki za informacijsko varnost.

3.1.2 Podjetja, ki se ukvarjajo z informacijsko varnostjo

Mnoga zasebna podjetja so se začela ukvarjati z izdelavo protivirusnih programov, lastnih požarnih zidov, opreme za virtualno poganjanje sumljive programske opreme in orodij za varno uničenje podatkov. Podjetja kot so Avira, F-Secure, Kaspersky, McAfee, Panda Security, Symantec in Sophos se ukvarjajo s produkcijo programske opreme za informacijsko varnost. Poleg opreme za zasebne uporabnike razvijajo tudi rešitve za poslovne uporabnike in organizacije. Poleg protivirusnih programov in požarnih zidov nudijo tudi opremo za virtualizacijo strežnikov, varno komunikacijo, varno shranjevanje podatkov, varna virtualna zasebna omrežja, strežniško zaščito in protivirusne rešitve za mobilne naprave. Podjetja hranijo lastne baze odkritih varnostnih groženj, ki jih uporabljajo za posodabljanje svoje opreme in nadgradnjo baze zlonamernih programov na strani uporabnika.

3.1.3 Organizacije za standardizacijo

Organizacije kot so IETF[7], IRTF[8], IANA[9] se ukvarjajo z razvojem spletnih standardov in protokolov, pri svojem delu upoštevajo tudi varnostne zahteve novo razvitih rešitev. Razlog za mnoge odkrite ranljivosti leži v tem, da v začetku razvoja omrežij, varnosti niso pripisovali velikega pomena. Omrežja so bila prvotno razvita kot zaprti sistemi za izmenjevanje informacij v vojski in v akademskih krogih. Cilj protokolnega sklada TCP/IP je predvsem zanesljiv prenos podatkov. Zato IPv4 nima vgrajenih varnostnih mehanizmov. Nadgrajen standard IPv6 vsebuje dodatne varnostne mehanizme. Sam naslovni prostor, ki vsebuje 128 bitov, že zaradi svoje velikosti nekoliko otežuje napade s pregledovanjem. IPv6 ima fiksno glavo, s tem zmanjšuje uporabo fragmentacije. Nudi polno podporo standardu IPsec, ki doda dva protokola AH (authentication header) in ESP (Encapsulating Security Payload), ki nudita avtentikacijo, integriteto, zaupnost in nadzor dostopa. Prav tako nadomesti protokol ARP z protokolom ND (Neighbour Discovery), ki odpravlja nekatere pomanjkljivosti ARP-a. Kljub temu, da so razviti novi protokoli, pa nekatere varnostne ranljivosti ostajajo. Popolnoma varnih protokolov ni mogoče razviti, saj vseh varnostnih ranljivosti med razvojem ni mogoče predvideti. Zato je potrebna stalna analiza protokolov in njihovo nadgrajevanje. Če protokol vsebuje veliko mehanizmov za varnost, se poveča tudi njegova kompleksnost. Aplikacije, ki implementirajo tak protokol pa postanejo manj prijazne do uporabnikov.

3.1.4 Proizvajalci programske opreme

Najpogosteje različne varnostne popravke izdajajo proizvajalci programske opreme, saj imajo poln dostop do izvorne kode in usposobljene strokovnjake za implementacijo popravkov. Nacionalni odzivni centri sprejmejo obvestilo o zlorabi in analizirajo, zaradi česa je mogoča. Nato o tem obvestijo programsko hišo, ki izdeluje opremo v kateri je zaznana ranljivost. Včasih pa ranljivost odkrijejo proizvajalci protivirusne opreme, preden jo odkrijejo uporabniki ali nacionalni odzivni centri, saj imajo najbolj sodobno opremo za analizo varnostih ranljivosti. V tem primeru lahko ranljivost sporočijo neposredno proizvajalcu programske opreme ali pa to storijo preko odzivnih centrov. Proizvajalci nato poiščejo način za odpravo ranljivosti in izdajo popravek. Varnostni popravki so v večini primerov brezplačni. Varnostne popravke največkrat izdajo kot programske popravke oziroma kot posodobitve operacijskega sistema ali njegovih komponent. Kadar pa to ni mogoče pa kot novo različico programa.

---

[7] IETF (ang. Internet Engineering Task Force) je organizacija, ki se ukvarja z razvojem in promocijo spletnih standardov.

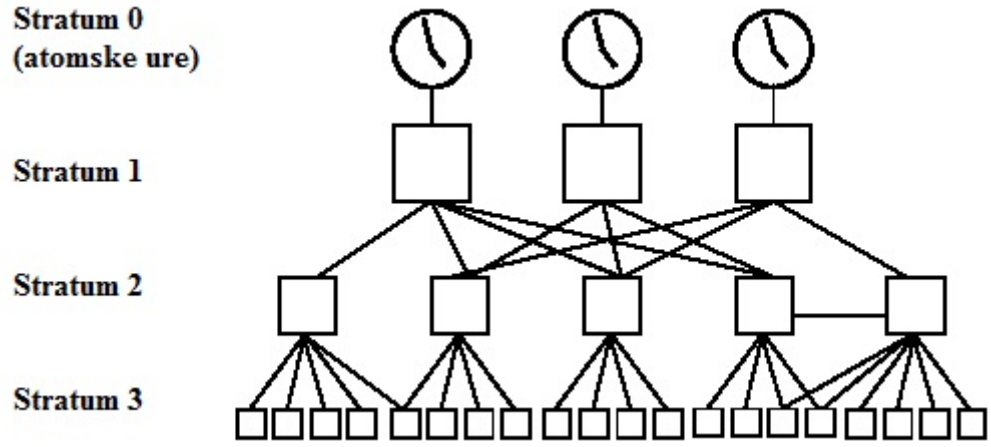
[8] IRTF (ang. Internet Research Task Force) se osredotoča na dolgoročni razvoj interneta in sodeluje z IETF.

[9] IANA (ang. Internet Assigned Numbers Authority) je organizacija, ki nadzira dodeljevanje naslovov IP, dodeljevanje številk avtonomnega sistema, nadzira korenske domenske strežnike, in številke in simbole povezane z internetnimi protokoli.

3.2 Napadi z odbojem preko strežnikov NTP

3.2.1 Protokol NTP

Protokol NTP (ang. Network Time Protocol) je eden izmed najstarejših in je namenjen sinhronizaciji časa sodelujočih naprav z natančnostjo do nekaj milisekund razlike od UTC[10]. Implementacije protokola pošiljajo in sprejemajo časovne žige preko protokola UDP na vratih 123. Časovni žigi so dolžine 64 bitov, pri čemer je 32 bitov za zapis časa v sekundah in 32 bitov za zapis dela sekunde. Prihodnje implementacije protokola bodo morda razširile časovni žig na 128 bitov. Protokol NTP uporablja hierarhično strukturo sistemov razdeljeno v nivoje imenovane "Stratum", ki so oštevilčeni od 0 do 3 [11]. Stratum 0 predstavljajo visoko zanesljive naprave za merjenje časa, kot so naprimer atomske ure. Računalniki Stratuma 1 so sinhronizirani z napako nekaj mikrosekund z napravami Stratuma 0. Računalniki Stratuma 2 so sinhronizirani z računalniki Stratuma 1 in tako naprej. Računalnik je sinhroniziran, če se nahaja v Stratumu do številke 15, Stratum 16 se uporablja kot oznaka, da naprava ni sinhronizirana. Algoritem protokola NTP ustvari Bellman-Fordovo vpeto drevo za izračunavanje najkrajše poti do računalnikov Stratuma 1 za vse odjemalce. V praksi ima drevesna hierarhija redko več kot 3 nivoje.



Slika 5: Arhitektura naprav v protokolu NTP.

[10]UTC (Coordinated Universal Time) je mednarodno sprejet standardni čas.

3.2.2 Analiza napada

Protokol je lahko zlorabljen, ker temelji na protokolu UDP, ki ne nadzira pretoka in ker vsaj eden od njegovih ukazov vrne odgovor veliko daljši od zahtevka. Protokol omogoča, da napadalec strežnikom pošlje nekaj bajtov podatkov naenkrat, strežniki pa žrtvi odgovorijo z nekaj kilo bajti. Večja količina prejetih podatkov oziroma prometa pa bolj obremeni komunikacijski kanal in sisteme, ki komunicirajo preko tega kanala. To imenujemo ojačitev napada. Protokol NTP vsebuje ukaz imenovan monlist (ali MON_GETLIST) [12], ki ga lahko pošljemo strežniku NTP za pregled naprav s katerimi je strežnik komuniciral. Ukaz vrne naslove do 600 naprav. Tako je odgovor veliko daljši kot zahtevek, kar lahko uporabimo za ojačitev napada. Za izvedbo ukaza poženemo v lupini unix ukaz:

"ntpdc –c monlist IP_naslov_strežnika"

Paket za zahtevek je dolg 234 bajtov, odgovor pa je odvisen od števil naslovov, ki jih vrne. Število je omejeno na 600 naslovov, vendar je tak odgovor razporejen na 100 paketkov s skupno velikostjo 48 kilo bajtov. Tako lahko dobimo faktor ojačitve 206 oziroma odgovor je 206 krat večji od zahtevka.

No.	Time	Source	Destination	Protocol	Length	Info
665	*REF*	10.114.1.118	1[redacted]9	NTP	234	NTP Version 2, private
666	0.144916000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
667	0.146839000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
668	0.148329000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
669	0.150853000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
670	0.152744000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
671	0.155101000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
672	0.156374000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
673	0.158604000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
674	0.160587000	1[redacted]9	10.114.1.118	NTP	482	NTP Version 2, private
675	0.160924000	1[redacted]9	10.114.1.118	NTP	122	NTP Version 2, private

Slika 6: Primer odgovora strežnika na ukaz monlist.

Največji primer takega napada se je zgodil 10. februarja 2014. Napadalec je povzročil promet s hitrostjo 400 Gb/s, pri tem je sodelovalo 4529 strežnikov iz 1298 različnih omrežji. Pri tem je v povprečju vsak od njih poslal 87 Mb/s prometa do žrtve v omrežju CloudFare. [12]

Ta dogodek je močno pripomogel k akciji za splošno posodabljanje strežnikov NTP s popravkom, ki preprečuje ponovne zlorabe tega ukaza.

3.2.3 Preprečevanje napada

Napad je možno preprečevati na strani strežnika in na strani odjemalca. Na strani odjemalca lahko omejimo nastavitev, da storitev teče samo kot odjemalec in ne sprejema nobenih poizvedb [13]. Na sistemih UNIX to storimo s sledečimi ukazi:

Delovanje v načinu NTP odjemalec:

"restrict -4 default nomodify nopeer noquery notrap"

"restrict -6 default nomodify nopeer noquery notrap"

Dovoli sporočila iz lokalnih naslovov:

"restrict 127.0.0.1"

"restrict ::1"

Strežniki s katerimi bo sinhroniziran čas:

"server 192.0.2.1 #IPv4 naslov"

"server 2001:DB8::1 #IPv6 naslov"

"server time.example.net #spletni naslov"

Če vzdržujemo strežnik NTP, ki mora biti javno dostopen, je potrebno, da je posodobljen na različico 4.2.7p26 ali novejšo. Popravek onemogoči, da strežnik odgovori na ukaz "monlist". Kadar vseeno potrebujemo ukaz kot je "monlist" je sedaj na voljo ukaz "mrulist", ki sedaj preveri ali je ukaz prišel iz naslova IP, ki je zabeležen v paketku UDP. Za skrbnike omrežij in ponudnike storitve je pomembno, da implementirajo standard BCP-38 [14], ki preprečuje vse napade, ki uporabljajo paketke s ponarejenimi izvornimi naslovi IP.

3.3 Napadi z DNS odbojem

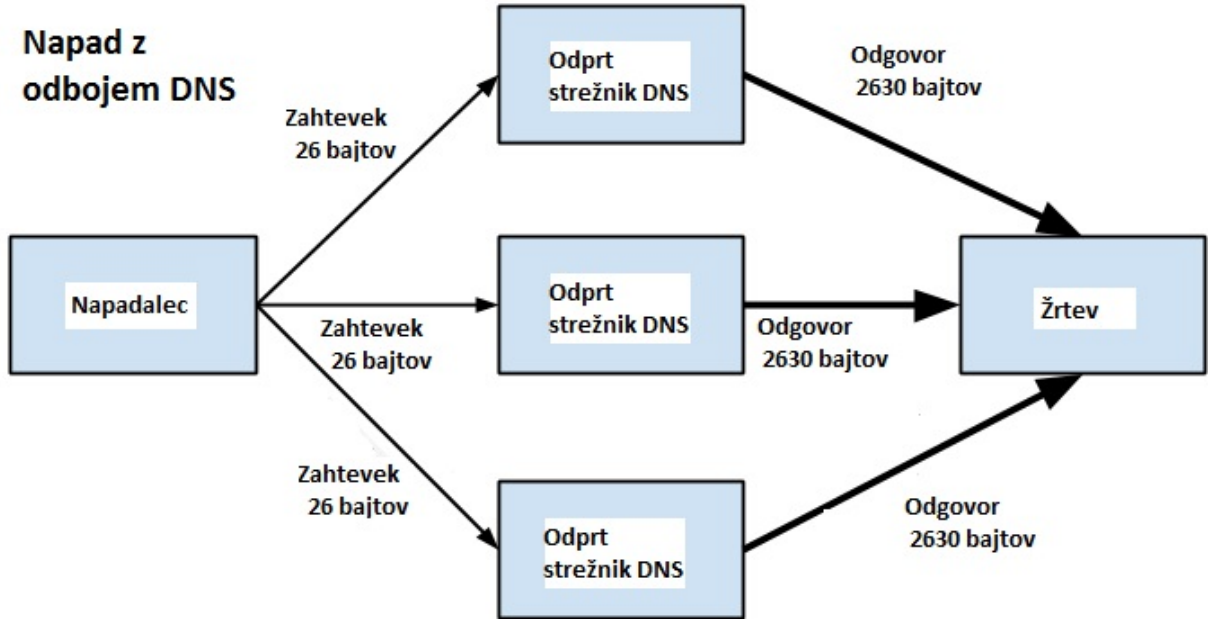
3.3.1 Odprti strežniki DNS

Za delovanje spleta potrebujemo strežnike DNS, ki preslikavajo spletne naslove v naslove IP in obratno, nudijo pa tudi druge storitve. Rekurzivni strežniki DNS so tisti, ki v primeru, da nimajo zapisa za iskani podatek, posredujejo zahtevo strežnikom DNS, ki so višje v hierarhiji. Avtoritativni strežniki DNS imajo podatke samo za svoje omrežje in posredujejo podatke rekurzivnim strežnikom DNS. Vsak ponudnik spletnih storitev ima nekaj odprtih rekurzivnih strežnikov DNS za omogočanje storitve svojim naročnikom. Nekateri rekurzivni strežniki so odprti bolj na široko, na primer Arnesov (193.2.1.66) je dostopen iz vseh slovenskih omrežij, popolnoma odprti pa so strežniki OpenDNS (208.67.222.222 in 208.67.220.220) in Google Public DNS (8.8.8.8 in 8.8.4.4) [15]. Nekateri strežniki DNS pa so široko dostopni tudi ker so napačno nastavljeni in odkrivajo ter sporočajo DNS preslikave vsakomur. Da bi lahko napad izvedli, je potrebno najti še ponudnika, ki omogoča ponarejanje izvornih naslovov, nekateri to omogočajo po pomoti ali pa to nudijo celo kot storitev.

3.3.2 Analiza napada

Da bi bil napad uspešen, mora napadalec poiskati dovolj veliko število odprtih strežnikov DNS, ki bodo skupaj ustvarili več prometa. Napadalec uporabi tudi poseben tip zahtevka DNS imenovan ANY. Zahtevek ANY vrne vse zapise, ki so na voljo za določeno ime domene. [16]. Dodatno podaljšajo odgovor še podatki standarda DNSSEC[11]. Pred uvedbo DNSSEC je dala poizvedba "ripe.net IN ANY?" na odprti rekurzivni strežnik DNS, ki je dolga 26 bajtov, odgovor dolg 250 bajtov, kar je faktor ojačitve 10. Z dodanimi DNSSEC podatki pa je odgovor dolg kar 2630 bajtov, kar je faktor ojačitve 100 [15].

[11] Domain Name System Security Extensions, varnostni dodatki za protokol DNS



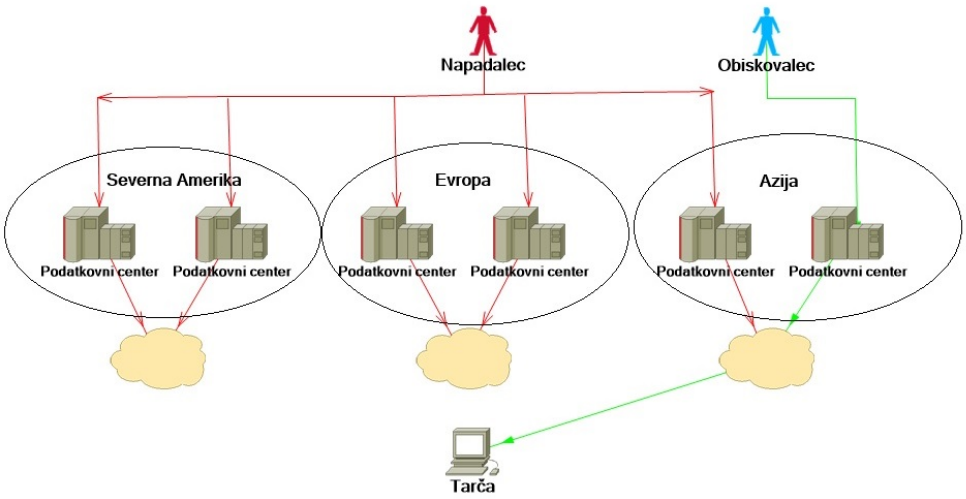
Slika 7: Skica napada

Slika 7 prikazuje tok podatkov od napadalca do žrtve. Napadalec strežnikom pošlje zahtevke z izvornim naslovom IP, ki pripada žrtvi. Ko strežniki odgovorijo na napadalčev zahtevek ustvarijo veliko prometa in z njim poplavijo žrtev. Iskanje vira napada je težje kot pri napadu DDoS z uporabo botnet-a, ker pri napadu ne sodeluje nadzorni strežnik preko katerega daje napadalec ukaze. Iskanje vira odboja DNS zahteva pravočasno sledenje poti preko več omrežij in se največkrat konča le z omejevanjem škode. Napadi te vrste postajajo ponovno aktualni zaradi njihove učinkovitosti. K temu pripomore standard DNSSEC, ki poveča dolžino odgovora in vedno večje število odprtih strežnikov DNS.

3.3.3 Preprečevanje napada

Napadov z odbojem DNS ni mogoče povsem preprečiti, saj za delovanje spleta nujno potrebujemo nekaj odprtih rekurzivnih strežnikov DNS. Napadalcu ne moremo preprečiti, da strežniku pošlje poizvedbo DNS. Lahko pa administratorji onemogočijo prehod paketkom za katere se zazna, da vsebujejo ponarejen izvorni naslov IP. Administratorji spletnih strani lahko omilijo posledice napada z gostovanjem pri ponudnikih za spletno gostovanje, ki imajo zelo veliko infrastrukturo. Velika infrastruktura povzroči, da je na voljo več virov kot jih napadalec lahko zaseže. Napad izgubi moč, ker se zlonameren promet porazdeli. Oslabljen tok zlonamernega prometa, ki prispe do posameznega podatkovnega centra ni dovolj velik, da bi onemogočil storitev. Naprave v podatkovnem centru so sposobne samodejno zaznati

zlonameren promet. Dodaten nadzor nudijo še zaposleni. Zlonamerne paketke usmerjevalniki v podatkovnih centrih zavržejo in tako omogočijo prehod samo pravemu prometu. V primeru, ko je onesposobljen celoten podatkovni center, je lahko storitev še vedno dostopna preko ostalih strežnikov v podatkovnih centrih. Slika 8 prikazuje napadalca, ki pošilja zlonameren promet, označen z rdečo. Zlonameren promet se porazgubi v infrastrukturi ponudnika spletnega gostovanja. Tudi če pride do preobremenitve posamezne linije lahko ponudnik promet preusmeri po delujočih linijah. V primeru, da so poti preko Severne Amerike in Evrope preobremenjene lahko obiskovalec svoj promet, označen z zeleno, še vedno pošilja preko podatkovnega centra v Aziji.



Slika 8: Napad DDoS se porazdeli v infrastrukturi ponudnika spletnega gostovanja.

Za preprečevanje teh napadov je pomembno, da kot upravljalec strežnika DNS omejimo njegovo področje delovanja in onemogočimo rekurzivno razreševanje naslovov. Na implementaciji Bind to storimo z ukazoma:

"options { allow-query {192.168.1.0/24;};};"

"options { recursion no; };"

SI-CERT priporoča, da upravljavci omrežij omejijo dostop do rekurzivnega strežnika z direktivo allow-recursion le na lastna omrežja. Vgrajeni strežnik DNS na operacijskih sistemih Microsoft Windows Server te možnosti žal nima [15]. Zato je omejitve potrebno nastaviti na požarni pregradi. V primeru, da želimo na operacijskih sistemih Microsoft Windows Server zaganjati tako avtoritativni, kot tudi rekurzivni strežnik, ju moramo ločiti.

Avtoritavni strežnik DNS je lahko javno dostopen, rekurzivni strežnik DNS pa mora biti dostopen samo z notranjega omrežja

3.4 Napadi na zaščito s protokolom SSL/TLS

3.4.1 Protokol SSL/TLS

Protokol TLS[12] in njegov predhodnik protokol SSL[13] sta kriptografska protokola, zasnovana za varno komunikacijo preko spleta. Uporabljata certifikate x.509 in asimetrično kriptografijo za izmenjavo ključev, ki jih nato uporablja za simetrično kriptiranje povezave, ker je simetrična kriptografija hitrejša za izvedbo. Protokol vzpostavi varno komunikacijo v štirih fazah.

1. Faza pogajanja (ang. Negotiation Phase)

    - Odjemalec pošlje sporočilo **ClientHello**, ki vsebuje najvišjo različico TLS, ki jo podpira, naključno število in seznam šifrirnih ter kompresijskih metod.

    - Strežnik se odzove s sporočilom **ServerHello**, ki vsebuje izbrano različico protokola, naključno število, šifrini sistem in kompresijsko metodo, ki jih izbere iz seznama poslanih s strani odjemalca. Tako imata oba potrebne podatke za vzpostavitev kriptirane komunikacije.

2. Strežnik pošlje sporočilo **ChangeCipherSpec**, ki odjemalcu pove, da bodo odslej sporočila kriptirana.

    - Strežnik pošlje šifrirano sporočilo **Finished**, ki vsebuje zgoščeno vrednost in MAC.

    - Odjemalec poskusi dekriptirati sporočilo, če mu ne uspe prebrati vrednosti in naslova MAC strežnika, se mora postopek začeti znova.

3. Odjemalec pošlje sporočilo **ChangeCipherSpec**, ki pove strežniku, da bodo odslej odjemalčeva sporočila kriptirana.

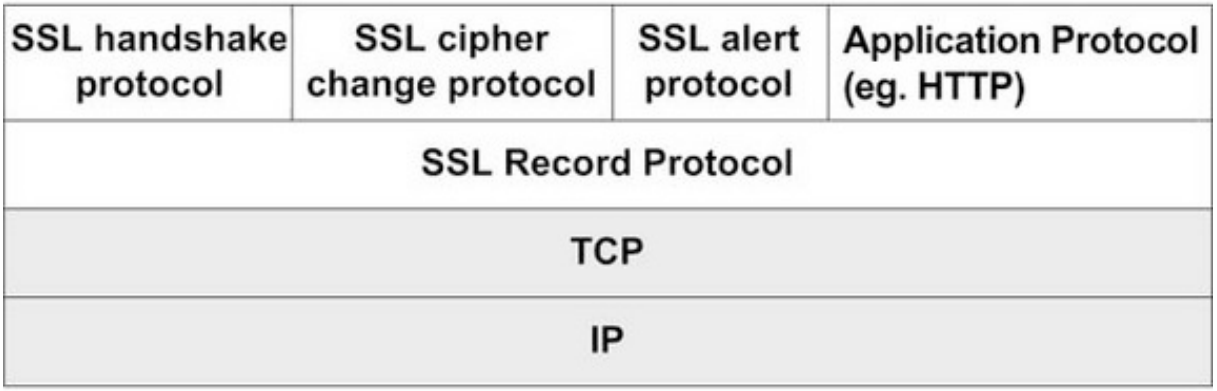
    - Odjemalec pošlje svoje sporočilo **Finished**.

    - Strežnik izvede dekripcijo in potrdi zgoščeno vrednost in MAC.

[12] Transport Layer Security
[13] Secure Sockets Layer

4. Aplikacijska faza

    - "Rokovanje" je končano, sporočila so sedaj kriptirana z enakim postopkom kot sporočili Finished.

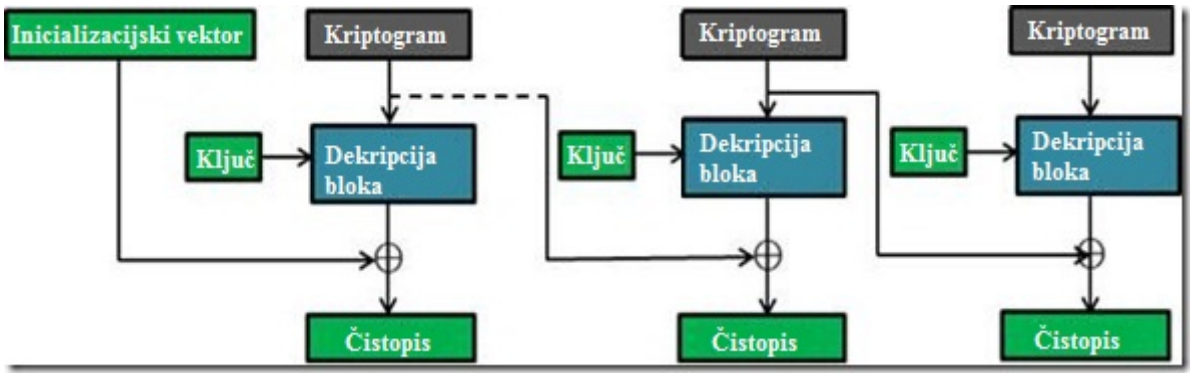


Slika 9: Arhitektura SSL/TLS.

3.4.2 Varnostna ranljivost BEAST

BEAST[14] izkorišča ranljivost v CBC[15], ki je dodatek k simetričnemu algoritmu za kriptiranje in dekriptiranje. CBC skrbi za veriženje tako, da uporablja inicializacijski vektor s katerim v binarni obliki naredi operacijo XOR nad čistopisom, da dobi kriptogram[16]. S tem preprečuje napad s substitucijo. Za dekripcijo izvede obraten postopek. Ranljivost izkorišča lastnost protokola, ki za kriptiranje oziroma dekriptiranje trenutnega bloka uporablja poleg ključa kriptirano sporočilo iz prejšnjega bloka.

[14] Browser Exploit Against SSL/TLS
[15] Cipher-block chaining
[16] Sporočilo, ki je spremenjeno v prikrito obliko



Slika 10: Potek CBC kriptiranja

Cilj napada je dekriptirati piškotek, ki ga uporabnik uporablja za vzdrževanje seje. Napadalec ima v rokah inicializacijski vektor. Napadalec poskuša uganiti čistopis tako, da na čistopisu uporabi operacijo XOR z inicializacijskim vektorjem, ki je kriptogram s prejšnjega bloka in primerja vrednost s trenutnim kriptogramom. Na primer, da imamo sporočilo v katerem se nahaja identiteta seje "JSESSIONID=Gxs36NepewqeMI763Hej31pkl". Napadalec lahko predvideva del "JSESSIONID=". Ker bi bilo ugibanje cele vrednosti zamudno, jo lahko ugiba tudi znak po znak in vsakič pogleda, do kam se binarni niz ujema s kriptogramom. Tako lahko po nekaj poskusih ugane "JSESSIONID=G" in tako dalje do konca niza. Napad je sicer težje izvedljiv saj mora napadalec zadostiti naslednjim pogojem:

- Napadalec mora biti na istem omrežju in izvesti napad s posrednikom
- Napadalec mora spreminjati promet, da lahko primerja, če se rezultat ujema. Da to stori, mora za vsako ugibanje posredovati več zahtevkov
- Napadalec lahko ugane samo en blok naenkrat.

Varnostno ranljivost odpravimo tako, da posodobimo različico protokola TLS. Ranljivost je prisotna v različici TLS 1.0 in odpravljena v različicah 1.1 in 1.2 z uporabo edinstvenih inicializacijskih vektorjev.

3.4.3 Varnostna ranljivost CRIME

CRIME[17] je varnostna ranljivost, ki izkorišča zgoščevanje v TLS in SPDY [17]. Protokol SPDY je odprt omrežni protokol, ki so ga razvili pri podjetju Google za prenos spletnih vsebin. Protokol SPDY spreminja promet http z namenom hitrejšega nalaganja spletnih strani in z namenom izboljšati varnost. To doseže z uporabo kompresije, multipleksiranja in

[17] Compression Ratio Info-leak Made Easy

prioritizacije [18].Varnostno ranljivost Truncation sta razkrila raziskovalca računalniške varnosti Juliano Rizzo in Thai Duong, ki sta razvila tudi orodje za izkoriščanje ranljivosti BEAST. [19] Ranljivost je bila razkrita leta 2012 na varnostni konferenci ekoparty. Temelji na analizi kriptograma, če napadalcu uspe razkriti vsebino skritih piškotkov, lahko prevzame že vzpostavljeno sejo in jo izkoristi za nadaljevanje napadov. Zgoščevanje podatkov si poglejmo na primeru preprostega niza črk. Ponavljajoče se znake zapišemo kot znak in število ponovitev.

Primer: niz "AAAAABCDEFGH" lahko zapišemo kot "5ABCDEFGH".

```
Izvorni niz          Zgoščeni niz     Kriptogram
-----------------------------------------------------
AAAAABCDEFGH    =    5ABCDEFGH    =    QvnQSHvQWB3*QR
```

Slika 11: Zgoščevanje in kriptiranje niza nezgoščenim

Ko tako zgoščeno vrednost kriptiramo dobimo krajši kriptogram. Tako lahko napadalec k podatkom dodaja naključne nize in opazuje kako se spremeni kriptogram. Kadar mu uspe dodati znake, ki jih vsebuje prvotni niz, opazi skrajšanje prvotnega niza v kriptogramu. Tako lahko ve, da niz vsebuje znake, ki jih je dodal. Če napadalcu neznani podatki vsebujejo niz "ABCDEFGH", potem bo ob dodajanju niza "AAA" prišlo do zgoščevanja in rezultat bo krajši kriptogram. Na tak način lahko napadalec ugotovi, da niz vsebuje "AAA".

```
Izvorni niz              Zgoščeni niz                    Kriptogram
-------------------------------------------------------------------

ZZZ[Neznani podatki] = [Zgoščeni neznani podatki] = QvnQSHvQWB3*QR
YYY[Neznani podatki] = [Zgoščeni neznani podatki] = f*fB&M7sya*u7F
AAA[Neznani podatki] = [Zgoščeni neznani podatki] = rAW^26uffH%8
```

Slika 12: Kriptogram je krajši kadar pride do zgoščevanja.

Ugibanje celotnega niza na ta način je zamudno. Napadalec lahko ugibanje pospeši tako, da vstavlja v znake, ki jih ugiba, vse kar ve o prometu, ki se prenaša. Poskusi lahko s polji v glavi http zahtevka, katerih imena in zaporedje so znani ali z drugimi polji in vrednostmi, ki jih lahko predvideva. Ranljivost se lahko odpravi z izklopom zgoščevanja TLS in zgoščevanja SPDY. S tem izgubimo hitrost nalaganja spletnih strani. Zgoščevanje na strani strežnika je odvisno od zgoščevanja na strani odjemalca. Če je zgoščevanje izključeno na strani odjemalca, strežnik samodejno izključi uporabo zgoščevanja. Potrebna je tudi nadgradnja spletnih brskalnikov, ki uporabljajo zgoščevanje TLS na najnovejšo različico.

3.4.4 Varnostna ranljivost BREACH

Varnostna ranljivost BREACH[18]temelji na ranljivosti CRIME. Razkrili so jo na varnostni konferenci Black Hat avgusta 2013. Deluje podobno kot CRIME, vendar ne izkorišča varnostne ranljivosti v SSL/TLS, omogoča pa, da se dekriptira kriptograme zajete v povezavi SSL/TLS. BREACH izrablja zgoščevanje v protokolu HTTP. Zgoščevanje poteka tako, da podniz, ki ga najdemo v nizu, zapišemo kot zaporedje znakov z v prvotnem nizu z uporabo indeksov. Na primer niz "Googling the googles" bi bil zapisan kot "Googling the g(-13,4)s". Oznaka "(-13,4)" pomeni, da se bodo na mestu oznake nahajali znaki, ki se začnejo z znakom trinajst znakov pred oznako in vsebujejo naslednje štiri zaporedne znake. Za zgoščevanje se uporablja tudi Huffmanovo kodiranje. Celoten postopek je opisan v RFC 2616 [20]. Napadalec vstavlja v nize, ki se zgoščujejo, znake. Vstavljajo vsak znak posamično in opazujejo, kako se je spremenil kriptogram. Kadar je kriptogram manjši kot v prejšnjem poskusu, napadalec ve, da je našel pravilen znak.

Da je ranljivost mogoče izkoristiti, morajo biti izpolnjeni naslednji pogoji:

- Uporabljen mora biti strežnik HTTP, ki uporablja zgoščevanje
- Uporabnikov vnos mora biti vključen v odgovore HTTP. V odgovorih HTTP mora biti vključena skrivnost, na primer žeton CSRF
- Napadalec mora najprej žrtvin promet usmeriti preko svojega računalnika oziroma uporabiti napad s posrednikom.

Napad BREACH je možno izvesti s samo nekaj tisoč zahtevki in je lahko končan v roku 60 sekund [21]. Varnostna ranljivost napadalcu omogoča, da ugane sporočilo HTTP. Napadalčev cilj je pogosto, da ugane sporočilo, ki vsebuje skriti spletni piškotek za avtentikacijo. To napadalcu omogoči krajo že prijavljene seje.

Za preprečevanje napada ne obstaja celovita rešitev. Na voljo pa je nekaj tehnik [22]. Tukaj so navedene po njihovi učinkovitosti od najbolj do najmanj učinkovite:

- Izklop zgoščevanja HTTP

[18] Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext

Z izklopom zgoščevanja v protokolu HTTP ranljivost popolnoma odpravimo. Vendar pa tak ukrep občutno upočasni nalaganje spletnih strani zato je potrebno poiskati še dodatne rešitve.

- Ločevanje skrivnosti od uporabniškega vnosa

  Eden izmed pristopov je, ki reši problem je, da uporabimo drugačen pristop za zgoščevanje uporabniškega vnosa, kot za zgoščevanje skrivnosti. Zgoščujemo samo uporabniški vnos, za skrivnosti pa zgoščevanja ne uporabljamo. S tem dosežemo, da napadalec ne more dekriptirati skrivnosti z uporabo varnostne ranljivosti BREACH. Aplikacije pa ni vedno mogoče implementirati na tak način.

- Skrivanje dolžine

  Ker napad temelji na opazovanju dolžine kriptograma, napad otežimo če poskusimo skriti dolžino kriptograma. V vsak odgovor HTTP dodamo nekaj naključnih podatkov. S tem dosežemo, da prava dolžina kriptograma ostane skrita. Kljub temu pa napadalec še vedno lahko ugane sporočilo, poslati pa mora tako veliko več zahtevkov http, da napad postane nepraktičen.

- Maskiranje skrivnosti

  Napad se zanaša na to, da iskana skrivnost ostane enaka med zahtevki. Namesto, da skrivnost S pripnemo v spletno stran lahko v vsakem zahtevku generiramo enkratni "pad" P. V spletno stran nato pripnemo $P||(P \oplus S)$[19]. To sicer podvoji dolžino vsake skrivnosti ampak zagotovi, da skrivnosti ni mogoče zgoščevati.

- Omejevanje števila zahtevkov in njihov nadzor

  Čeprav napad za uspešno izvedbo ne potrebuje zelo velikega števila zahtevkov, še vedno potrebuje veliko več zahtevkov, kot pa bi jih poslal človek. Če omejimo število zahtevkov v določenem času, ki jih lahko pošlje posamezen uporabnik, lahko izvedbo napada občutno upočasnimo. To pripomore k odvračanju napadalcev od izvedbe napada.

[19] Tukaj znak || pomeni konkatenacijo in znak $\oplus$ pomeni operacijo XOR

- Ščitenje ranljivih strani z uporabo zaščite proti CSRF[20]

  Napadi CSRF so pogosto uvod v napad z uporabo varnostne ranljivosti BREACH. Napadalec poskuša žrtvi ukrasti skrivnost, pogosto v obliki skritega spletnega piškotka. To lahko stori tako, da žrtvi pošlje skrito povezavo na drugo spletno stran, ki hrani žrtvin piškotek, preko spletne strani v katero je žrtev prijavljena. Žrtvin brskalnik bo brez njene privolitve poslal napadalčev zahtevek HTTP spletni strani, ki žrtvi zaupa [23]. Kljub temu bo promet med spletno stranjo in žrtvijo kriptiran, napadalec pa lahko kriptiran promet poskuša dekriptirati z uporabo varnostne ranljivosti BREACH. Napade CSFR lahko preprečujemo tako, da v zahtevek dodamo unikatni žeton s katerim potrdimo, da je zahtevek res poslal pravi uporabnik.

[20] Cross Site Request Forgery

3.4.5 Varnostna ranljivost Truncation

Varnostna ranljivost omogoča, da napadalec prepreči, da bi žrtvin zahtevek za izpis iz spletne storitve prispel do strežnika. Ko žrtev pošlje zahtevek za izpis, ji strežnik odgovori s kodo, ki se izvrši na njenem brskalniku, žrtvi prikaže sporočilo, da je bila odjavljena, storitvam na strežniku pa pošlje zahtevke za prekinitev povezave. Napadalec lahko te zahtevke prestreže, žrtvi pa odgovori z nekriptiranim sporočilom TCP FIN. Tako žrtev misli, da je odjavljena iz storitve, vendar pa je njen brskalnik s storitvijo še vedno povezan. Napadalec nato potrebuje fizičen dostop do žrtvinega računalnika. Napad je bil prikazan na varnostni konferenci Black Hat USA 2013 na primeru računov Hotmail in Google. Po opravljenem napadu, ko žrtev misli, da je uspešno odjavljena, napadalec v žrtvin brskalnik vnese naslov poštnega predala žrtve in tako dobi poln dostop do njenega poštnega predala. Čeprav za menjavo gesla storitev zahteva vnos starega gesla, nekatere storitve omogočajo zamenjavo na druge načine. Hotmail omogoča vnos nadomestnega poštnega naslova, kamor posredujejo povezavo za neposredno zamenjavo gesla. Na ta način lahko napadalec žrtvi celo prevzame poštni račun. Za ta način napada so ranljivi predvsem uporabniki javnih računalnikov. [24] Zato je učinkovit način za preprečevanje tovrstnih zlorab nadzor oseb, ki do naprav dostopajo in fizično varovanje opreme.

3.4.6 Napadi na šifrirni sistem RC4 v protokolu TLS

RC4[21] je šifrirni algoritem. Razvil ga je Ronald Rivest leta 1987. Za šifriranje uporablja simetrični ključ spremenljive dolžine od 1 do 256 zlogov za inicializacijo 256-zlogovne tabele stanj. [25] Na konferenci USENIX Security Symposium 15. Avgusta 2013 je bil predstavljen način napada na RC4 v protokolu TLS, ki so ga razvili raziskovalci skupine Information Security Group iz Royal Holloway, University of London. Napad potrebuje več sej in išče čistopis. Čistopis se mora nahajati vedno na enakem mestu v več povezavah ali sejah TLS. Izkorišča korekcije dolžine enega ali dveh bajtov v prvih 256 bajtih niza simbola ključa. Število kriptogramov, ki so potrebni, da zanesljivo dekriptira 16 zaporednih bajtov čistopisa je $10*2^{30}$, vendar je uspešnost 50 odstotna že pri $6*2^{30}$ zajetih kriptogramih. Celoten postopek napada na RC4 je opisan v dokumentu On the Security of RC4 in TLS and WPA, ki je dostopen na spletni povezavi http://www.isg.rhul.ac.uk/tls/RC4biases.pdf [26]. Pred napadi na šifrirni sistem RC4 se lahko varujemo tako, da damo pri vzpostavljanju povezave TLS prednost drugim naborom šifrirnih sistemov, ki jih implementacija protokola TLS podpira.

[21] Rivest Cipher 4

3.4.7 Varnostna ranljivost Heartbleed

Heartbleed je varnostna ranljivost v kriptografski knjižnici OpenSSL, ki je najbolj razširjena implementacija protokola TLS. Ranljivost je prvi sporočil Neel Mehta, varnostni strokovnjak pri podjetju Google, 1. Aprila 2014. Ime Heartbleed pa je skoval uslužbenec Finskega podjetja Codenomicon, ki je tudi registriralo domeno Heartbleed.com in ustvarilo logotip krvavečega srca. Pri podjetju Codenomicon priznavajo, da je ranljivost prvi sporočil Neel Mehta vendar pravijo, da so ranljivost odkrili tudi sami brez zunanje pomoči. V bazi ranljivosti Common Vulnerabilities and Exposures je označen kot CVE-2014-0160. Ranljivost je še posebej pomembna, ker napadalcu ni potrebno izvesti napada s posrednikom, temveč lahko napade ranljivi strežnik neposredno. Prav tako v sistemu ne pusti nobene sledi tako, da ni mogoče oceniti ali je prišlo do kraje podatkov.

Heartbleed je bil deležen velike pozornosti, ker je bilo zaradi napake prizadetih veliko število spletnih strežnikov. Razlogov za to je več. Knjižnico OpenSSL, ki vsebuje programsko napako, uporabljajo strežniki Apache in strežniki nginx. Podjetje Netcraft je aprila 2014 opravilo raziskavo o uporabi spletnih strežnikov, v kateri so ugotovili, da je tržni delež strežnikov Apache in ngix na spletu kar 66% [27]. OpenSSL pa se uporablja tudi pri zaščiti strežnikov za elektronsko pošto, ki uporabljajo protokole SMTP, POP in IMAP, strežnikov za spletni klepet, ki uporabljajo protokol XMPP in omrežij VPN[22], ki uporabljajo protokol SSL za zaščito seje. Prizadete so bile tudi storitve najbolj popularnih spletnih strani kot so Amazon, Google, Facebook, Netflix, SoundClound in YouTube [28].

Zato smo se odločili prikazati napad z uporabo varnostne ranljivosti Heartbleed. Namen je demonstrirati, kako lahko ugotovimo, ali je spletni strežnik ranljiv, kako in kakšno škodo lahko tak napad povzroči, ter kako ranljivost odpravimo.

[22] Virtual Private Networks

Poglavje 4 Izvedba napada z uporabo ranljivosti Heartbleed

4.1 Analiza napada

Ranljivost je označena kot "buffer over-read" oziroma predolgo branje iz pomnilnika. Protokol uporablja napako v ukazu - razširitvi protokola TLS imenovani Heartbeat. Heartbeat omogoča, da se ohranja povezava TLS klub temu, da ne pošiljamo nobenih podatkov. Protokol Heartbeat vsebuje ukaz "heartbeat request" s katerim odjemalec pošlje sporočilo, ki vsebuje podatke in njihovo dolžino. Strežnik na odjemalčevo zahtevo odgovori z enakimi podatki. Napaka v programski kodi pa omogoča, da napadalec pošlje sporočilo, ki vsebuje en bajt podatkov, poleg tega pa strežniku posreduje, da so podatki dolgi 65535 bajtov. To pa povzroči, da strežnik v pomnilniku poišče pomnilniški naslov na katerem se nahajajo napadalčevi podatki, nato pa zajame še naslednjih 65534 bajtov podatkov, ki se slučajno nahajajo v pomnilniku in jih posreduje nazaj napadalcu. Posredovani podatki so sicer naključni, vendar lahko vsebujejo uporabniška imena, gesla ali pa ključe, ki so bili uporabljeni za kriptiranje povezave SSL/TLS.

4.2 Ranljivost v kodi

4.2.1 Zahtevek heartbeat

Koda pravi, da naj "payload" in "padding" ne presegata 16381 bajtov. To se preverja v funkciji OPENSSL_assert. Nato se prenese zahtevek, ki vsebuje:

- En bajt 0x01, ki pove, da je to TLS heartbeat zahtevek (TLS_HB_REQUEST)
- Dva bajta, ki vsebujeta 16 bitno predstavitev števila 34
- Dva bajta podatka "payload", ki vsebujeta 16 bitno zaporedno število
- Šestnajst bajtov naključnih podatkov, da se zapolni 18 bajtni "payload"
- Šestnajst naključnih "padding" bajtov, ki jih potrebuje standard

```
unsigned int padding = 16; /* Use minimum padding */

/* Check if padding is too long, payload and padding
 * must not exceed 2^14 - 3 = 16381 bytes in total.
 */

OPENSSL_assert(payload + padding <= 16381);

/* Create HeartBeat message, we just use a sequence number
 * as payload to distuingish different messages and add
 * some random stuff.
 *  - Message Type, 1 byte
 *  - Payload Length, 2 bytes (unsigned int)
 *  - Payload, the sequence number (2 bytes uint)
 *  - Payload, random bytes (16 bytes uint)
 *  - Padding
 */
buf = OPENSSL_malloc(1 + 2 + payload + padding);
p = buf;
/* Message Type */
*p++ = TLS1_HB_REQUEST;
/* Payload length (18 bytes here) */
s2n(payload, p);
/* Sequence number */
s2n(s->tlsext_hb_seq, p);
/* 16 random bytes */
RAND_pseudo_bytes(p, 16);
p += 16;
/* Random padding */
RAND_pseudo_bytes(p, padding);

ret = dtls1_write_bytes(s, TLS1_RT_HEARTBEAT, buf, 3 + payload + padding);
```

Slika 13: Implementacija zahtevka heartbeat v programskem jeziku C.

4.2.2 Odgovor na zahtevek heartbeat

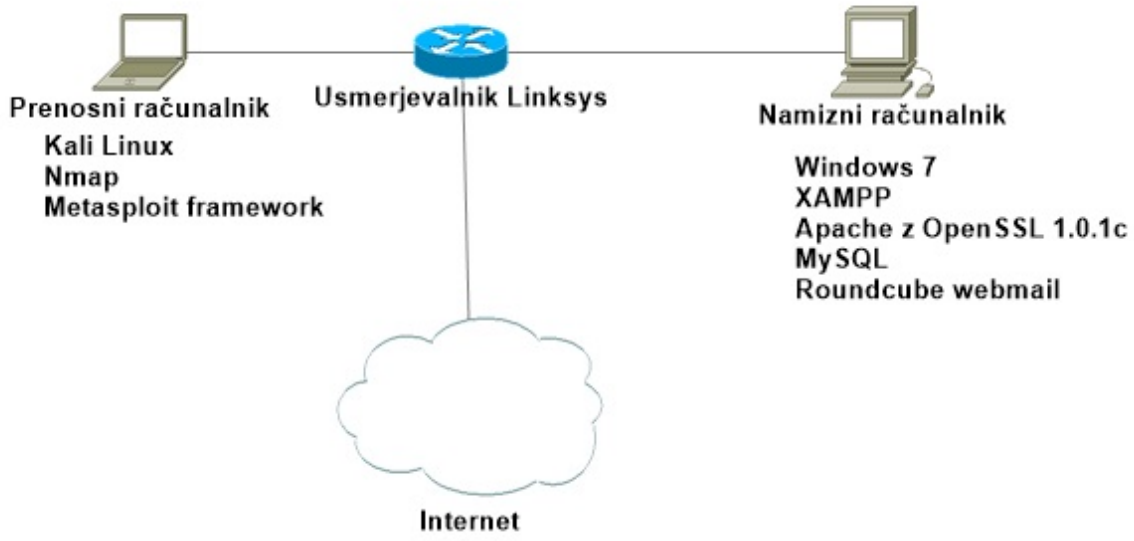
Ko pa se ranljiva različica OpenSSL odzove na zahtevek heartbeat, ni pozorna na pravilnost podatkov, ki jih prejme. Za zagotovitev, da je povezava kriptirana v obe smeri je potrebno, da strežnik vrne enake podatke, kot jih je posredoval odjemalec. Izkaže se, da lahko pošljemo majhen zahtevek heartbeat, vendar nastavimo njegovo dolžino na 0xFFFF oziroma 65335 bajtov. Tako se v odgovor prepišejo prejeti podatki in še naslednjih 65334 bajtov iz pomnilnika. Tako nastane iztekanje podatkov v velikosti približno 64KB vsakič, ko strežnik prejme zlonameren zahtevek heartbeat.

```
/* Allocate memory for the response, size is 1 byte
 * message type, plus 2 bytes payload length, plus
 * payload, plus padding
 */
buffer = OPENSSL_malloc(1 + 2 + payload + padding);
bp = buffer;

/* Enter response type, length and copy payload */
*bp++ = TLS1_HB_RESPONSE;
s2n(payload, bp);
memcpy(bp, pl, payload);
bp += payload;
/* Random padding */
RAND_pseudo_bytes(bp, padding);

r = dtls1_write_bytes(s, TLS1_RT_HEARTBEAT, buffer, 3 + payload + padding);
```

Slika 14: Implementacija odgovora na zahtevek heartbeat v programskem jeziku C.

4.3 Orodja za prikaz in izvedbo

Za prikaz in izvedbo napada je bil pripravljen izoliran testni poligon - dva računalnika, prenosni in namizni osebni računalnik. Računalnika sta povezana v lokalno omrežje s privzetim prehodom 192.168.1.1. Na prenosni računalnik (napadalčev) je bil nameščen operacijski sistem Kali Linux, ki vsebuje veliko orodij za izvedbo penetracijskih testov. Uporabili smo orodje Nmap, s katerim smo preverili, ali je ranljivost prisotna. Za izkoriščanje ranljivosti smo uporabili openssl_heartbleed modul orodja Metasploit Framework. Na namiznem računalniku (žrtvinem) teče operacijski sistem Microsoft Windows 7. Nanj je bil nameščen paket strežnikov XAMPP, ki vsebuje strežnik Apache in strežnik MySQL. Na strežnik Apache je bil nameščen odprtokodni odjemalec elektronske pošte Roundcube.



Slika 15: Omrežje za prikaz napada z uporabo ranljivosti Heartbleed

4.3.1 Kali Linux

Kali Linux distribucija Linuxa je zasnovana za digitalno forenziko in penetracijsko testiranje. Kali Linux vzdržuje in razvija podjetje Offensive Security. Je naslednik distribucije BackTrack, ki so jo prav tako razvijali pri Offensive Security. Ob namestitvi vsebuje veliko orodij, med njimi tudi Nmap, Wireshark in Metasploit Framework. Namestitev lahko opravimo v ukazni vrstici, na voljo pa je tudi sodoben grafični vmesnik.

4.3.2 Nmap

Nmap je leta 1997 izdal Gordon Lyon pod licenco GNU GPL in ga še vedno razvijajo številni strokovnjaki računalniške varnosti. [7] Omogoča odkrivanje gostiteljev, ki se odzovejo na TCP in ICMP zahtevke, izpiše vrata na katerih naprave poslušajo in komunicirajo, zazna operacijski sistem, omogoča pa tudi omejeno interakcijo z napravo preko vgrajenega skriptnega sistema. Za uporabo obstajajo različni grafični vmesniki, v osnovi pa se program izvaja v ukazni vrstici.

Primeri ukazov v orodju Nmap:

- Za podatke o tarči

```
nmap <tarčin URL ali IP naslov s presledki med posameznimi bloki>
```

- Za zaznavanje operacijskega sistema

```
nmap -O <tarčina domena ali IP naslov>
```

- Za pregledovanje z uporabo TCP paketkov z zastavico SYN

```
nmap -sS -p <vrata tarče> <tarčin IP naslov>
```

4.3.3 Metasploit Framework

Metasploit Framework je produkt projekta Metasploit Project izdan pod različnimi licencami. Zasnovan je kot orodje za razvoj in izvedbo kode za izkoriščanje varnostnih ranljivosti na oddaljeni napravi. V različici 3.0 so vsebovana tudi orodja za odkrivanje znanih varnostnih ranljivosti na oddaljeni napravi. Osnovni koraki za uporabo vključujejo:

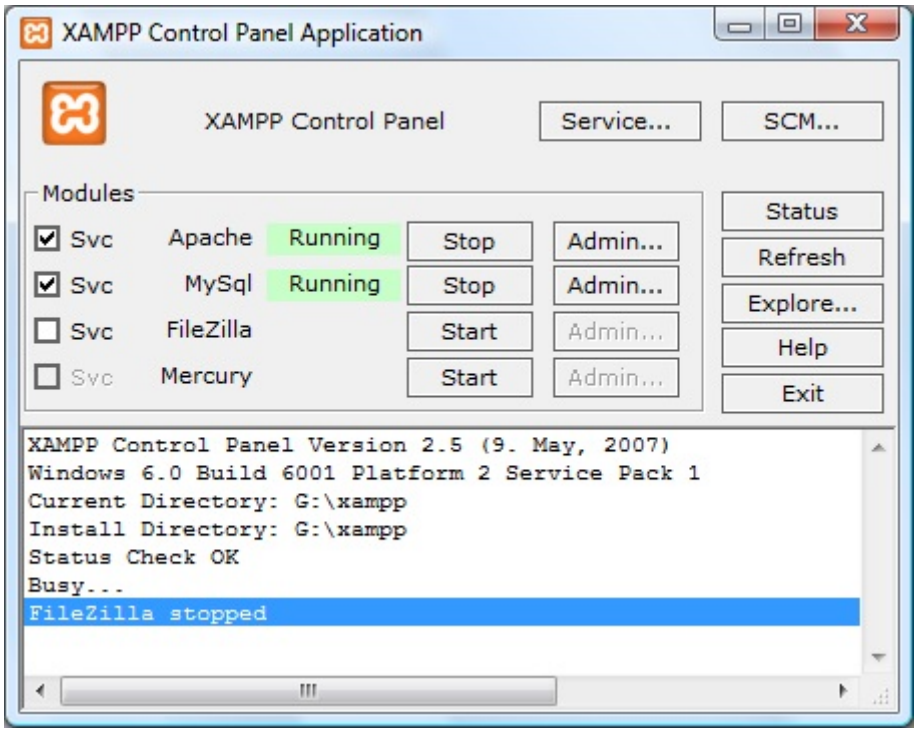
- Izbiro modula za izkoriščanje želene varnostne ranljivosti
- Pregledovanje oddaljene naprave za ranljivost
- Izbira in nastavitev kode, ki se bo izvršila
- Izbira tehnike kodiranja za preprečitev zaznavanja vdora

- Izvedba kode

Metasploit teče na operacijskih sistemih Unix in Windows. Ima tudi sposobnost vnosa podatkov, ki jih izvozijo programi za skeniranje ranljivosti kot so Nmap, Nexpose ali Nessus. Podatki pripomorejo k učinkoviti izvedbi kode za izrabo varnostne ranljivosti.

4.3.4 XAMPP

XAMPP je paket strežnikov, ki omogoča njihovo grafično namestitev, konfiguracijo in upravljanje. Vsebuje strežnike Apache, MySQL, FileZilla FTP, Tomcat in Mercury. Ime je akronim za Cross Apache MySQL PHP Perl, ki so strežniki in programski jeziki katerih konfiguracijo in uporabo omogoča. Zasnovan je bil za testiranje spletnih produktov na lokalnem sistemu, brez dostopa do spleta. Odlikujejo ga preprosta namestitev z uporabo grafičnega vmesnika, preprosto upravljanje vključenih produktov in robustnost.



Slika 16 Nadzorna plošča orodja XAMPP

4.3.5 Apache strežnik

Spletni strežnik Apache je zelo razširjen strežnik http. Njegov razvoj je zelo pripomogel k širitvi svetovnega spleta. Podpira programske jezike PHP, Tcl, Perl in Python. Za naš prikaz nadapa je pomembo tudi, da podpira SSL z uporabo kriptografske knjižnice OpenSSL. Za namen demostracije je bila nameščena različica 1.0.1c, ki vsebuje varnostno ranljivost. Da je bil omogočen openssl na lokalnem strežniku je bila v datoteko php.ini dodana vrstica "extension=php_openssl.dll". Na spletni strežnik Apache je bil nato nameščen odjemalec elektronske pošte Roundcube.

4.3.6 MySQL

MySQL je odprtokodna implementacija podatkovne baze, ki za svoje delovanje uporablja jezik SQL in sistem za upravljanje s podatkovnimi bazami. Napisan je programskih jezikih C in C++. Deluje tako na Unix kot na operacijskih sistemih Windows. Nameščena je bila v sklopu paketa XAMPP. S pomočjo sistema MySQL je bila ustvarjena podatkovna baza "Email", ki je bila uporabljena za delovanje odjemalca elektronske pošte Roundcube.

4.3.7 Roundcube odjemalec elektronske pošte

Roundcube je IMAP odjemalec elektronske pošte, zasnovan za delovanje na spletnem strežniku. Napisan je v programskem jeziku PHP. Za svoje delovanje potrebuje dostop do strežnika IMAP in SMTP. Za primer napada sta bila uporabljena Googlova strežnika. Namestitev se zažene preko spletnega brskalnika. Datoteke odjemalca najprej kopiramo v mapo, ki je dostopna na Apache strežniku. Kadar uporabljamo XAMPP je to običajno C:\xampp\htdocs\imemape. Nato se s spletnim brskalnikom povežemo na localhost/imemape in zažene se nam namestitveni vmesnik za Roundcube. Izbrane so bile privzete nastavitve, pri izbiri podatkovne baze je bila vnesena baza, ki smo jo ustvarili v sistemu MySQL. Ker smo želeli uporabiti varno povezavo SSL so bili vneseni ssl://imap.gmail.com in vrata 993 ter ssl://smtp.gmail.com in vrata 465. Poleg tega so bili vneseni še prijavni podatki za Google račun.

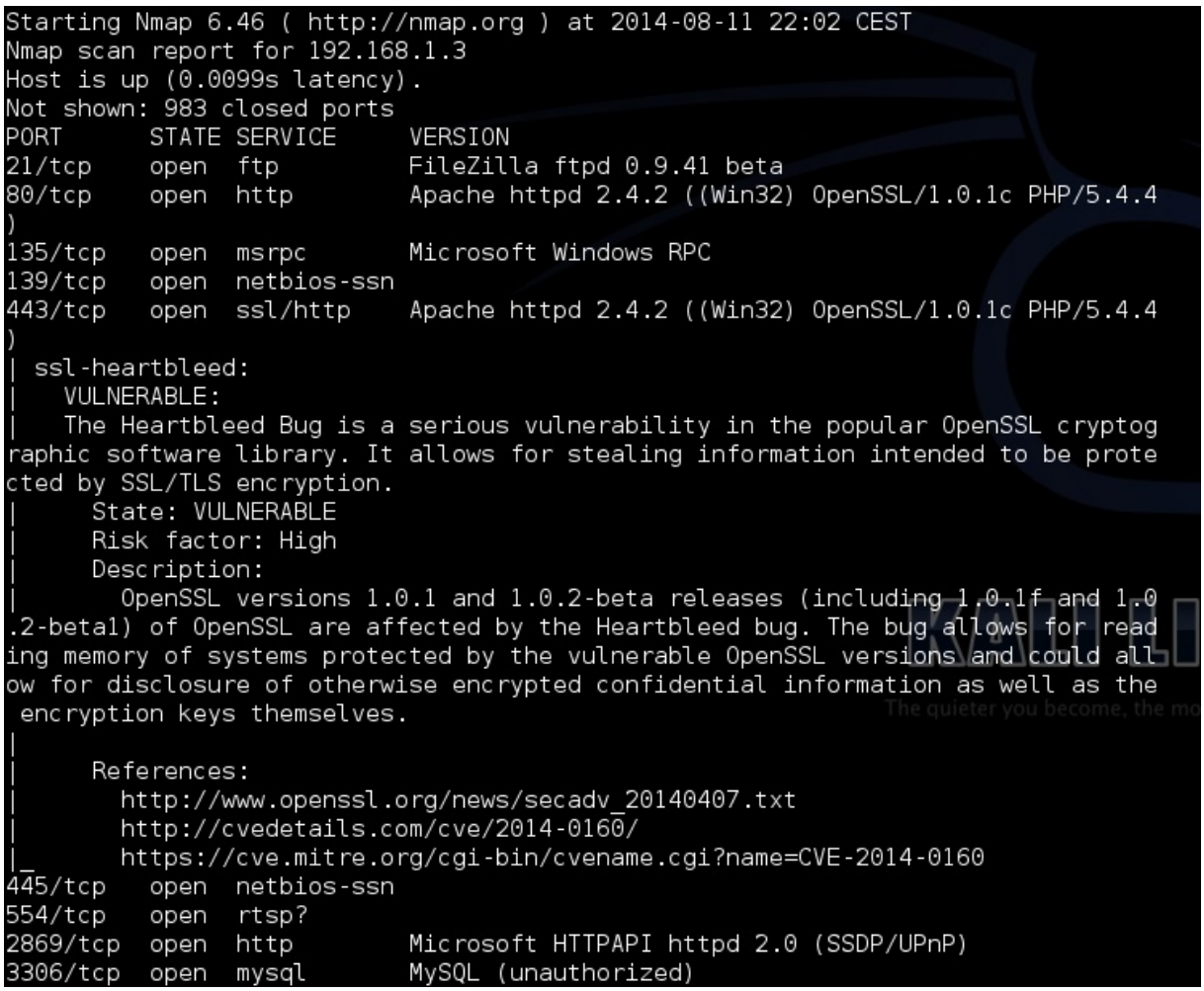
4.4 Prikaz napada

4.4.1 Pregledovanje za ranljivost z uporabo Nmap

Uporabljen je bil ukaz "nmap -sV --script=ssl-heartbleed <target>" v orodju Nmap na prenosnem računalniku na katerem teče Kali Linux.

Slika 17: Ukaz v ukazni lupini na Kali Linux.

Po nekaj sekundah izvajanja je Nmap izpisal, da je na strežniku prisotna ranljivost na strežniku.



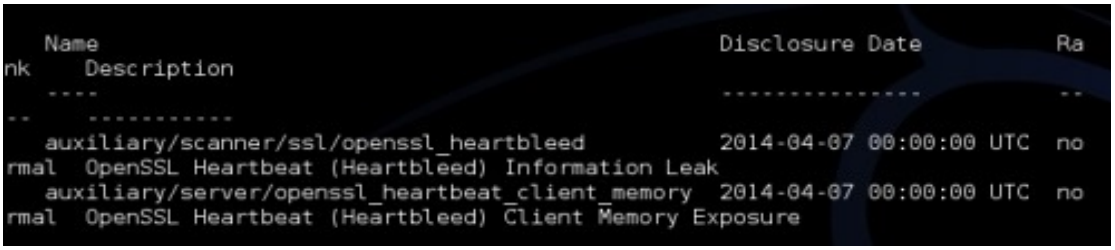
Slika 18: Izpis orodja Nmap v ukazni lupini Kali Linux.

4.4.2 Izvedba z uporabo Metasploit Framework

Ko smo se prepričali, da je ranljivost prisotna, zaženemo Metasploit Framework. To storimo z ukazom "msfconsole". Odpre se nam pozdravni zaslon orodja.

Slika 19: Pozdravni zaslon orodja Metasploit

Nato smo poiskali modul, ki vsebuje kodo za izkoriščanje ranljivosti Heartbleed z ukazom "search heartbleed".



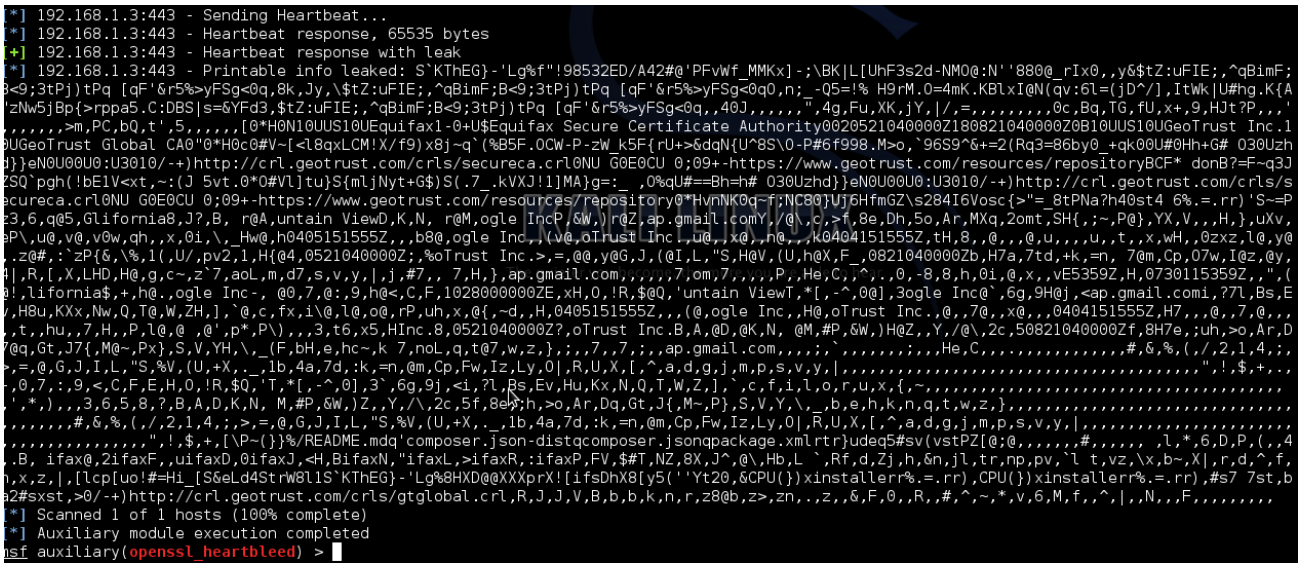
Slika 20: Izpis iskanja modula.

Uporabili smo modul "use auxiliary/scanner/ssl/openssl_heartbleed". Pomoč za uporabo modula je dostopna v dokumentaciji Metasploit Framework ali pa z ukazom "show options". Nastaviti je potrebno še dva parametra. Parameter "verbose" nam omogoči, da se izpisujejo komentarji ob izvedeni operaciji. Parameter "rhost" pa vsebuje naslov IP strežnika na katerem želimo izkoristiti ranljivost.

```
msf auxiliary(openssl_heartbleed) > set verbose true
verbose => true
msf auxiliary(openssl_heartbleed) > set rhosts 192.168.1.3
rhosts => 192.168.1.3
```

Slika 21: Nastavitev parametrov "verbose" in "rhosts".

Nato je potrebno program samo še pognati. To smo storili z ukazom "run".

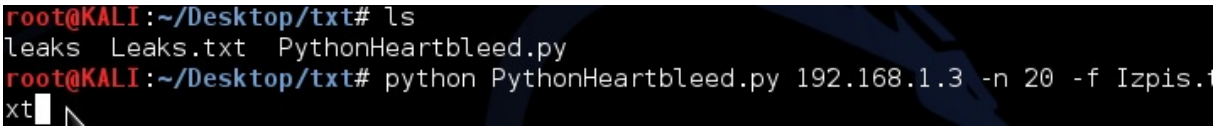


Slika 22: Izpis programa.

Na zaslonu dobimo ACII predstavitev 65535 bajtov, ki jih program vrne iz pomnilnika. Verjetnost, da se v tej omejeni količini podatkov nahajajo občutljivi podatki (na primer geslo), je majhna. Ranljivost Heartbleed pa omogoča, da napad ponavljamo. V ta namen je bila uporabljena skripta, napisana v programskem jeziku Python, ki je opisana v nadaljevanju.

4.4.3 Izvedba z uporabo skripte v programskem jeziku Python

Uporabili smo skripto, napisano v programskem jeziku Python [29], katere avtor je Jared Stafford (jspenguin@jspenguin.org). Avtor si ne lasti nobenih pravic in dodaja, da so pri njenem izpopolnjevanju sodelovali številni drugi avtorji. Skripto smo pognali v ukazni lupini Kali Linux z ukazom "python PythonHeartbleed.py 192.168.1.3 –n 20 –f Izpis.txt". Ukaz "python" lupini pove naj pri izvajanju uporabi tolmač programskega jezikaPython , sledi parameter z imenom skripte, parameter z naslovom IP tarče, parameter "-n" nam omogoča nastavitev koliko paketkov "heartbeat" naj pošlje, parameter "-f" pa določi datoteko kamor bo izpis zapisan. Tako lahko zajamemo večjo količino podatkov in jih kasneje analiziramo.



Slika 23: Ukaz za izvedbo skripte napisane v programskem jeziku Python ki izkorišča ranljivost Heartbleed.

4.5 Kateri podatki so ogroženi

4.5.1 Zasebni ključi

Najbolj nevarna je kraja zasebnega ključa, ki napadalcu omogoča, da dekriptira vsa sporočila, ki so zaščitena z X.509 certifikati. To vključuje tudi sporočila, ki jih je napadalec morebiti zajel v preteklosti. Napadalec se lahko tudi predstavi kot druga oseba na spletu ali nudi storitve v imenu žrtve. Kadar je ukraden zasebni ključ, je potrebno digitalno potrdilo preklicati in naročiti novo. Javni ključ novega potrdila je nato potrebno posredovati vsem udeleženim. Še več škode nastane, če je ključ odtujen avtoriteti, ki je visoko v verigi zaupanja ali celo na vrhu oziroma korenski avtoriteti. [30]

4.5.2 Uporabniška imena in gesla

Pomnilniška slika lahko vsebuje tudi podatke za prijavo uporabnika. To so lahko uporabniška imena, gesla, sejni ključi ali piškotki. Taki podatki lahko omogočijo napadalcu, da se legalno prijavi v sistem. Kadar pride do kraje teh podatkov, je potrebno zamenjati gesla ali pa tudi uporabniška imena in razveljaviti vse sejne ključe in piškotke.

4.5.3 Zaščiteni podatki

Zaščiteni podatki so podatki, ki jih ščiti kriptografija. Z zlorabo ranljivosti Heartbleed jih je mogoče ukrasti ne da bi bila ukradena gesla ali celo zasebni ključi, saj jih napadalec prebere iz pomnilnika v nekriptirani obliki. To so lahko: osebni podatki, finančni podatki, elektronska ali neposredna sporočila, dokumenti ali drugi podatki, ki so bili preneseni preko kriptirane povezave. Ali so bili taki podatki odtujeni, je težko sklepati, saj napad ne pušča posledic v sistemu, razen v primeru, da omrežni administrator zajame napadalčeve paketke "heartbeat", ki vsebujejo občutljive podatke in iz njih razbere kaj je bilo napadalcu dejansko poslano.

4.5.4 Postranski podatki

Postranski podatki zajemajo vse ostale podatke, ki jih lahko napadalec prebere iz pomnilnika. To zajema: pomnilniške naslove, delovne podatke procesov in druge tehnične podatke. Taki podatki imajo za napadalca samo začasno in majhno vrednost. Napadalcu lahko pove na primer kje v pomnilniku se nahajajo podatki določenega procesa. Podatki vrednost izgubijo, ko se prekine seja TCP.

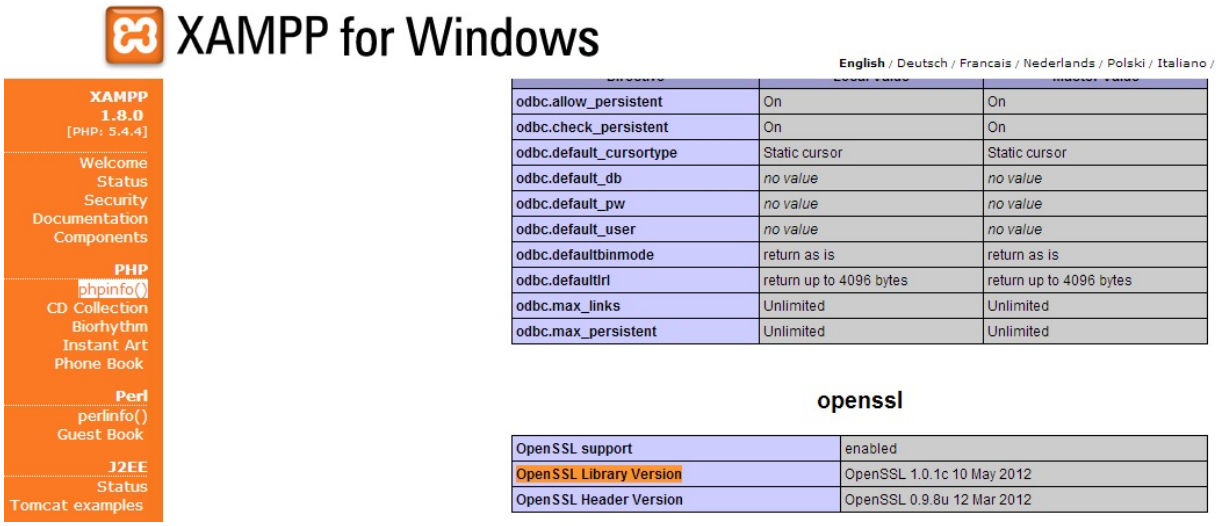
4.6 Odpravljanje ranljivosti

4.6.1 Posodobitev različice OpenSSL

Ranljive so različice OpenSSL 1.0.1 do vključno 1.0.1f. Različica OpenSSL 1.0.1g ni ranljiva. Prav tako niso ranljive različice iz veje OpenSSL 1.0.0 in OpenSSL 0.9.8. [30] Če strežnik uporablja ranljivo različico OpenSSL jo je potrebno posodobiti.

4.6.2 Postopek za posodobitev na paketu strežnikov XAMPP

Da ugotovimo če je naš strežnik ranljiv v brskalnik vpišemo "ip_naslov/xampp/index.php". Nato pritisnemo na zavihek "phpinfo()" in poiščemo "OpenSSL Library Version". Prikaže se nam zaslon, ki vsebuje podatke o knjižnici OpenSSL [31].



Slika 24: Pregled nameščene različice OpenSSL.

Če uporabljamo operacijski sistem Windows je postopek sledeč:

1. Prenesemo namestitveni paket "xampp-opensslfix-win32.zip" s spletne strani www.apachefriends.org
2. Zaustavimo strežnik Apache z uporabo nadzorne plošče XAMPP
3. Razširimo paket "xampp-opensslfix-win32.zip", ki vsebuje datoteke "libeay32.dll", "ssleay.dll" in "openssl.exe"

4. Premaknemo se v mapo "bin", ki se privzeto nahaja v " c:\xampp\apache\bin"
5. Ustvarimo mapo za varnostno kopiranje datotek "libeay32.dll", "ssleay.dll", openssl.exe in jih prenesemo vanjo
6. Datoteke vsebovane v "xampp-opensslfix-win32.zip" razširimo v mapo "bin"
7. Premaknemo se v mapo "php", ki se privzeto nahaja v " c:\xampp\php"
8. Tudi tukaj ustvarimo mapo za varnostno kopiranje kamor premaknemo datoteki

   " libeay32.dll", "ssleay32.dll"
9. Datoteke vsebovane v "xampp-opensslfix-win32.zip" razširimo v mapo "php"
10. Ponovno zaženemo strežnik Apache in preverimo ali je knjižnica OpenSSL posodobljena.

Na operacijskem sistemu Linux moramo najprej preveriti ali uporabljamo 32-bitno ali 64-bitno različico. To storimo z ukazom "file /opt/lampp/bin/openssl". Nato prenesemo ustrezen paket s popravki in sledimo korakom.

1. Ustavimo strežnik Apache z ukazom "sudo /opt/lampp/xampp stopapache"
2. Razširimo vsebino paketa s popravki: "tar -xzf xampp-opensslfix-linux.tar.gz -C /tmp/"
3. Premaknemo se v mapo, ki vsebuje namestitev XAMPP: "cd /opt/lampp/"
4. Ustvarimo mapo za varnostno kopiranje: " sudo mkdir -p /opt/lampp/opensslbackup/"
5. Kopiramo datoteke "libssl", "libcrypto", "engines", "openssl" v prej ustvarjeno mapo:

   "sudo mv /opt/lampp/lib/libssl* /opt/lampp/opensslbackup/"

   "sudo mv /opt/lampp/lib/libcrypto* /opt/lampp/opensslbackup/"

   "sudo mv /opt/lampp/lib/engines /opt/lampp/opensslbackup/"

   "sudo mv /opt/lampp/bin/openssl /opt/lampp/opensslbackup/"

6. Kopiramo vsebino razširjenega paketa s popravki v namestitveno mapo XAMPP:

   "sudo cp -rp /tmp/xampp-opensslfix-linux/lib/* /opt/lampp/lib/"

   "sudo cp -rp /tmp/xampp-opensslfix-linux/bin/openssl /opt/lampp/bin/"

7. Ponovno zaženemo strežnik Apache in preverimo ali je knjižnica OpenSSL posodobljena:

   "sudo /opt/lampp/xampp startapache"

   " sudo /opt/lampp/bin/openssl version".

Po končanem postopku na operacijskem sistemu Windows ali Linux mora biti nameščena različica OpenSSL 1.0.1g ali novejša. V knjižnici je posodobljena metoda, ki pošlje odgovor na zahtevek Heartbeat in vsebuje kodo, ki preverja dolžino prejetih podatkov in vrednost spremenljivke "payload".

```
/* Read type and payload length first */
if (1 + 2 + 16 > s->s3->rrec.length)
        return 0; /* silently discard */
hbtype = *p++;
n2s(p, payload);
if (1 + 2 + payload + 16 > s->s3->rrec.length)
        return 0; /* silently discard per RFC 6520 sec. 4 */
```

Slika 25: Koda, ki odpravlja ranljivost Heartbleed.

Po nadgradnji knjižnice OpenSSL smo ponovno pognali orodje Metasploit, ki nam je izpisalo, da ni nobenih odtekanj podatkov.

```
[*] 192.168.1.3:443 - Sending Heartbeat...
[-] 192.168.1.3:443 - No Heartbeat response...
[-] 192.168.1.3:443 - Looks like there isn't leaked information...
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(openssl_heartbleed) >
```

Slika 26: Izpis orodja Metasploit po nadgradnji knjižnice OpenSSL.

Poglavje 5 Sklepne ugotovitve

V diplomskem delu smo preiskali različne načine omrežnih napadov, opisali njihovo delovanje in predlagali načine za varovanje pred njimi. Pokazali smo, da jih je mogoče razporediti na različne načine in prikazali nekaj razdelitev. Ugotovili smo tudi, da je za izvedbo določene kategorije predpogoj izvedba pregleda in napada s posrednikom.

Nato smo naredili pregled večjih odkritih varnostnih ranljivosti v zadnjih treh letih, pri čemer smo se zaradi njihove pogostosti osredotočili na porazdeljene napade za zavrnitev storitve in napade na varovanje s protokolom SSL/TLS. Ugotovili smo, da so porazdeljeni napadi z zavrnitvijo storitve možni predvsem zaradi nepravilnih nastavitev omrežja in na strežnikih, ki so udeleženi v napadu. Pokazali smo, da napad omogoča ojačenje, ki ga povzročijo dolgi odgovori strežnikov in veliko število udeleženih strežnikov. Da strežnik ne sodeluje v napadu, morajo poskrbeti njegovi skrbniki s pravilnimi nastavitvami in skrbniki omrežja z nastavitvami, ki onemogočajo pošiljanje paketkov s ponarejenimi izvornimi naslovi IP. Kot skrbnik spletne strani lahko nastavimo požarno pregrado tako, da preprečuje prehod zlonamernemu prometu in preverimo varnostne ranljivosti v kodi naše spletne strani. Lahko tudi poiščemo ponudnika, ki je specializiran za nudenje neprekinjene storitve in uredimo gostovanje pri njem.

Napadi na zaščito s protokolom SSL/TLS so bili v zadnjih letih pogosti. Analizirali smo odkrite varnostne ranljivosti in ugotovili, da je osnova za njihovo preprečevanje posodobljena programska oprema in knjižnica OpenSSL. Za dodatno zaščito je potrebno pregledovanje omrežnega prometa, v primeru varnostne ranljivosti Truncation pa napake na strani ponudnika še niso odpravljene, zato je potrebna kontrola fizičnega dostopa do naprave.

Prikazali smo še napad z uporabo varnostne ranljivosti Heartbleed. Heartbleed je bil odkrit v letu 2014 in je bil deležen velike medijske pozornosti in pozornosti strokovnjakov. Učinkovit je, ker za njegovo izvedbo ni potreben napad s posrednikom, napadalec lahko napade ranljivi strežnik neposredno. Ob posameznem odzivu lahko pridobimo 64 kilo bajtov podatkov, vendar je mogoče poslati poljubno veliko število zahtevkov in tako preiskati celoten pomnilnik. Namestili smo operacijski sistem Kali Linux na računalnik, ki je predstavljal odjemalca-napadalca in paket strežnikov XAMPP ter odjemalec elektronske pošte Roundcube na računalnik, ki je predstavljal strežnik- žrtev. Za namen prikaza smo na strežnik namestili ranljivo različico OpenSSL 1.0.1c. Napad smo prikazali z uporabo orodja Metasploit Framework in z uporabo skripte, napisane v programskem jeziku Python. Kljub temu, da smo s strežnika uspešno vrnili podatke, nam ni uspelo prebrati nobenih zasebnih ključev ali

podatkov za prijavo, saj napad vrne naključne podatke. Čeprav je verjetnost, da napad z uporabo ranljivosti Heartbleed uspe v prvem poskusu majhna, se verjetnost, da bo napad uspešen, z vsakim ponovnim poskusom povečuje. Nato smo pripravili še navodila za posodobitev knjižnice OpenSSL v paketu XAMPP na operacijskih sistemih Windows in Linux. Različica OpenSSL, ki vsebuje varnostne popravke za odpravljanje ranljivosti Heatbleed, mora biti OpenSSL 1.0.1g ali višja. Ranljivost je bila prisotna od 14. marca 2012 z uvedbo OpenSSL 1.0.1 in odpravljena 7. aprila 2014. Za sisteme, pri katerih bi lahko prišlo do kraje zasebnih ključev, je bilo potrebno preklicati digitalna potrdila in ustvariti nova. Številne spletne strani so svoje uporabnike pozvale, naj zamenjajo svoja gesla.

Osnova za zavarovanje omrežja in vanj vključenih sistemov je varovanje fizičnega dostopa do naprav, posodobljen operacijski sistem, posodobljena programska oprema in nameščena protivirusna programska oprema. Potrebna je tudi pravilno nastavljena požarna pregrada, ki nadzira promet in tako prepreči preproste omrežne napade. Pomembno je tudi neprestano spremljanje novo odkritih varnostnih ranljivosti, ki jih sporočajo nacionalni odzivni centri in druge organizacije za informacijsko varnost ter poiskati načine za njihovo odpravljanje.

Literatura

[1] F. Ali, „IP Spoofing,“ 12 2007. [Elektronski]. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-4/ipj_10-4.pdf. [Poskus dostopa 1 8 2014].

[2] C. Preimesberger, „www.eweek.com,“ 28 5 2014. [Elektronski]. Available: http://www.eweek.com/security/slideshows/ddos-attack-volume-escalates-as-new-methods-emerge.html. [Poskus dostopa 9 8 2014].

[3] US-CERT, „US CERT Understanding Denial-of-Service Attacks,“ 9 4 2009. [Elektronski]. Available: https://www.us-cert.gov/ncas/tips/ST04-015. [Poskus dostopa 2 9 2014].

[4] L. GANAPATHY, „XSS Attack Examples (Cross-Site Scripting Attacks),“ 16 2 2012. [Elektronski]. Available: http://www.thegeekstuff.com/2012/02/xss-attack-examples/. [Poskus dostopa 5 8 2014].

[5] Committee on National Security Systems, „National Information Assurance (IA) Glossary,“ 26 4 2010. [Elektronski]. Available: http://www.ncix.gov/publications/policy/docs/CNSSI_4009.pdf. [Poskus dostopa 9 7 2014].

[6] G. Božič, „Poročilo o omrežni varnosti 2012 - SI-CERT,“ 8 4 2013. [Elektronski]. Available: https://www.cert.si/fileadmin/slike/si-cert/fokus/2013/SI-CERT_porocilo_2012.pdf. [Poskus dostopa 15 7 2014].

[7] ISO/IEC, „Information technology Open Systems Interconnection Basic Reference Model: The Basic Model,“ 21 6 2000. [Elektronski]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=20269. [Poskus dostopa 27 7 2014].

[8] P. Phatak, „Cyber Attacks Explained: Network Sniffing,“ 30 1 2012. [Elektronski]. Available: http://www.opensourceforu.com/2012/01/cyber-attacks-explained-network-

sniffing/. [Poskus dostopa 28 7 2014].

[9] US-CERT, „US-CERT DDoS by OSI layer,“ 29 1 2014. [Elektronski]. Available: http://www.us-cert.gov/sites/default/files/publications/DDoS%20Quick%20Guide.pdf. [Poskus dostopa 28 7 2014].

[10] SI-CERT, „SICERT o Centru,“ 8 5 2009. [Elektronski]. Available: https://www.cert.si/en/detailed-information-rfc-2350/. [Poskus dostopa 6 8 2014].

[11] D. .. Mills, „Network Time Protocol (NTP),“ 9 1985. [Elektronski]. Available: http://tools.ietf.org/html/rfc958. [Poskus dostopa 11 8 2014].

[12] J. Graham-Cumming, „Understanding and mitigating NTP-based DDoS attacks,“ 9 1 2014. [Elektronski]. Available: http://blog.cloudflare.com/understanding-and-mitigating-ntp-based-ddos-attacks. [Poskus dostopa 12 8 2014].

[13] Team Cymru, Alan Amesbury,Barry Greene,Anthony Maszeroski,Donald Smith, „SECURE NTP TEMPLATE,“ 2014. [Elektronski]. Available: http://www.team-cymru.org/ReadingRoom/Templates/secure-ntp-template.html. [Poskus dostopa 13 8 2014].

[14] D. S. P. Ferguson, „Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,“ 5 2000. [Elektronski]. Available: http://tools.ietf.org/html/bcp38. [Poskus dostopa 13 8 2014].

[15] G. Božič, „SI CERT vrnitev odpisanih,“ 19 10 2012. [Elektronski]. Available: https://www.cert.si/vrnitev-odpisanih/. [Poskus dostopa 13 8 2014].

[16] P. Mockapetris, „RFC 1035 Domain Implementation and Specification,“ 11 1987. [Elektronski]. Available: http://tools.ietf.org/html/rfc1035#page-12. [Poskus dostopa 13 8 2014].

[17] D. Goodin, „Arstechnica,“ 13 9 2012. [Elektronski]. Available: http://arstechnica.com/security/2012/09/crime-hijacks-https-sessions/. [Poskus dostopa 3 9 2014].

[18] R. P. M. Belshe Twist, „The Chrominium Projects,“ Google, Inc, 2 2012. [Elektronski]. Available: https://tools.ietf.org/html/draft-mbelshe-httpbis-spdy-00. [Poskus dostopa 2 9 2014].

[19] D. Goodin, „Crack in Internet's foundation of trust allows HTTPS session hijacking," 13 9 2012. [Elektronski]. Available: http://arstechnica.com/security/2012/09/crime-hijacks-https-sessions/. [Poskus dostopa 19 8 2014].

[20] R. Fielding, UC Irvine, J. Gettys, J. C. Mogul, Compaq, H. Frystyk, W3C/MIT, L. Masinter, P. Leach, T. Berners-Lee, Microsoft, „RFC2616," [Elektronski]. Available: http://tools.ietf.org/html/rfc2616. [Poskus dostopa 20 8 2014].

[21] N. H. A. A. (. P. YOEL GLUCK, „SSL, GONE IN 30 SECONDS," [Elektronski]. Available: http://breachattack.com/. [Poskus dostopa 20 8 2014].

[22] N. H. A. A. (. P. YOEL GLUCK, „BREACH: REVIVING THE CRIME ATTACK," 12 7 2013. [Elektronski]. Available: http://breachattack.com/resources/BREACH%20-%20SSL,%20gone%20in%2030%20seconds.pdf. [Poskus dostopa 3 9 2014].

[23] I. Ristic, Apache Security, Sebastopol: O'Reilly Media, 2005.

[24] A. P. Ben Smyth, „Truncating TLS Connections to Violate Beliefs in Web Applications," 1 8 2013. [Elektronski]. Available: https://media.blackhat.com/us-13/US-13-Smyth-Truncating-TLS-Connections-to-Violate-Beliefs-in-Web-Applications-WP.pdf. [Poskus dostopa 20 8 2014].

[25] P. P. a. P. Krishnamurthy, „sl.wikipedia.org/wiki/RC4," 2003. [Elektronski]. Available: http://web.archive.org/web/20131203082918/http://www.sis.pitt.edu/~is3966/group5_paper2.pdf. [Poskus dostopa 20 8 2014].

[26] D. J. B. K. G. P. B. P. J. C. N. S. Nadhem J. AlFardan1, „On the Security of RC4 in TLS and WPA," 8 7 2013. [Elektronski]. Available: http://www.isg.rhul.ac.uk/tls/RC4biases.pdf. [Poskus dostopa 20 8 2014].

[27] Netcraft, „April 2014 Web Server Survey," 2 5 2014. [Elektronski]. Available: http://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html. [Poskus dostopa 25 8 2014].

[28] M. TEAM, „The Heartbleed Hit List: The Passwords You Need to Change Right Now," 10 4 2014. [Elektronski]. Available: http://mashable.com/2014/04/09/heartbleed-bug-websites-affected/. [Poskus dostopa 25 8 2014].

[29] a. Staffor, „Skripta Python za prikaz napada z ranljivostjo Heartbleed,“ [Elektronski]. Available: https://gist.github.com/sh1n0b1/10100394. [Poskus dostopa 20 8 2014].

[30] C. Ltd., „The Heartbleed Bug,“ 29 4 2014. [Elektronski]. Available: http://heartbleed.com/. [Poskus dostopa 2014 8 2].

[31] Apache Friends, „Heartbleed OpenSSL Bug,“ 2014. [Elektronski]. Available: https://www.apachefriends.org/blog/heartbleed-bug.html#linuxfiles. [Poskus dostopa 20 8 2014].

[32] M. Cantoni, „Sledenje DNS DDOS,“ [Elektronski]. Available: http://www.nothink.org/honeypot_dns.php. [Poskus dostopa 13 8 2014].

Seznam slik

Seznam tabel