TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduste instituut

Informaatika eriala

Erki Vaino

Teenusetõkestusründed ja kaitse lahendused

Bakalaureusetöö (6 EAP)

Juhendaja: Meelis Roos

Autor: ……... „……“ mai 2013

Juhendaja: ……... „……“ mai 2013

Lubada kaitsmisele

Professor: ……... „……“ mai 2013

TARTU 2013

Sisukord

Sissejuhatus

Viimastel aastatel on teenusetõkestusründed muutunud üha tavalisemaks nähtuseks. Suurimad ründed on ületanud juba 100 Gbps piiri ja on igapäevased nähtused. Lisaks kasutatakse ründeid, mis ei vaja kahju tekitamiseks suurt mahtu ja mida saab teostada tavalise sülearvutiga.

Tegemist on referatiivse tööga, kus kasutatakse ründe ja kaitsemeetodite jaotamiseks raamatut „Network Security“ [2] ning ründe täpsemaks kirjeldamiseks erinevaid materjale, mis on Internetis vabalt kättesaadavad.

Töö on suunatud Tartu Ülikooli matemaatika-informaatika tudengitele ja inimestele, kes soovivad laiendada oma silmaringi teenusetõkestusrünnete osas. Töö eesmärk on eesti keeles tuua välja erinevate rünnete liigid ja nende kasutamine. Lisaks ka erinevad meetodid, mida saab kasutada kaitsmiseks ning milliseid tooteid on loodud erinevate firmade poolt rünnete mõju vähendamiseks.

Esimeses peatükis on kirjeldatud teenusetõkestusründeid üldiselt, kuidas neid saab jaotada ja üldised tunnusjooned. Järgnevas neljas peatükis on kirjeldatud erinevaid ründeid vastavalt sellele, millist nõrkust nad efekti saavutamiseks ära kasutavad. Viimases peatükis kirjeldatakse meetodeid, kuidas rünnete eest on võimalik kaitsta. Välja on toodud ka erinevate firmade tarkvaralised ja riistvaralised lahendused teenusetõkestusrünnete peatamiseks.

1. Teenusetõkestusrünnetest üldiselt

Teenusetõkestusrünneteks loetakse tegevust, kus pahatahtlik kasutaja sihilikult blokeerib arvutisüsteemi või võrgu ressursse niimoodi, et teised kasutajad ei saa neid kasutada. Sellised ründed on muutunud Internetis väga populaarseks ja neid kasutatakse iga päev pankade, firmade ja riigiasutuste vastu.

Ründe idee seisneb selles, et tarvitatakse ära erinevad ressursid, näiteks protsessori jõudlus, vahemälu, võrgu läbilaskevõime ning veebiserveri ühendused, nii et teised kasutajad ei saa enam ohvri pakutavaid teenuseid kasutada.

Peamised ründajate motiivid on raha väljanõudmine, poliitiline vastuseis ja *online* protesteerimine. Teenusetõkestusründed on muutnud tavaliseks mooduseks, mille abil raha välja pressida. Organisatsioonile saadetakse kiri, milles öeldakse, et kui nad ei kanna raha ründaja kontole, siis võetakse firma pakutav teenus maha.

2007. aastal toimunud Pronksiöö tagajärjel sattusid Eesti riigiasutuste, pankade ja uudisteportaalide leheküljed teenusetõkestusrünnete alla. Selle tulemusena oli nende veebilehtede külastamine häiritud.

Lisaks on viimastel aastatel levima hakanud ka *online* protesteerimine. Rühmitus Anonymous on läbi viinud suuri ründeid selliste firmade vastu nagu MasterCard, PayPal, Visa ja Amazon. Nad protesteerisid selle vastu, et antud organisatsioonid lõpetasid WikiLeaksi toetamise. Rünnetes osalesid mitmed tuhanded inimesed vabatahtlikult, väljendades sellega oma meelepaha.

Teenusetõkestusrünnete alla kuuluvad hajusad teenusetõkestusründed (*Distributed Denial of Dervice*) [1]. See ründe liik on tänapäeval väga aktuaalne, kuna võib põhjustada suuri kahjusid võrkudele ja organisatsioonidele. Üks võimalus hajusaks teenusetõkestusründeks on kasutada robotvõrke, mis koosnevad mitmetest tuhandetest arvutitest. Teine lahendus on rünnete võimendamiseks kasutada erineva nõrkusega võrke ja süsteeme.

1.1 Rünnete klassifitseerimine

Teenusetõkestusründeid saab jaotada vastavalt rünnaku protokolli tasemele.

- Võrguseadme tasemel ründed tarbivad ära võrguseadme vabad ressursid või kasutavad ära vigu seadme tarkvaras.
- Operatsioonisüsteemi tasemel ründed kasutavad ära selle, kuidas süsteemid protokolle realiseerivad, näiteks Ping of Death.
- Rakenduse tasemel ründed kasutavad ära nõrkusi rakendustes, põhjustades ohvri ressursside väärkasutamist. Teine võimalus on leida suure algoritmilise keerukusega viga rakendusest.
- Andmete ummistusrünnete korral saadetakse võimalikult palju andmeid ohvri võrku, kasutades sellega ära vaba ribalaiuse. Näiteks Smurf ja Fraggle rünne.
- Protokolli omaduse rünnete korral kasutatakse ära kindlat protokolli omadust. IP võltsimine on üks näide sellest.

Hajusaid teenusetõkestusründeid saab jaotada ründe intensiivsuse järgi [2]:

- Pideva vooga rünne
- Muutuva vooga rünne
    - Kasvavad
    - Kõikuvad

Pideva vooga ründe puhul kasutab ründaja kõiki oma ressursse, et tekitada koheselt võimalikult palju kahju. Kasvava võimsusega ründe puhul alustatakse aeglaselt, püüdes jääda märkamatuks ja aja jooksul suurendatakse võimsust, kasutades ära kõik vabad ressursid.

Kõikuva võimusega rünnete puhul muudab ründaja võimsust vastavalt sellele, kuidas ohver reageerib. Võimsust vähendades, soovides nii jääda märkamatuks ja võimsust suurendades, et tarbida võimalikult palju vabu ressursse.

Võib ka jaotada vastavalt sellele, kuidas ründaja kontrollib ründeseadmeid hajusate rünnete korral.

- Manuaalne
- Poolautomaatne
    - Otsene
    - Kaudne
- Täisautomaatne

Varem pidi ründaja otsima endale sobivad masinad, murdma neisse sisse ja seadistama manuaalselt ründekoodi. Poolautomaatsete rünnete korral on ründajal olemas vahelüli tema ja rünnet teostavate seadmete vahel. Otsese suhtluse korral peavad ründaja ja vahelüli üksteist teadma. Selleks on tavaliselt vahelülidel teada ründaja IP. Peamine probleem sellega on see, et kui tuvastatakse vahelüli, siis saab ka ründajat leida. Kaudse suhtluse korral on ründaja tuvastamine keerulisem, sest otsest suhtlust ei toimu. Üks näide sellest on IRC serverite kasutamine, et kontrollida rünnakuid. Täisautomaatse ründe korral ei pea olema suhtlust ründaja ja ründeseadme vahel, mis vähendab riski vahele jääda. Ründe meetod, kestvus ja ohver seadistatakse eelnevalt ja enamasti on tegemist ühe käsuga. Sellised ründed on aga üsnagi piiratud.

Selle järgi, milline on ründe mõju ohvrile saab ründe jagada kaheks. Esimene on täielik segav mõju, mille tulemusena tekib ohvril koheselt teenusetõkestus. Teine võimalus on häirida teenuste pakkumist aeglasemalt, vältides sellega kohest ründe avastamist ohvri poolt.

Viimane rünnete jaotamise viis on vastavalt sellele, millist nõrkust ära kasutatakse [2]:

- Vigaste pakettide ründed
- Ummistusründed
- Võimendusründed
- Ründed, mis kasutavad ära nõrkusi protokollis

Vigaste pakettide rünnete korral valmistab ründaja spetsiaalseid pakette, mis põhjustavad ohvrisüsteemide hangumist ja kokku jooksmist. Ummistusrünnete korral saadetakse ohvrile võimalikult palju andmeid, nii et kasutatakse ära kogu vaba ribalaius. Võimendusrünnete korral kasutatakse peegeldajaid, et võimendada rünnakut kordades suuremaks ja viimane jaotus on ründed, mis kasutavad nõrkusi protokollides.

Selle jaotumise järgi on antud töö üles ehitatud, igas peatükis on kirjas vastavad ründed.

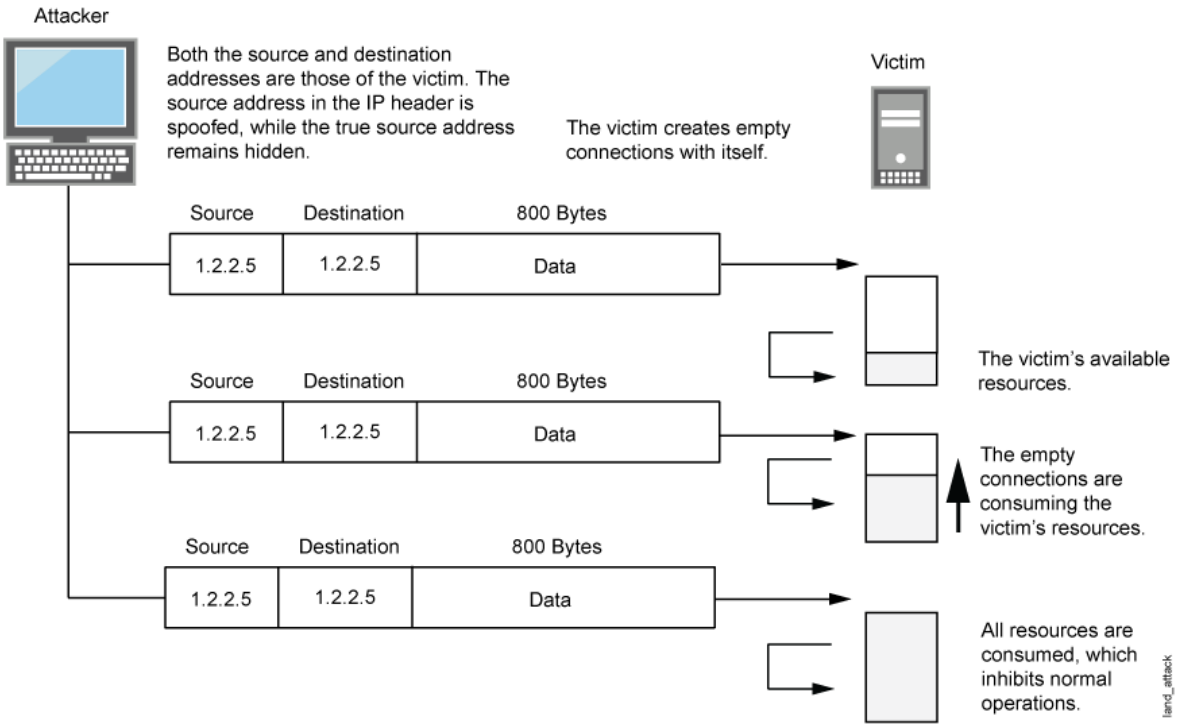
2. Vigaste pakettide ründed

Sellised ründed põhinevad ideel, et ründaja genereerib vigase paketi, mis saadetakse ohvrile ja mis põhjustab süsteemide ebanormaalset käitumist. Teenusetõkestusrünne tekib näiteks siis, kui arvuti või server hangub või teeb taaskäivituse. LAND (*Local Area Network Denial*) ja *Christmas tree* rünnete puhul tekitab teenusetõkestuse see, millised andmed on paketis.

Teine suurem variatsioon vigaste pakettide rünnetest kasutab ära nõrkust fragmenteeritud pakettide kokku panemisel. Kuna üks osa võrguseadmeid ei saa suurte pakettide käsitlemisega hakkama, siis jaotatakse pakettides olev info väiksematesse pakettidesse ja saadetakse üle võrgu. Sihtpunktis olev seade võtab need paketid vastu ja paneb andmed uuesti kokku ning annab edasi kõrgema kihi rakendusele. Ründaja saadab modifitseeritud pakette, mis võivad süsteemi kokku jooksutada ning põhjustada süsteemi mitte tavapärast käitumist. Selle tagajärjel tekibki teenusetõkestus, sest süsteemid ei saa teenindada teisi kasutajaid.

2.1 LAND rünne

Tuntud ründemeetod, kus ründaja saadab ohvri masinale eriliselt loodud TCP SYN paketi. IP lähteaadress võltsitakse ja määratakse samaks, mis on rünnatava masina aadress. Selle tulemusena hakkab masin iseendale vastama ja tekib lõpmatu tsükkel, mis kasutab ära kogu protsessori jõudluse. Tänapäeval on kõigil operatsioonisüsteemidel olemas turvapaik selle nõrkuse vastu. Lisaks viskavad marsruuterid ja jagajad sellised paketid kohe minema ega lase neil võrgus edasi liikuda [3][4][5][6].



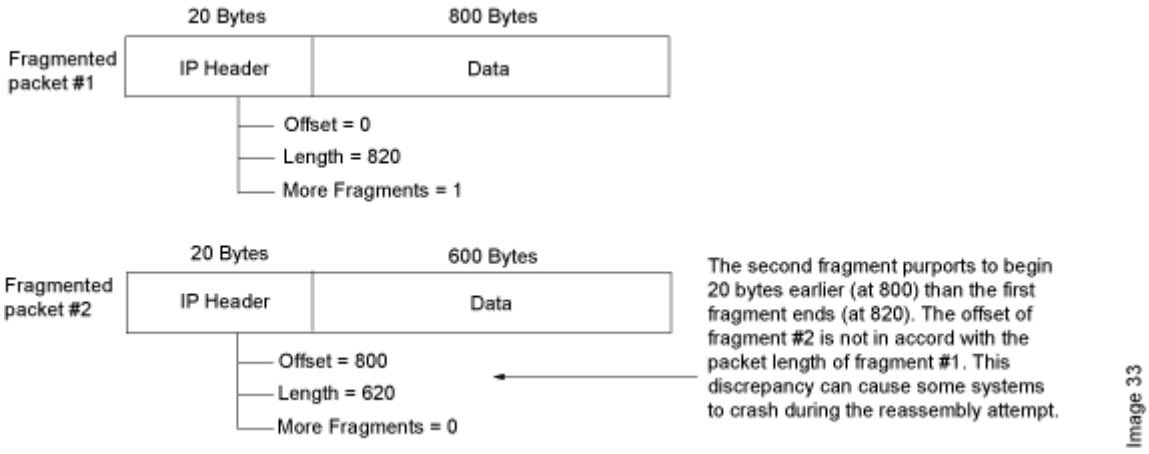
Joonis 1: LAND ründe seletus [4]

2.2 Christmas tree ründe

Ründaja saadab ohvrile pakette, mille kõikvõimalikud protokolli lipud on määratud tõeseks, näiteks FIN, PSH ja URG [5]. Kuna paljud operatsioonisüsteemid reageerivad sellistele pakettidele erinevalt, siis kasutatakse seda rünnet tuvastamaks, milline on rünnatav süsteem. Kuna aga selliste pakettide protsessimine nõuab palju ressursse, siis saab kasutada seda ka kui teenusetõkestusrünnet [7].

2.3 Teardrop ründe

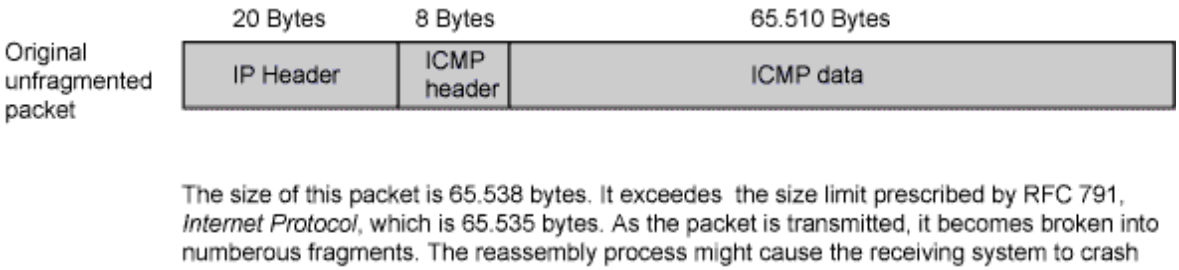
Teardrop on rünne, kus ründaja moodustab sellised fragmenteeritud paketid, milles olev info kattub. Näiteks esimene pakett ütleb, et temas olev info jätkub keset teist paketti ja teine pakett ütleb, et temas olev info algas juba esimeses pakettis. Vanemates operatsioonisüsteemides, mis kasutasid veel koodiveaga TCP/IP fragmentatsioonide kokkupanemist, tekitavad sellised pakettid segadust ja tihti jooksid operatsioonisüsteemid kokku või tegid iseseisva taaskäivituse [8][9].



Joonis 2: Fragemneteeritud pakettide kattumise rünne [8]

2.4 Ping of Death

RFC 791 (*Internet protocol*) määrab selle, et suurim IPv4 pakett võib olla 65535 baiti. IP paketi päis on 20 baiti ja ICMP *echo request* 8 baiti pikk. Seega võib ICMP *echo* paketi kehas olla 65507 baiti andmeid. Ründaja aga loob ICMP paketti, kus on lubatust rohkem andmeid. Selliseid fragmenteerituid pakette kokku pannes hangusid paljud vanad operatsioonisüsteemid. See rünne on tuntud ning tänapäeval viskavad operatsioonisüsteemid sellised paketid minema. Selliste pakettide saatmine viitab ründele ning tulemüürid kirjutavad logifailidesse, et toimub kahtlane võrguliiklus [10][11].



Joonis 3: Näide Ping of Death paketist [11]

2.5 ROSE rünne

Selle ründe puhul saadetakse fragmenteeritud paketti paar esimest ja viimast baiti. Ohvri arvuti puhver jääb ootama andmeid paketti keskelt, aga neid tegelikult kunagi ei saadetagi. Kui selliseid väikseid pakette saadetakse piisavalt palju, siis fragmentidele eraldatud mälu täitub ning see ei saa enam uusi pakette vastu võtta ja töödelda. Lisaks võib rünne ära kasutada kogu vaba ribalaiuse [12].

2.6 New Dawn

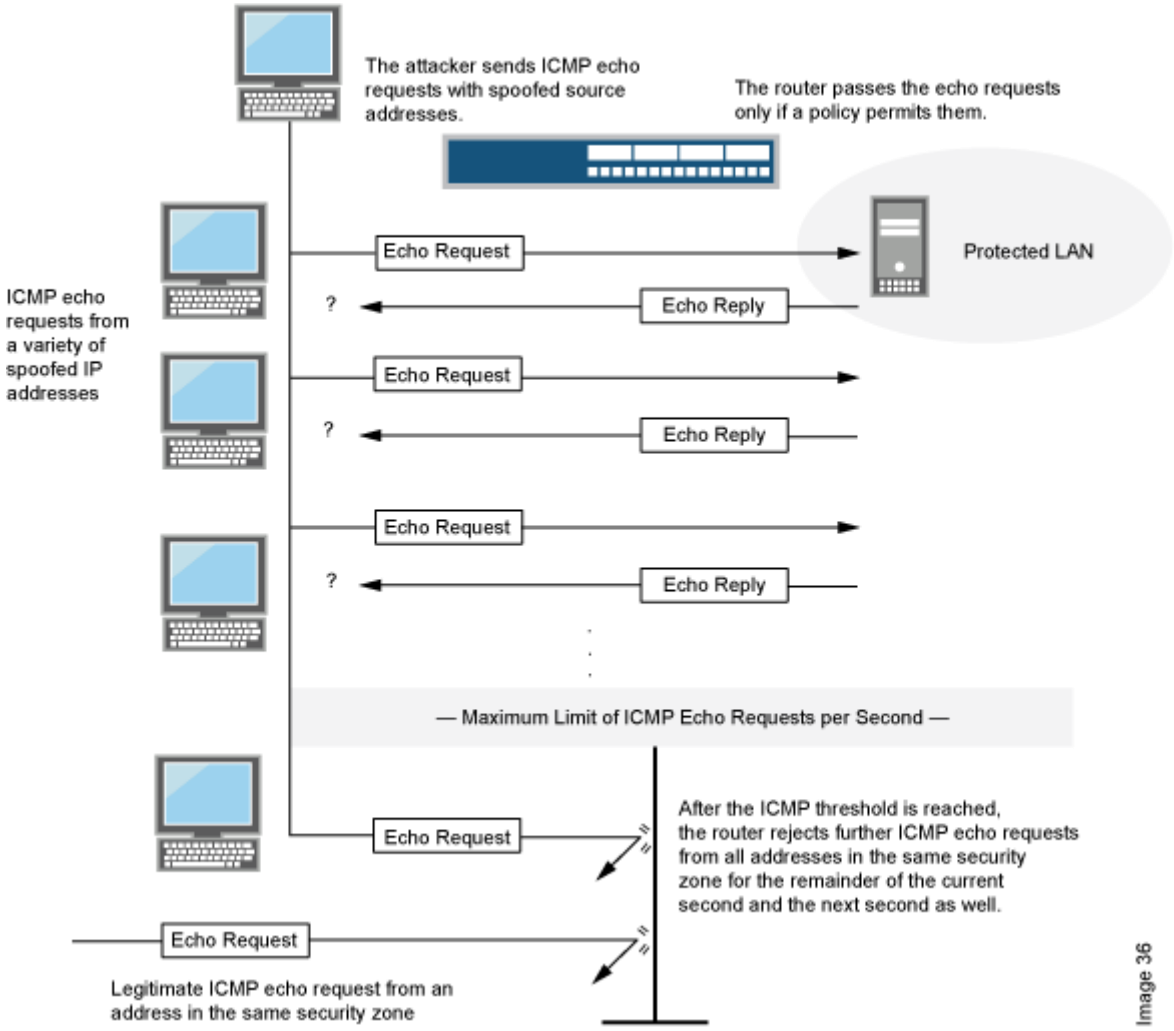
Rose ründe edasiarendus. Algul saadetakse fragmendi alguse pakett ja siis hakatakse saatma järjest väikseid osasid, aga vahelt jäetakse mõned saatmata. Terve pakett ei jõuagi kunagi kohale. Lõpuks pannaks korduvalt teele fragmendi viimast paketti. Selle peale üritab ohvri arvuti protsessor korduvalt sõnumit kokku panna, aga ei saa, sest osa pakette ei saadetud kohale [12].

3. Ummistusründed

Ummistusrünnete puhul saadetakse ohvrile väga palju võrguliiklust, mille läbitöötamine nõuab ressursse. Osad sellised ründed ummistavad sidekanali ja tekib suur paketikadu, mistõttu ei pääse läbi ka õigete kasutajate andmevood. Osad ründed aga koormavad sihtarvutit ennast, segades selle tööd. Kõiki selliseid ründeid saab robotivõrku kasutades muuta efektiivsemaks.

3.1 Ping paketide ummistus

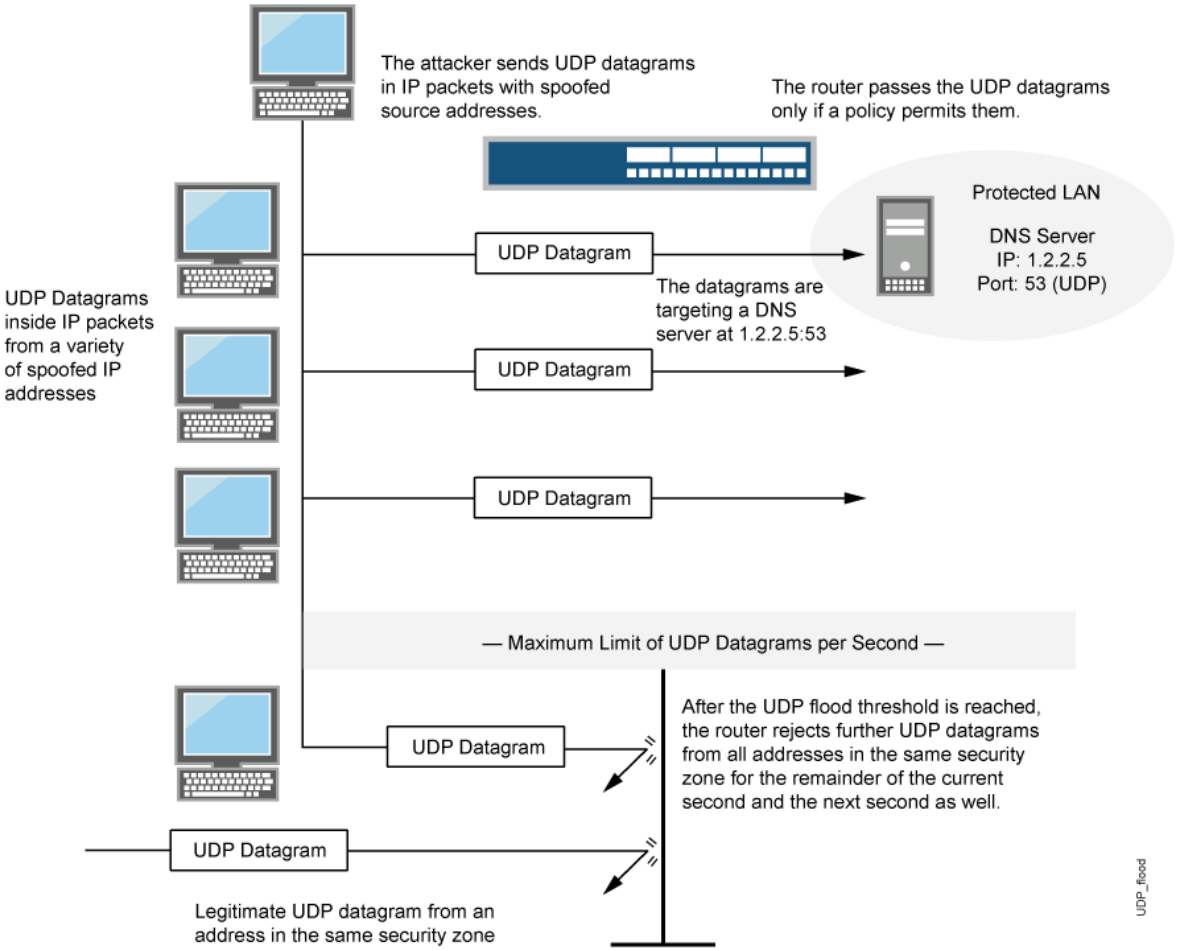
Ping kasutab ICMP (*Internet Control Message Protocol*) protokolli, et kontrollida, kas võrgus olevad seadmed on kättesaadavad. Sihtkohas olevale masinale saadetakse ICMP *echo request*, mille peale vastatakse *echo reply*. Rünnete ajal saadab ründaja võimalikult kiiresti väga palju ping pakette. Ohver tavaliselt vastab kõigile neile, kasutades selleks oma protsessorit ja ka üleslaadimise kiirust. Kuna ICMP ei loo ühendust kahe arvuti vahel, siis võib ründaja võltsida IP tagastusaadressi [13].



Joonis 4: Ping pakettide ummistus [13]

3.2 UDP ummistus

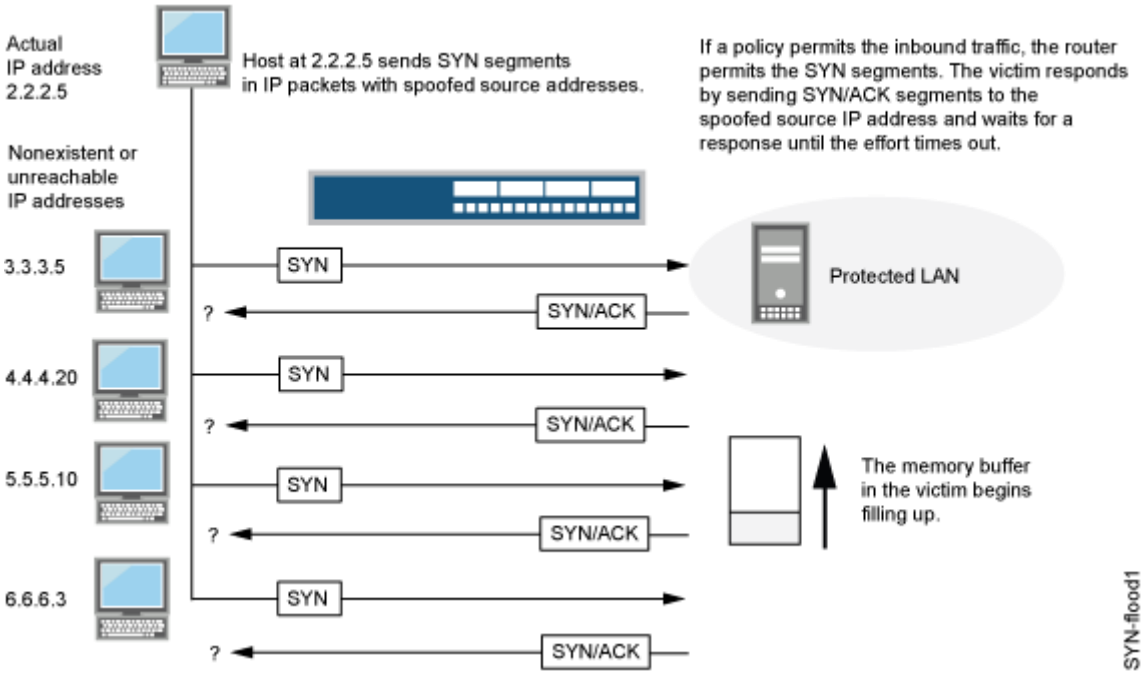
UDP (*User Datagram Protocol*) uputusründega on tegemist siis, kui ohvrisüsteemile saadetakse palju UDP pakette. Sellega kasutatakse ära vaba ribalaius ning tekkib võrguliikluse küllastus. Lisaks peab ohver genereerima ICMP paketi teatega „*destination unreachable*“ ja selle tagasi saatma. Teenuse tõkestus tekib siis, kui ohvrimasin ei suuda enam teenindada teisi kasutajaid, sest on hõivatud ICMP pakettide genereerimise ja saatmisega. Ründaja saab ründe ajal anonüümseks jääda, sest UDP ei loo ühendust kahe arvuti vahel ja seega saab võltsida lähteaadressi, kust paketid pärinevad [14][15].



Joonis 5: UDP ummistus ründe seletus [15]

3.3 SYN ummistus

Kahe arvuti vahelise ühenduse loomiseks kasutatakse TCP *three-way-handshake*'i. Klient saadab näiteks serverile SYN paketti, server vastab kliendile saates SYN-ACK paketi ja jääb vastust ootama. Peale seda, kui klient vastab omaltpoolt ACK paketiga avatakse täisühendus kahe masina vahel ja hakatakse üksteisele saatma andmeid. Ründaja aga ei saada kunagi tagasi viimast ACK paketti, mille tulemusena hoiab server ühendust mõnda aega poolavatud seisundis. Kui ründaja suudab luua poolavatuid ühendusi kiiremini, kui need jõuavad aeguda, siis tekkib teenusetõkestusrünne, sest uusi ühendusi teistelt kasutajatelt ei võeta enam vastu. Tänapäeval kasutavad mitmed operatsioonisüsteemid SYN *cookies* mehhanismi, mis paneb SYN paketiga kaasa ühenduse info ja kui tegemist on päris ühenduse algatusega, siis saavad SYN-ACK vastusega selle info tagasi [16][17].



Joonis 6: SYN ummistus ründe seletus [17]

3.5 RA ummistus

Router Advertisement nõrkus, mis toimib küll ainult OSI-mudeli teise kihi kohtvõrgu piires. Kuna IPv6 toetab väga suurt aadresside hulka, siis saab kasutada seda teenusetõkestusründeks. Ründaja genereerib palju RA pakette erinevate MAC aadresside ja IPv6 eesliidetega. Arvutid, millel on automaatne olekuseisundi seadistamine lubatud, hakkavad IPv6 eesliiteid välja arvutama ja oma marsruuditabeleid uuendama. See põhjustab omakorda protsessori 100 %-list kasutamist, mille tagajärjel süsteemid enam ei toimi ja vajavad enamasti taaskäivitamist [18][19].

4. Võimendusründed

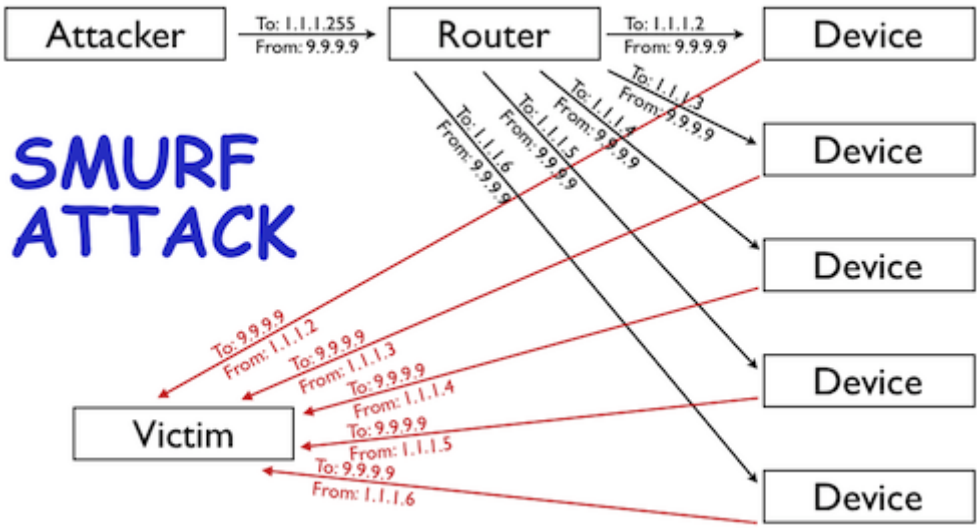
Tavaliste ummistusrünnete korral üritab ründaja saata piisavalt võrguliiklust, et tekitada teenusetõkestusrünne. Tänapäeval aga suudavad enamus arvutivõrkude kaitsesüsteeme sellised ründed kas lihtsalt peatada või ära taluda ning nüüd on kasutusele võetud võimendusründed. Vahelülisid ründaja ja ohvri vahel nimetatakse peegeldajateks, paketi saamisel nad vastavad suurema hulga paketiga. Veebiserverid, DNS-serverid ja marsruuterid on peegeldajad, sest peale SYN või teiste TCP pakettide saamist, vastavad nad SYN-ACK või *Reset connection* (RST) paketiga.

Klassikaliste võimendusrünnete puhul kasutab ründaja ära marsruuteritel olevat IP leviedastusaadressi võimalust. Saates paketi marsruuteri leviedastusaadressile, edastavad nad selle edasi kõigile seadmetele, mis kuuluvad samasse võrku.

Ründe ajal saadab ründaja peegeldajatele vastustnõudvaid võltsitud pakette. Pakettide lähteaadress võltsitakse ja asendatakse ohvri aadressiga. Pärast pakettide saamist vastavad peegeldajad ja saadavad paketid seejärel ohvrile edasi. Ohvri seisukohalt on ründe teostajateks peegeldajad, sest neilt pärineb andmevoog. Peegeldajatele jääb mulje, et ohver ründab neid, sest päringud tulevad ohvri IP-aadressiga.

4.1 Smurf rünne

Smurf rünne on tavalise ping ummistusründe edasiarendus, kus ründaja saadab ICMP *echo request* pakette võltsitud ohvri lähteaadressiga võrgu marsruuteri leviedastusaadressile, mis siis omakorda edastab need paketid kõigile seadmetele selles võrgus. Seadmed vastavad ping päringule ja saadavad ohvrile vastused. Tänapäeval on selliseid ründeid raske teostada, sest enamus marsruutereid on seadistatud mitte edastama ICMP päringuid leviedastusaadressil [1].



Joonis 7: Smurf rünne [22]

4.2 Fraggle rünne

UDP ummistusründe edasiarendus, kus ründaja saadab peegeldajale palju UDP pakette. Marsruuter edastab pakettid edasi oma sisevõrku ja seal olevad masinad vastavad ICMP *destination unreachable* paketiga ründaja võltsitud läheaadressile ehk siis ohvrile [20].

4.3 SMTP rünne

Ründaja saadab Internetis asuvale halvasti seadistatud STMP (*Simple Mail Transfer Protocol*) serverile ettevalmistatud meili. Server võtab kirja vastu ja tuvastab, et sellise kasutajanimega kirja vastuvõtjaid ei ole. Iga CC: ja BCC: päistes olev kehtetu kasutaja kohta genereeritakse NDN (*non-delivery notification*) sõnum ehk *bounce* ja saadetakse see kirja lähtekohta tagasi. Kuna aga lähteaadress on ründaja poolt võltsitud, siis saadetakse veateated ohvri SMTP serverile. Olenevalt sellest kuidas vahelüli moodustab NDN sõnumi, võivad edastatavad kirjad olla väga suured, omades näiteks originaalsõnumit, kirjale kaasa pandud lisasid ja SMTP serveri omaveateadet. NDN sõnumi moodustamise kohta ei ole kindlat protokolli, nii et iga SMTP server moodustab selle nii, nagu see on seadistatud [21].

Tänapäeval korralikud seadistatud SMTP serverid ei võta selliseid kirju enam vastu ja annavad veateate juba SMTP seansi ajal.

4.4 DNS ummistusrünne.

Selle ründe peegeldajateks on DNS (*Domain Name System*) serverid. Ründaja teeb päringu DNS serverile ohvri aadressiga, server genereerib vastuse ja saadab selle ohvrile. Ründaja

üritab küsida DNS serverilt võimalikult palju andmeid, et võimendusefekt oleks võimalikult suur. Kuna ründaja päringu pakett on väiksem kui DNS serveri vastus, siis saab ründaja väikse vaevaga väga efektiivse tulemuse. Probleemiks on avatud DNS serverid, mis on halvasti konfigureeritud ja vastavad kõigile päringutele. Kui DNS server toetab DNSSEC signatuure ja ründaja neid küsib, siis saab rünnakut veel suuremaks võimendada, sest vastusepaketid on DNSSEC signatuuride ja linkimisinfo võrra suuremad [1][22][23].

5. Ründed protokolli nõrkuste pihta

5.1 SSL ründed

SSL/TLS on protokoll, mis võimaldab ühenduse teist osapoolt autentida ning andmeid edastada krüpteeritult ja tervikluskontrolliga, enne kui nad üle võrgu saadetakse. TLS ühendusel on kaks faasi, esimene on käepigistus ja teine on andmete saatmine. Esimene neist on üldjuhul teisest arvutuslikult kallim ja peamise osa arvutusest peab tegema veebiserver, mitte klient. Seda nõrkust kasutataksegi teenusetõkestuste tegemiseks.

5.1.1 SSL käepigistuste ummistus

Lihtne rünne, kus ründaja avab serveriga palju turvalisi ühendusi. Kuna iga ühenduse loomine nõuab kliendilt 10-15 korda vähem arvutusi ja andmetöötlust kui serverilt, siis saab väga kiiresti tekitada serverile teenusetõkestuse. Sellel ajal kui protsessor tegeleb ründaja ühenduste arvutamisega, ei saa see teenindada teisi kasutajaid. Keskmine server suudab teha 150-300 käepigistust sekundis, samas klient võib nõuda üle 1000 käepigistuse sama aja jooksul. Ründe teeb efektiivseks see, et klient ise teeb väga vähe ja server peab kasutama väga palju ressursse [57].

5.1.2 SSL renegotian rünne

SSL-i üks võimalus on see, et iga ühendus võib nõuda uue käepigistuse tegemist. Seega saab ründaja ühe ühenduse abil pidevalt nõuda, et server teeks arvutused uuesti. Arvutamine omakorda kasutab palju protsessori jõudlust ja muudab serveri aeglaseks [23][24][25][57].

Lahenduseks oleksid süsteemid, mis sunnivad klienti tegema teatud lisaarvutusi. Kui klient peab serveriga tegema sama palju arvutusi, siis kaob ründe mõte ära. Rünnet on raske avastada, sest välised teenusetõkestusrünnete mõju vähendajad näevad ainult ühte TCP ühendust.

5.2 HTTP ründed

Need on suunatud veebiserverite pihta. Sellised ründed on tulemuslikud sellepärast, et nad on kõrgema kihi ründed ja 4. kihi teenusetõkestusrünnete kaitsemeetodid neid ei peata. Need on populaarsed, sest neid on lihtne teostada, vajavad ründaja poolt vähe arvutusliku jõudu ja tihti on neid keeruline tuvastada. Nad loovad täieliku TCP ühenduse ja jätavad

mulje, et tegemist on täiesti tavalise ühendusega. Selliste rünnete jaoks ei ole vaja suurt robotivõrku, saab hakkama ka tavalise sülearvutiga [26][27].

5.2.1 Slowloris

Slowloris on aeglane ja varjatud rünne. Erinevalt ummistusrünnetest ei proovita pakettidega ära uputada tervet võrku, vaid rünnatakse ainult veebiserverit, jättes teised teenused kasutuskõlblikuks. Slowloris hoiab ühenduse serveriga avatuna, saates osalisi HTTP päiseid iga teatud aja tagant, nii et server ei saaks ühendust sulgeda. Slowloris peab ootama kuni veebisokkel vabaneb, et seda kasutada. Kui tegemist on populaarse leheküljega, siis võib aega minna, enne kui kõik veebisoklid vabanevad.

Slowlorise teeb varjatud ründeks see, et esiteks saab serverile saata erinevaid kliendi päiseid. Teine põhjus on aga see, et logidesse ei kirjutata enne ühenduse lõppemist midagi. Kui rünne lõppeb või sessioon lõpetatakse, siis ilmub veebiserveri logidesse veateateid „400 *bad request*“.

Tegemist ei ole TCP-l põhineval rünnakul, sest luuakse terviklikke TCP ühendusi, aga selle asemel tehakse poolikud HTTP ühendused. Slowloris lubab väga kiiresti veebiserveril minna tagasi normaalsesse seisundisse, vabastades veebisoklid teistele kasutajatele.

Kuna selle ründe edukus sõltub serveri tarkvarast, siis kasutatakse ründe mõju vähendamiseks sellist lahendust, kus haavatavate serverite kaitsmiseks pannakse püsti vähem haavatavad serverid. Näiteks nginx abil kaitsti Apachet [28][29][30][31].

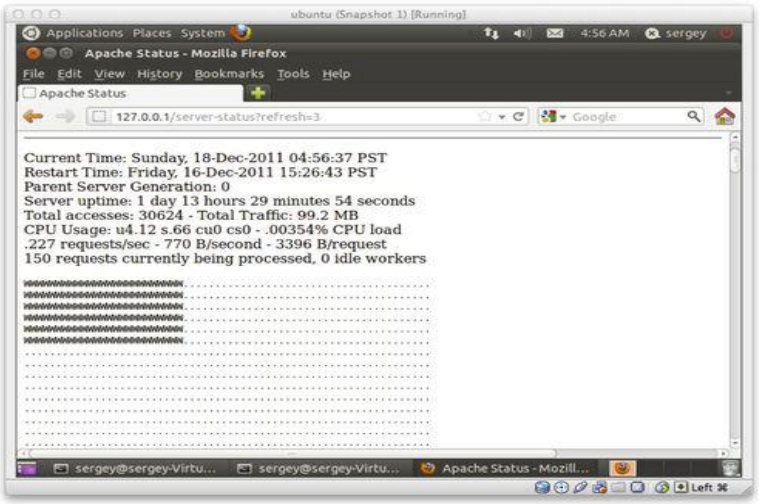
5.2.3 R-U-DEAD-YET (RUDY)

Sarnaselt Slowlorise rünnakule on seda rünnet raske tuvastada ja peatada, sest pakette luuakse vähe ja aeglaselt. Kui kasutaja täidab veebilehel vormi, siis serverile saatmiseks kasutatakse HTTP POSTi. Server töötleb need andmed ära ja valmis saades sulgeb ühenduse ja hakkab teiste külastajate päringuid töötlema. Kui aga kasutatakse RUDY ründeprogrammi, siis saadetakse HTTP päis, kuhu pannakse kirja „*content-length*“ ja seejärel saadetakse HTTP sõnumi andmed serverile ühe baiti kaupa. Server peab seetõttu ühenduse lahti hoidma ja sellega raiskab oma ressursse. Selleks, et server liiga vara ühendust ei sulgeks saadetakse ründepakette kindlate intervallide tagant, millega simuleerib ründaja aeglase Internetiühendusega kasutajaid. Sellise ründe peatamiseks on vaja määrata mõistlik *timeout* päringute lugemiseks veebiserveris [30][32][33][34].

5.2.4 Slow READ, socketstress

Tegemist on ründega, mis kasutab TCP akna suurust. Slow READ ründe puhul hoiab ründaja serveri ühendusi lahti, lugedes serveri poolt saadetud andmeid väga aeglaselt. Kui server hakkab andmeid saatma, siis küsitakse kliendi käest, kui suur on hetkel tema paketti kättesaamise akna suurus. Ründaja vastab, et akna suurus on 0 baiti. Selle peale hoiab server ühenduse avatuna ja küsib teatud aja pärast uuesti kliendi akna suurust. Kui ründaja avab mitu ühendust serveriga ja sunnib ühendusi lahti hoidma, siis tekkib teenusetõkestus teistele kasutajatele, sest serveril ei ole piisavalt vabu ühendusi ja mälu, et teisi veebilehe külastajaid teenindada. Tegemist on ründega TCP tasandil.

Teine võimalus tekitada teenusetõkestusrünne on vastata serverile, et akna suurus on 4 baiti, mis sunnib serverit kogu andmete hulga jagama väikestesse pakettidesse, kasutades sellega ära kogu oma vaba mälu [35][36][37].



Joonis 8: Apache server, mis Slow READ ründe all [37]

5.2.5 Keep-alive rünne

Keep-Alive on HTTP/1.1 protokolli osa ja lubab ühe ühenduse ajal teha palju päringuid. Selle tulemusena saab teha palju päringuid, ilma et süsteemi kaitsemehhanismid aktiveeruksid. Ründajale on see hea, sest iga ühenduse avamine nõuaks ründaja enda ressursse, aga ühe ühenduse lahti hoidmiseks ei ole palju vaja. Kui tavaliselt saadab veebilehitseja GET või POST päringuid, siis server saadab kliendile tagasi andmeid ja ründaja enda võrguriba laius saab otsa. Lahenduseks kasutatakse HEAD-i. See sunnib serverit päringut tegema, aga ei saada tulemust tagasi ründajale. Kuna see rünne kasutab ära veebiserveri CPU ja RAM-i, siis kasutatakse seda rünnet nendel lehekülgedel, mille genereerimine nõuab palju ressursse, näiteks otsingud [38].

5.2.6 HTTP GET rünne

Klassikaline rünne, kus ründaja laseb oma robotivõrgul ohvri veebilehelt alla laadida väga suuri faile, näiteks videoid. Server koormatakse päringutega üle ja muutub aeglaseks, häirides sellega teiste kasutajate veebikülastusi. See ei ole tänapäeval kuigi efektiivne, sest sellise ründe filtreerimine on väga lihtne [39].

5.3 P2P rünne

P2P (*peer-to-peer*) tehnoloogiat kasutatakse failide jagamiseks üle võrgu ilma keskse infrastruktuurita. Kuna neid süsteeme kasutavad paljud inimesed, siis on avastatud ka see, kuidas kasutada seda suurt hulka inimesi, tõkestusrünnete tegemiseks. Peamine ründe meetod kasutab indeksfaili mürgitamist.

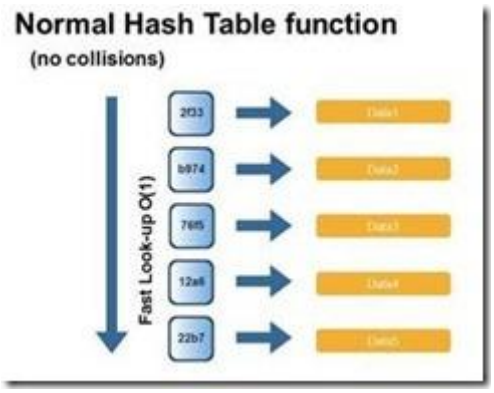
Paljud P2P süsteemid kasutavad indeksfaili, kus on kirjas mingid väärtused ja nende asukohad. Näiteks Skype indeksfailis on kirjas kasutajanimi ja tema aadress. Torrentvõrgud kasutavad samasugust süsteemi, omades infot selle kohta, kes ja kui palju omab allalaetavat faili.

Ründaja mürgitab indeksfaili öeldes, et näiteks filmi või raamatut saab ohvri aadressilt. Kui teised võrgu kasutajad otsivad seda populaarset faili, siis indeks annab neile teada, et see fail on kättesaadav ohvri aadressil. Iga klient loob ohvriga TCP ühenduse ja üritab faili alla laadida, aga kuna ohver ei saa nõutava päringuga, siis ta lihtsalt vastab veateatega ja sulgeb TCP ühenduse [40][51].

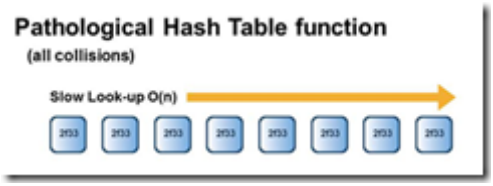
5.4 HashDos

Rünne paisktabelite pihta. Paisktabeleid kasutatakse selleks, et luua kiire ligipääsuga andmestruktuur, mille abil saavad arendajad mugavalt infot kasutada. Modernsed veebirakendused sisaldavad tavaliselt veebivorme, kus moodustatakse võtmeväärtuste ja andmete paarid, mis saadetakse rakendusele. Enamasti pannakse need väärtused sõnastikku (*dictionary*). Sõnastikud kasutavad andmete hoidmiseks paisktabeleid. Ründaja loob palju sama andmevõtmega väärtusi ja laseb need sisestada paisktabelitesse. Kui nüüd küsitakse võtmeväärtusega andmeid, siis kuna neid väärtusi on hästi palju, siis tekivad võtme kollisioonid. Teenusetõkestus tekibki sellest, et selliste väärtuste poole pöördumine võtab

protsessorilt märgatavalt rohkem aega lisaahelate läbimise tõttu [41][42].



Joonis 9: Normaalne räsitabelite kasutamine [64]



Joonis 10: Kui ründaja tekitab palju võtmeväärtuse kokkupõrkeid , siis kulub otsimiseks aega O(n) [64]

6. Kaitsemeetodid

Viimasel ajal on kahju tekitamiseks üha rohkem kasutatud teenustõkestusründeid. Kaitseks on kasutusele võetud erinevaid süsteeme ja erinevad firmad on loonud seadmeid ja teenuseid, mis aitavad ründe mõju vähendada.

Kaitsemeetodid saab jagada neljaks [2].

- Ründe peatamine
- Ründe avastamine
- Ründele reageerimine
- Ründe mõju vähendamine

Parim lahendus teenusetõkestusrünnete vastu on nende peatamine enne kahju tekitamist. Näiteks globaalsed filtrid peatavad ründe enne, kui see jõuab sisevõrku. Lisaks peab jälgima, et kõigil süsteemidel oleksid uusimad turvapaigad. *Honeypot'*ide ja võrguliikluse tasakaalustajate kasutamine tagab selle, et ohvril on rohkem aega ründele reageerida.

Kindlasti peab ründe selle mõju vähendamiseks kiiresti avastama. Sissetungi tuvastamiseks on kaks peamist meetodit. Esimene neist on eelnevate kogemuste põhjal moodustatud signatuurid. Teine on kahtlase tegevuse ja võrguliikluse tuvastamine süsteemis. Võrgusüsteemide jälgimisel saab luua standardseisundi ja paika panna ootused süsteemile. Anomaaliate avastamine võrguliikluses võib olla märgiks, et tegemist on teenusetõkestusründega.

Ründe mustreid teades on võimalik rünnet võrguliiklust jälgides hõlpsasti tuvastada. Signatuuride omamine on väga efektiivne kaitse. Kui süsteeme rünnatakse, siis on võimalik seda kiiresti tuvastada. Probleemiks on teenusetõkestusrünnete pidev muutumine. See omakorda eeldab signatuuride andmebaasi pidevat ajakohastamist, aga andmebaasi uuendamine ja haldamine on suhteliselt keerukas ja ajamahukas.

Ründele reageerimine on järgmine oluline etapp. Ründe avastamise järel uuritakse, kust rünne pärineb ja alustatakse blokeerimistööd. Üheks lahenduseks on IP tagasijälitus, mille puhul üritatakse rünnet jälitada selle alguspunkti, tuvastades sellega ründaja identiteedi. Kuna IP-aadresse on võimalik võltsida, siis on ründe päritolu raske tuvastada. Teiseks

võimaluseks on võrguliikluse analüüs, mis aitab leida ründe karakteristikud ja omadused. Tulemusi saab kasutada võrguliikluse tasakaalustajates. Samuti võimaldab see ründe peatamiseks rakendada uusi filtreerimistehnikaid tulemüürides. Kolmas võimalus on analüüsida sündmuste logisid, mille tagajärjena saab avastada uusi ründeid. Selleks kasutatakse tulemüüre, liikluse pealtkuulajaid, serveri logisid ja *honeypot*'e.

Viimane ja kõige olulisem on ründe mõju vähendamine. Kuna üldjuhul ei saa rünnet otseselt peatada, siis peavad olema kasutuses süsteemid, mis tagaksid teenuse pakkumise jätkamise. Üks võimalus on tõrkekindlus ehk võrgu teenused ja süsteemid on dubleeritud. Süsteem saab jätkata tegevust ka siis, kui üks osa süsteemidest ei ole kasutatav. Teine võimalus on QoS (*Quality of Service*) süsteemide rakendamine, mis klassifitseerivad võrguliikluse ja tagavad selle, et prioriteediga liiklus võib võrgu läbida enne, kui teised andmevood. QoS tagab selle, et isegi ründe all olles suudavad võrk ja teised süsteemid tagada teenuste pakkumist oma kasutajatele.

Vastavalt sellele, kus kohas kaitsemeetodeid rakendatakse, saab neid jaotada järgnevalt

- Ohvri võrgus
- Ründaja ja ohvri vahelises võrgus
- Ründe allika võrgus

Enamus lahendusi ja süsteeme rakendatakse ohvri võrgus, sest see kannatab ründe korral kõige rohkem ja üldiselt on see ohvri enda vastutusel. Ohvrist ülesvoolu olevatele võrkudele saab ka rakenda kaitsemeetodeid ja nende efektiivsus on väga hea, kuid kuna need võrgud üldiselt ei ole rünnakutest mõjutatud, siis ei kasutada väga palju kaitsemeetodeid. Kui ründe allika juures olevatel võrkudel on rakendatud kaitsemeetodeid, siis võib peatada ründe enne, kui see jõuab Interneti tuuma välja. Lisaks saab ründaja kiiremini tuvastada. Ainuke probleem on selle süsteemi juures see, et kui ei suudeta korrektselt rünnet tuvastada, siis võidakse piirata tavaliste kasutajate Interneti kasutust.

6.1 Kaitsemeetmeid vastavalt rünnetele

Enamus vigaste pakettide ründeid on hõlbus peatada, kuna selliste pakettide avastamine on väga lihtne. Tulemüürid ja teenusetõkestusrünnete peatamise süsteemid viskavad ründepaketid minema enne, kui need jõuavad sisevõrku. Näiteks Ping of Death, Christmas tree, ja LAND pakettid filtreeritakse võrguliiklusest välja. Tänapäeval suudavad

tulemüürid panna fragmenteeritud paketid tagasi kokku ja alles siis rakendavad neile filtreerimist. Vajadusel korral saavad nad teha ka parandusi.

Ummistusrünnete peatamine on keerulisem, kuna ohvril pole selliste rünnete peatamiseks tavaliselt piisavalt vabu ressursse. Kõige tavalisem lahendus on määrata andmevoogudele limiidid.

SYN ummistusründe vastu on aja jooksul välja kujunenud erinevad kaitsetehnikad. Esimene neist on SYN proksi kaitse, mis leidub paljudes kaasaegsetes tulemüürides. Aeglustatakse TCP ühendusi ja filtreeritakse välja ründeühendused. Teine lahendus on SYN puhver, mis optimeerib mälutabeleid, et mahutada rohkem ühendusi. Kolmas lahendus on SYN küpsiste kasutamine. Kasutatakse krüpteeritud järjendite numbreid, et filtreerida välja kehtetud sessioonid. Süsteemid panevad SYN paketiga kaasa ühenduse info ja kui tegemist on päris ühenduse algatusega, siis saavad SYN-ACK vastusega selle info tagasi.

RA uputuse korral on kõige lihtsam lahendus lõpetada IPv6 kasutamine või kui see pole võimalik, siis keelata *Router Discovey* võimalus. Kuna mõlemad lahendused ei ole kõikides süsteemides võimalikud, siis parim viis ründe peatamiseks on keelata tulemüüris võlts *Router Advertisments* ja lubada teateid ainult autoriseeritud võrguväravatelt. Turul on nüüd saadaval ka kommutaatorid, mis blokeerivad RA uputused.

Võimendusrünnete mõju ja kasutamist vähendab see, kui sulgeda nõrkused peegeldajates. Näiteks Smurf ja Fraggle rünnete puhul peaksid võrguadministraatorid sulgema marsruuteri võimaluse edastada päringuid leviedastusaadressile. See tagab selle, et võrku ei kasutada rünnete teostamiseks. Ohvri seisukohalt tuleb üles seada süsteemid, mis jälgivad võrguliiklust ja määrata vastavatele andmevoogudele piirid. Juhul kui piirist minnakse üle, siis visatakse paketid minema.

SMTP rünnete piiramiseks tuleb seada piirangud, näiteks genereerida vähe ja väiksed veateated, see vähendab võimenduse mõju. Määrata ära, millised kasutajad võivad üldse kirju saada ja paika panna ülim piir, kui palju võib olla kirja saajaid ühe sessiooni ajal. Tänapäeval ei võta SMTP serverid selliseid kirju vastu ja annavad veateate juba SMTP seansis [21].

DNS võimendusrünnete puhu on lahenduseks piirata, millistele päringutele DNS serverid vastavad. Probleemiks on see, et paljud hoiavad nimeserverid avatuna ja nad vastavad

kõigile päringutele. Lahendus on lubada päringuid teha ainult usaldusväärsetelt võrkudelt [22][63].

SSL *renegotiation* ründe puhul ei aita selle võimaluse ära keelamine, sest siis muudetakse rünne ümber tavaliseks SSL käepigistusummistuseks. Lahenduseks on SSL arvutuste tegemine, liikuda serverilt ära ja panna teised süsteemid seda tegema, näiteks võrguliikluse tasakaalustajad või spetsiaalsed SSL arvutussüsteemid.

Aeglaste HTTP rünnete korral tuleb ära määrata lubatud agressiivsete piirangute arv:

- Määrata limiidid päistele ja sõnumi osadele, vastavalt oma süsteemi eripäradele.
- Kindel aegumine ühendustele. Valides liiga lühikese aja piiratakse õigeid kasutajaid, valides liiga pika aja, ei saada kaitset ründe eest.
- Lisada serverile süsteem, mis toetab pooleliolevate ühenduste salvestamist ja hiljem vastamist.
- Määrata minimaalne sissetulev andmevoo suurus ühenduse kohta

Üks võimalus on ka süstida JavaScript koodi veebilehtedesse. Sellega saab eraldada robotvõrgu robotid õigetest kasutajatest [62].

Tavalise GET ummistuse korral tuleb määrata piirangud vastavalt serveri jõudlusele. Lisaks ka piirangud, kui palju ühendusi võib tulla ühelt IP-aadressilt ning kui palju võib kindlat veebi ressurssi alla laadida.

P2P ründe üheks peatamise võimaluseks on enne kontrollida, kas indeksfailis reklaamitav aadress kuulub P2P võrku. Enamasti ohver ei kuulu P2P võrku ja selle abil saab kiiresti eemaldada väärad aadressid. Teine võimalus on krüpteerida võrguliiklus ja lubada vaid sõlmedel ennast reklaamida P2P võrgule. Ohvri lahenduse kaitseks on visata paketid P2P võrgust minema [40][61].

HashDoS rünnet saab peatada signatuuriga. Kui avastatakse POST, mis sisaldab palju võtmeväärtusi või millel on liiga palju andmeid kaasas, siis visatakse pakett minema, nii et andmebaasid ei pea seda kasutama. Osad programmeerimiskeeled on selle probleemi juba lahendanud kasutades suvalist sõna ja XOR sissetulevate andmete puhul. Teine võimalus on piirata sissetulevate andmete kogust programmeerimiskeele tasemel [42].

6.2 Teenusetõkestusründe allika tuvastamine

Rünnete allika tuvastamine on parim viis, kuidas peatada teenusetõkestusrünne. Kahjuks on allika jälitamine väga keerukas. Kindlasti tuleb teha koostööd Interneti-teenuse pakkujaga, sest mõnikord saavad nad aidata ründe mõju vähendamisega, filtreerides andmevoogusid enne, kui need jõuavad ohvrini.

Teenusetõkestusrünnete jälitamine on keeruline ja aeganõudev töö. Kõige tavalisem lahendus on tuvastada, milline marsruuter saadab ründepakette ülesvoolu. Tavaliselt kuuluvad need mingile Interneti-teenuse pakkujale, kellega tuleb ühendust võtta ja lasta neil seadistada filter, mis eemaldab ründepaketid.

Üks võimalus teenusetõkestusründe peatamiseks on tuvastada isik, kes saab kasu sellest, et ohvri võrk või veebileht ei ole kättesaadav. Nendeks võivad olla kas endine pahatahtlik töötaja, konkurent või kuritegelik rühmitus. Põrandaaluste foorumite ja jututubade jälgimine, kus toimub robotvõrkude rentimine ja rünnete arutamine, võib anda infot, kes soovib halba. Selline uurimistöö nõuab kogemustega inimesi ja palju koostööd, kuid ründaja tuvastamine annab kohese efekti.

Kuna IP-aadressid on võltsitavad, siis on väga raske tuvastada, kust ründed pärinevad. Siiski on välja arendatud tehnikad, mille abil saab vähendada rünnete mõju, kasutades selleks filtreerimist ülesvoolu marsruuterites.

Üks lahendus teenusetõkestusründe jälitamiseks on manuaalne ACL (*Acess Control List*) tagasijälitus. Interneti-teenuse pakkuja määrab marsruuteris algul üldiste parameetritega ACL ja mida rohkem saadakse teada ründe kohta, seda spetsiifilisemaks muudetakse parameetrid, kuni lõpuks saadakse teada, millised on ründavate andmevoogude karakteristikud. Selle info abil saab määrata ülesvoolu oleva marsruuteri allikaliidese ja MAC aadressi. Siis peab seadmes kordama sama protseduuri, kuni jõutakse ründe allikani. See on aga ajakulukas ja kui tegemist on hajusa teenusetõkestusründega, siis hargneb jälitustöö ülesvoolu olevates marsruuterites mitmeks.

Teine võimalus on hajusjälitus. Interneti-teenuse pakkuja ääremarsruuterid tuvastavad ründevood ja viskavad need paketid minema, genereerides sellega ICMP *unreachable* paketti ja saadavad need tagasi võltsitud IP-ga aadressidele. Kui aga *sinkhole* marsruuter reklaamib era- või kasutamata aadressiruumi, siis need ICMP paketid suunatakse lõpuks

sinna seadmesse. Siis peab lihtsalt jälgima, et millised marsruuterid genereerivad neid pakette ja saadakse teada IP-d [58][59][60].

6.3 Tooted

Paljud firmad on turule tulnud oma süsteemiga, mis kaitseb teenusetõkestusrünnete eest. Kindlasti tuleb jälgida kuhu need süsteemid võrgus paigutatakse. Üldiselt soovitatakse panna selliseid süsteeme võrgu äärealadele, et ründed võimalikult vara peatada.

6.3.1 Cisco Systems

Cisco pakub kahte erinevat süsteemi teenusetõkestusrünnete jaoks. Esimene on Cisco Traffic Anomaly Detector XT [43], mille ülesandeks on passiivselt võrku jälgida ja anomaaliate tekkimisel teavitada sellest koheselt teist seadet Cisco Guard XT [44], mis alustab võrguliikluse analüüsimist ja filtreerimist. Kui tuvastatakse ründaja saadetud paketid, siis kaotatakse need koheselt ja teiste kasutajate paketid suunatakse edasi õigesse sihtpunkti.

Cisco Guard XT kasutab rünnete vastu viie-astmelist MVP (*Multiverification*) struktuuri [45].

- Pakettide filtreerimine on esimene moodul struktuuris. Staatilised filtrid, mis blokeerivad mittevajaliku võrguliikluse jõudmise kasutajani. Filtrid on Cisco poolt juba eelnevalt seadistatud. Juhul kui tuvastatakse pahatahtliku andmevoo eripära, siis saavad dünaamilised filtrid oma reeglid teistelt moodulitelt.
- Aktiivse tõendamise mooduli ülesandeks on avastada võltsitud aadressiga pakette. Lisaks leidub erinevaid mehhanisme, mis tagavad selle, et andmevoost eemaldatakse ainult ründepaketid.
- Anomaalia tuvastamise moodul jälgib eelmised moodulid läbinud võrguliiklust ja võrdleb seda varem salvestatud normaalse võrguliikluse tunnustega. Kuna ründe andmevood erinevad teatud aspektides tavaliste kasutajate omadest, siis saab võrdluse teel eemaldada kahtlased paketid.
- Protokolli analüüs on järgmine etapp ning antud moodul analüüsib rünnete tuvastamiseks seda liiklust veelgi täpsemalt. Kasutatakse selleks, et tuvastada ründed, mis kasutavad kindlaid protokolle, näiteks HTTP.

- Andmevoogude piiramine on viimane moodul ja selle abil analüüsitakse liialt kasutaja ressursse kasutavaid andmevooge, mis aeglustab kasutaja ligipääsu sisevõrgule.

6.3.2 F5 Networks

F5 Networks töötab välja ja müüb võrguseadmeid. Nende lipulaev BIG-IP oli alguses võrguliikluse tasakaalustaja. Hiljem on funktsionaalsust juurde lisatud. Seadmed saavad teenusetõkestusründe peatamisega hakkama. Välja kujunenud on oma Application Delivery Controller (ADC) [46].

BIG-IP Local Traffic Manager – peatab vigaste pakettide ründed.

- *Packet Velocity Accelerator* – eraldi disainitud riistvara protsessor, mis aitab BIG-IP LTM vähendada klassikaliste ummistusrünnete mõju.
- Täisproksi arhitektuur – tagab turvalisuse sellega, et klientidelt tulev võrguliiklus analüüsitakse enne, kui see saadetakse rakenduskihile.
- Protokolli kontroll – jälgitakse, et võrku ei siseneks valesti määratud lippude või puudulike andmetega pakette. Peatab *FRAG* ja *Christmas tree* ründed.

BIG-IP Global Traffic Manager – kaitseb hajusate teenusetõkestusrünnete eest ja DNS Express kontrollib üle kõik DNS päringud enne võrku lubamist.

BIG-IP Advanced Firwall Manager – AFM abil saavad võrguadministraatorid kiiresti ja efektiivselt luua turvareeglistike. Lisaks jälgitakse ja antakse teada, kui teenusetõkkerünne toimub.

BIG-IP Application Security Manager – peatab seitsmenda kihi ründed. Oskab eristada inimeste ja robotite ründeid. Süstib veebilehtedesse JavaScript *redirect* koodi, eemaldades sellega robotivõrgustiku orjad. Kui avastab, et tegemist on ründega, siis määrab ka kiirusepiirangud andmevoogudele.

iRules on skriptimiskeel, mis laseb võrguadministraatoritel luua kiiresti ja tõhusaid turvareegleid.

6.3.3 CloudFlare

CloudFlare on sisuedastusvõrk (CDN) ja hajus domeeninimede teenus (DNS). Firmal on kogu maailmas 23 andmebaaside ja serverite klastrit, mis tagavad selle, et klientide lehed oleksid kiiresti kättesaadavad. Kasutatakse *Anycast* tehnoloogiat, selleks et veebikülastaja

DNS-i päring jõuaks temale lähimasse CloudFlare serverite klastrisse. Seal asuv puhverdatud veebileht saadetakse veebikülastajale. Kõik DNS päringud tasakaalustatakse kõigi 23 klastri vahel ning see on üldjuhul väiksemate hajusate rünnete puhul juba piisav, et kliendi veebileht jääks kättesaadavaks. Isegi kui mõni klaster muudetakse ründe tulemusena töövõimetuks, saavad ülejäänud andmebaasid ja klastrid külastajaid teenindada. Lisaks puhastatakse igas klastris täiendavalt andmevoogusid. Näiteks visatakse minema kõik DNS vastused, sest CloudFlare ei tee ühtegi DNS päringut.

Rakenduskihi rünnete jaoks on loodud eraldi teenus: „*I'm under attack*“, mis lisab täiendava kihi turvalisust HTTP rünnete vastu. Veebikülastajale näidatakse vahelehekülge, mis käitub kui automaatne CAPTCHA ja selle taustal tehakse täiendavad testid ja analüüsid, kontrollimaks kas tegemist on ründaja või tavalise külastajaga [47][48].

6.3.4 Check Point

Check Point on rahvusvaheline firma, mis pakub erinevaid tarkvaralisi ja riistvaralisi infoturbe lahendusi oma klientidele [49].

SmartEvent Blade – kasutatakse ründaja profiili ründe mustrite kiireks tuvastamiseks.

SmartLog funktsioon – analüüsides logisid erinevatelt süsteemidelt, saab SmatLog tuvastada ründe.

Firewall Software Blade – sisseehitatud süsteemid teenusetõkestusrünnete peatamiseks.

- Agressiivne vananemine – ühendused, mis on avatud kauem kui algselt määratud, suletakse ja kustutatakse võrguvärava tabelitest. Kaitseb aeglaste HTTP rünnete vastu.
- Võrgu kvoot – määrab ära, kui palju ühendusi võib ühelt IP-aadressilt olla. Kui avatakse lubatust rohkem ühendusi, siis kas keelatakse uute avamine või jälgitakse täpsemalt andmevoogu.
- Blokeeritakse ICMP/UDP – sellised ründepaketid visatakse võrgu perimeetril minema.
- Olekuga ühenduste inspekteerimine – vähendatakse aega, kui kaua võib mõni ühendus olla avatud. Toimib rünnete vastu, mis on aeglased ja nõuavad palju ressursse.

IPS Software Blade – täiendavad kaitsemeetodid teenusetõkestusrünnete vastu.

- Riikide kaupa blokeerimine – blokeeritakse võrguliiklus riikidest, kust pärineb palju rünnakuliiklust.
- Ussipüüdjate signatuurid – blokeeritakse URL-id, mida kasutatakse ründeks.
- TCP akna suuruse jõustamine – tagab kaitse ründe vastu, mis kasutavad TCP akna suurust.
- SYN ummistusrünnete kaitse – käivitatakse siis, kui võrku siseneb üle 200 SYN paketi 5 sekundi jooksul.
- HTTP ummistusrünnete kaitse – käivitatakse siis, kui tehakse rohkem kui 10000 päringut 10 sekundi jooksul.

Check Point DDoS protector [50] – riistvaraliselt kiirendatud ja spetsiaalse tarkvaraga süsteem, mis paigaldatakse väljapoole võrgu perimeetrit. Tuvastab ja peatab teenusetõkestusründed, enne kui need jõuavad sisevõrku.

- Võrgu ummistusrünnete kaitse – jälgitakse, millised on tavapärased võrguliikluse mustrid ja ebatavaline andmevoo avastamisel alustatakse filtreerimisega.
- Serveri ummistuskaitse – genereerib igale ühendusele unikaalse signatuuri jälgimaks ühendusi ja ründe korral need peatatakse.
- Rakenduskihi kaitse – blokeerib automaatsed tööriistade ründed ja võltskasutajad, kasutades selleks väljakutse/vastuse tehnikat. Samal ajal suunatakse tavakasutajad edasi oma sihtpunkti.

6.3.5 Radware

Radware pakub rakenduse kättetoimetamise, võrgu turvalisuse ja võrguliikluse tasakaalustaja lahendusi. Radware DefensePro [52] on seade, milles on ühendatud IPS (*Intrusion Prevention System*), NBA (*Network Behavioral Analysis*), *DoS Protection* ja *Reputation Engine*. Koostöös suudavad need neli süsteemi peatada erinevaid sissetungi ründed. Teenusetõkestusrünnete avastamiseks kasutatakse ründe signatuuride tuvastamist [52][53].

DoS Mitigation Engine (DME) on riistvaraline lahendus hajusate teenusetõkestusrünnete vastu. Omades kuni 40Gbps läbilaskevõimet suudab see tuvastada ja peatada teenusetõkestusründeid.

Network Behavioral Analysis moodul kasutab signatuuridel põhinevad kaitsetehnikat. Teenusetõkestusrünnete puhul süstitakse reaalaja signatuur otse DME riistvarasse vabastades sellega seadme protsessori ja jättes kogu töö DME teha.

Denial-of-service Protection moodul kasutab erinevaid tehnoloogiaid teenusetõkestusrünnete peatamiseks. Signatuuride avastamine, käitumispõhised reaalaja signatuurid ja SYN *cookies* mehhanism, mis esitavad väljakutse enne, kui uued ühendused saavad luua seansi serveriga.

5.3.6 Arbor Networks

Arbor Network on tarkvarafirma, mis müüb võrguturbe ja võrguseire tarkvara. Koostöös firmadega Cisco, IBM ja Juniper Networks on välja töötatud erinevaid lahendusi robotvõrkude, võrguside ja teenusetõkestusrünnete vastu.

Peakflow SP Threat Management System (TMS) vähendab teenusetõkestusrünnete mõju, eemaldades ründe andmevoo tavaliste kasutajate omadest. Süsteem toetab seda, et kõik võrku läbivad andmevood saab ründe korral suunata TMS-i, mis eemaldab ründepaketid ja suunab puhtad paketid tagasi võrku.

Teadaolevate ohuallikate blokeerimiseks kasutatakse musti ja valgeid nimekirju. HTTP-põhiste rünnete peatamiseks on veel lisaks IP-põhised piirangud. Eemaldatakse vigased paketid ja piiratakse andmevoogusid, mis tahavad kasutada liiga palju ressursse.

Peakflow SP TMS õpib automaatselt, millised on normaalsed võrguliikluse mustrid ja kohandab oma reegleid vastavalt sellele. Kui vaja siis võib need ka ümber seadistada vastavalt vajadusele ning see lubab kasutada TMS süsteemi koheselt karbist välja võttes [55][56].

Kokkuvõte

Töö alguses on kirjeldatud üldiselt teenusetõkestusründeid, kuidas neid klassifitseerida ja jaotada. Vastavalt sellele, millist nõrkust ära kasutatakse, on ründed jaotatud neljaks ja igas peatükis on kirjeldatud erinevaid selle jaotuse ründeid. Refereerides on kasutatud Internetis vabalt kättesaadavaid materjale.

Töö teises pooles on kirjeldatud kaitsemeetmeid ning kuidas neid jaotatakse. Lisaks on välja toodud ka lahendused rünnetele, mida on eelnevalt töös kirjeldatud. Lõpuks on kaitsemeetodite illustreerimiseks toodud välja ka erinevate firmade tooteid ja lahendusi, mis on loodud rünnete tuvastamiseks ja mõju vähendamiseks ning kirjeldatud nende ülesehitust.

Tulevikus võiks seda tööd edasi arendada näiteks uurides rakenduskihis teostatavaid ründeid. Kuna selliseid ründeid arendatakse tõenäoliselt tulevikus rohkem välja, siis on mõistlik olla nendega kursis. Teiseks uurimisobjektiks võiks olla robotvõrgud – kuidas neid kontrollitakse, kuidas leitakse uusi lülisid võrku ja kuidas neid kasutatakse rünnete teostamiseks.

Denial of Service Attacks and Defense Solutions

Bachelor's Thesis (6 ECTS)

Erki Vaino

Summary

Over the last years denial of service attacks have been gaining a lot of popularity amongst hackers and activists. New more sophisticated methods of attack have been developed and used against users across Internet. Idea behind the attack is to consume enough victims resources that he is no longer able to serve other legitimate users. In the beginning there is a short overview of DoS attacks and how can they be classified.

DoS attacks by exploited vulnerability:

1. Malformed packet attacks
2. Flooding attacks
3. Amplification attacks
4. Protocol exploit attack

This method of classification is used to segment diffrent attacks in to groups.

Early days of DoS attacks consisted mostly of malformed packet attacks and attacks that flooded networks with a lot of data. On today's network these attacks have little effect because packets with faulty data will be dropped by routers and switches before any damage can be done.

10-15 years ago flooding attacks were serious problems to victims. But because of today's powerful computers simple flooding attacks have lost their effect on networks. Distributed denial of serve attacks are much more powerful and will cause serious damage to networks and systems. DNS server and unprotected networks are used to amplified the attacks and can cause serious outage to networks. Hackers can also easily rent botnets to do attacks on the victim.

More sophisticated attacks are used on application layer. These attacks don't require a large botnet to do damage. A simple laptop will be able to take a webserver offline with Slowlorise or RUDY attack. These attacks are hard to detect and mitigate.

Popular P2P technologies are also used in denial of service attacks because of their large user base, who can be used as attackers without them even knowing.

Due to SSL requiring a lot of computation power from the server some attacks have been developed to use that in attackers advantage.

Second half of the work is to give a overview of defense methods. Organizations need to understand that DoS attacks can cause serious financial and reputation loss. Over the years defense methods and solutions have been created to combat the rising threat of DDoS attacks.

DDoS defense mechanism by activity [2]

1. Intrusion prevention
2. Intrusion detection
3. Intrusion response
4. Intrusion tolerance and mitigation

Preventing attacks even getting to the network is the best kind of defense, but not always possible, since some application layer DoS attacks can be stealthy and go unnoticed until it is too late, detecting attacks is really important. Also reacting to attacks needs to be considered, having a plan, and response should be considered by IT departments. Since DoS attacks are really hard to stop completely, mitigating and tolerating the effects is the best solution.

Many companies have developed solutions against attacks. Cisco, F5 and Check Point have developed special hardware and software products against denial of service attacks.

CloudFlare is Content Delivery Network and due to its distributed nature can easily protect against layer 3 and 4 flooding attacks. In addition CloudFlare have developed solutions against higher level attacks and are able to keep websites up, even during serious DDoS attacks.

Viited

1. CloudFlare advanced DDos protection https://www.cloudflare.com/ddos (12.05.2013)
2. Christos Douligeris ja Dimitrios N. Serpanos, Network Security Current Status and Future Directions, 2007
3. LAND Attacks http://www.imperva.com/resources/glossary/land_attacks.html (12.05.2013)
4. Understanding Land Attacks http://www.juniper.net/techpubs/software/junos-security/junos-security96/junos-security-swconfig-security/land-attacks-understanding.html#land-attacks-understanding (12.05.2013)
5. LAND Attack http://security.radware.com/LAND-attack.aspx (12.05.2013)
6. The LAND attack (IP DOS) http://insecure.org/sploits/land.ip.DOS.html (12.05.2013)
7. Christmas Tree Attacks http://www.aries.net/home/demos/Security/chapter2/2_1_4.html (12.05.2013)
8. Understanding Teardrop Attacks http://www.juniper.net/techpubs/software/junos-es/junos-es93/junos-es-swconfig-security/understanding-teardrop-attacks.html (12.05.2013)
9. Teardrop Attack http://security.radware.com/knowledge-center/DDoSPedia/teardrop-attack/ (12.05.2013)
10. Stelios Antoniou, The PING of Death and Other DoS Network attacks http://www.trainsignal.com/blog/ping-of-death-and-dos-attacks (12.05.2013)
11. Understanding Ping of Death Attacks http://www.juniper.net/techpubs/software/junos-security/junos-security96/junos-security-swconfig-security/id-64486.html#id-64486 (12.05.2013)
12. Rose Frag Attack Explained http://digital.net/~gandalf/Rose_Frag_Attack_Explained.htm (12.05.2013)
13. Understanding ICMP Flood Attacks http://www.juniper.net/techpubs/software/junos-security/junos-security96/junos-security-swconfig-security/id-25581.html#id-25581 (12.05.2013)
14. CERT Advisory ca-1996-01 UDP Port Denial-of-Service Attack http://www.cert.org/advisories/CA-1996-01.html (12.05.2013)
15. Understanding UDP Flood Attacks http://www.juniper.net/techpubs/software/junos-security/junos-security96/junos-security-swconfig-security/id-60351.html#id-60351 (12.05.2013)
16. TCP SYN Flooding Attacks and Common Mitigations http:/tools.ietf.org/html/rfc4987 (12.05.2013)

17. Understanding SYN Flood Attacks http://www.juniper.net/techpubs/software/junos-security/junos-security96/junos-security-swconfig-security/id-34128.html#id-34128 (12.05.2013)

18. Layer 7 DoS Attacks and Defenses http://www.youtube.com/watch?v=7zQ8lcgxeZk&list=PL23099A7D790EA725 (12.05.2013)

19. ICMPv6 Router Announcement flooding http://www.mh-sec.de/downloads/mh-RA_flooding_CVE-2010-multiple.txt (12.05.2013)

20. How to Prevent Denial of Service Attack http://www.ids-sax2.com/articles/PreventDosAttacks.htm (12.05.2013)

21. Stefan Frei,Ivo Silvestri, Gunter Ollamann, Mail Non Delivery Message DDoS Attacks http://www.techzoom.net/publications/mail-non-delivery-attack/ (12.05.2013)

22. Matthew Prince, Deep Inside a DNS Amplification DDoS Attack http://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack (12.05.2013)

23. THC-SSL-DOS Attack Tool, http://www.youtube.com/watch?v=Ex2xz0ZOKKs (12.05.2013)

24. SSL/TLS and Computational DoS http:/www.educatedguesswork.org/2011/10/ssltls_and_computational_dos.html (12.05.2013)

25. TLS Renegotiation and Denial of Service Attacks https://community.qualys.com/blogs/securitylabs/2011/10/31/tls-renegotiation-and-denial-of-service-attacks (12.05.2013)

26. Sergey Shekyan, New Open-Source Tool for Slow HTTP DoS Attack Vulnerabilities https://community.qualys.com/blogs/securitylabs/2011/08/25/new-open-source-tool-for-slow-http-attack-vulnerabilities (12.05.2013)

27. Sean Michael Kerner, Denial of Service Attacks Get more Sophisticated http://www.esecurityplanet.com/trends/article.php/3921156/Denial-of-Service-Attacks-Get-more-Sophisticated.htm (12.05.2013)

28. Slowloris HTTP DoS http://ha.ckers.org/slowloris/ (12.05.2013)

29. Slowloris HTTP DoS http://ha.ckers.org/blog/20090617/slowloris-http-dos/ (12.05.2013)

30. ModSecurity Advanced Topic of the Week: Mitigating Slow HTTP DoS Attacks http://blog.spiderlabs.com/2011/07/advanced-topic-of-the-week-mitigating-slow-http-dos-attacks.html (12.05.2013)

31. Sergey Shekyan, Identifying Slow HTTP Attack Vulnerabilities on Web Applications https://community.qualys.com/blogs/securitylabs/2011/07/07/identifying-slow-http-attack-vulnerabilities-on-web-applications (12.05.2013)

32. r-u-dead-yet https://code.google.com/p/r-u-dead-yet/ (12.05.2013)

33. R-U-Dead-Yet, RUDY DDoS Attack Tool http://www.youtube.com/watch?v=k1o9Ya8qxlU (12.05.2013)

34. Kelly Jackson Higgins, Researchers To Demonstrate New Attack That Exploits HTTP http://www.darkreading.com/attacks-breaches/researchers-to-demonstrate-new-attack-th/228000532 (12.05.2013)

35. Windows TCP/IP Denial of Service Attacks (Sockstress) http://www.checkpoint.com/defense/advisories/public/announcement/090809-tcpip-dos-sockstress.html (12.05.2013)

36. Kelly Jackson Higgins, New Denial-Of-Service Attack Cripples Web Servers By Reading Slowly http://www.darkreading.com/attacks-breaches/new-denial-of-service-attack-cripples-we/232301367 (12.05.2013)

37. Are you ready for slow reading? http://shekyan.typepad.com/blog/2012/01/are-you-ready-for-slow-reading.html (12.05.2013)

38. LetDown and HTTP DoS attacks http://securityadventures.wordpress.com/2011/09/21/letdown-and-http-dos-attacks/ (12.05.2013)

39. Lori MacVittie, Layer 4 vs Layer 7 DoS Attack https://devcentral.f5.com/blogs/us/layer-4-vs-layer-7-dos-attack (12.05.2013)

40. Naoum Naoumov ja Keith Ross, Exploiting P2P Systems for DDoS Attacks, http://cis.poly.edu/~ross/papers/p2pddos.pdf (12.05.2013)

41. Denial of Service through hash table multi-collisions https://www.nruns.com/_downloads/advisory28122011.pdf (12.05.2013)

42. David Holmes, HashDos – The Post of Doom Explained https://devcentral.f5.com/blogs/us/hashdos-ndash-the-post-of-doom-explained (12.05.2013)

43. Cisco Traffic Anomaly Detector XT 5600 http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5879/ps6264/ps5887/product_data_sheet0900aecd800fa552.html (12.05.2013)

44. Cisco Guard XT 5650 http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5879/ps6264/ps5888/product_data_sheet0900aecd800fa55e.html (12.05.2013)

45. Defeating DDOS Attacks http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5879/ps6264/ps5888/prod_white_paper0900aecd8011e927.html (12.05.2013)

46. Mitigating DDoS Attacks with F5 Technology http://www.f5.com/pdf/white-papers/mitigating-ddos-attacks-tech-brief.pdf (12.05.2013)

47. CloudFlare security http://www.cloudflare.com/features-security (12.05.2013)

48. CloudFlare: How does CloudFlare work? http://www.quora.com/CloudFlare/How-does-CloudFlare-work# (12.05.2013)

49. Dos Attacks: Response Planning and Mitigation http://www.motiv.nl/documenten/whitepapers/check-point-ddos-whitepaper (12.05.2013)

50. Check Point DDoS Protector Appliances http://www.checkpoint.com/products/ddos-protector/ (12.05.2013)

51. Rich Miller, P2P Networks Hijacked for DDoS Attacks http://news.netcraft.com/archives/2007/05/23/p2p_networks_hijacked_for_ddos_attacks.html (12.05.2013)

52. DefensePro: All-in-One Attack Protection with IPS, NBA, DoS Protection and Reputation Services http://www.radware.com/Products/ApplicationNetworkSecurity/DefensePro.aspx (12.05.2013)

53. DefensePro DDoS Protection http://www.radware.com/Products/ApplicationNetworkSecurity/DDoS_Attack_Protection.aspx (12.05.2013)

54. Ipv6-ra-flood http://nmap.org/nsedoc/scripts/ipv6-ra-flood.html (12.05.2013)

55. Peakflow SP Threat Management System http://www.arbornetworks.com/component/docman/doc_download/8-peakflow-sp-tms-data-sheet-english?Itemid=442 (12.05.2013)

56. Peakflow SP Solution http://www.arbornetworks.com/component/docman/doc_download/6-peakflow-sp-data-sheet-english?Itemid=442 (12.05.2013)

57. DDoS and Security Reports: The Arbor Networks Security Blog http://ddos.arbornetworks.com/2012/04/ddos-attacks-on-ssl-something-old-something-new/ (12.05.2013)

58. Convery Sean, Network Security Architectures, Cisoc Press, 2004

59. Service Provider Security, http://www.cisco.com/web/about/security/intelligence/sp_infrastruct_scty.html#14 (12.05.2013)

60. Matthew Tanase, Closing the Floodgates: DDoS Mitigation Techniques http://www.symantec.com/connect/articles/closing-floodgates-ddos-mitigation-techniques (12.05.2013)

61. Sam Bowne, Win 7 DoS by RA Packets, http://samsclass.info/ipv6/proj/flood-router6a.htm (12.05.2013)

62. Sergey Shekyan, How to Protect Against Slow HTTP Attacks https://community.qualys.com/blogs/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks (12.05.2013)

63. Dave Pisicitello, Do More to Prevent DDNS DDoS Attacks http://blog.icann.org/2013/04/do-more-to-prevent-dns-ddos-attacks/ (12.05.2013)

64. David Holmes, VU#903934 Post of doom https://devcentral.f5.com/blogs/us/vu-903934-ndash-post-of-doom (12.05.2013)

Lisad

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina_____Erki Vaino_____

(*autori nimi*)

(sünnikuupäev: _____29.06.1989_____)

annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

_____Teenusetõkestusründed ja kaitse lahendused_____,

(*lõputöö pealkiri*)

mille juhendaja on _____Meelis Roos_____,

(*juhendaja nimi*)

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **12.05.2013**