VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE ÚTOKU DNS AMPLIFICATION Z PASIVNÍ ANALÝZY DNS PROVOZU

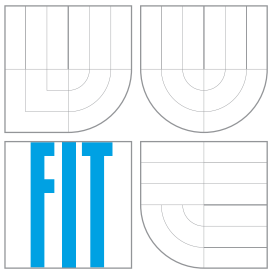
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DANIEL MÍŠANÝ

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE ÚTOKU DNS AMPLIFICATION Z PASIVNÍ ANALÝZY DNS PROVOZU

DNS AMPLIFICATION ATTACK DETECTION USING PASSIVE DNS ANALYSIS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DANIEL MÍŠANÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL KOVÁČIK

BRNO 2014

Abstrakt

Táto práce je zaměřená na analýzu a detekci útoku DNS amplification, který patří mezi útoky typu DoS. Úvod práce je zaměřený na základní teorii zahrnující počítačové sítě, službu DNS a útoky typu DoS. Větší část práce se zabývá analýzou útoku DNS amplification, návrhem a implementací nástroje pro detekci v jazyce C++. Závěr je věnovaný analýze výsledků detekčního nástroje.

Abstract

This thesis is focused on the analysis and detection of DNS Amplification attack which is type of the DoS attack. Introduction of this thesis is focused on fundamental theories involving computer networks, DNS and DoS attacks. The main part of the work deals with the analysis of DNS Amplification attack, design and implementation of detection tool in C++ programming language. The conclusion is devoted to analyzing the results of the detection tool.

Klíčová slova

služba DNS, útok DoS, detekce, útok DNS amplification

Keywords

DNS, DoS attack, detection, DNS Amplification attack

Citace

Daniel Míšaný: Detekce útoku DNS Amplification z pasivní analýzy DNS provozu, bakalářská práce, Brno, FIT VUT v Brně, 2014

Detekce útoku DNS Amplification z pasivní analýzy DNS provozu

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Michala Kováčika. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Daniel Míšaný
18. května 2014

Poděkování

Chcel by som poďakovať vedúcemu mojej práce Ing. Michalovi Kováčikovi za ochotu a spoluprácu pri tvorbe tejto práce.

Obsah

Kapitola 1

Úvod

Už od počiatku počítačových sietí, či internetu sa bolo potrebné zaoberať hrozbou počítačových útokov. Za touto hrozbou môžu stáť amatéri, ktorí to berú ako zábavu, experti, ktorí útočia zo zvedavosti, či organizované skupiny útočiace za úplatu.

Existuje široké spektrum útokov, od útokov zameraných pre zisk určitých informácií, až po útoky zamerané na vyradenie určitej služby, taktiež nazývané DoS útoky, z anglického *Denial of System*. Táto práca je zameraná na jeden špecifický DoS útok – útok DNS amplification, ktorý využíva službu DNS pre vylepšenie útoku. Služba DNS je veľmi kritickou a veľmi vyťažovanou službou na internete. Hlavnou úlohou tejto služby je preklad doménových mien na IP adresy a opačne, čo prináša možnosť identifikácie strojov v sieti doménovým menom, namiesto IP adresy. Avšak, služba DNS nie je potrebná pre samostatné fungovanie sietí, či internetu, no v tom prípade je potrebné používať priamo IP adresy, ktoré sú pre človeka ťažšie zapamatovateľné ako doménové mená.

V minulosti sa na útoky typu DoS nekládla až taká veľká pozornosť, akú by si zaslúžili. Práve skupina Anonymous v roku 2011 prilákala pozornosť na tento typ útokov, keď ich využívali pre dosiahnutie svojich cieľov. O aktuálnosti problému s týmito útokmi a o nepripravenosti inštitúcií na tieto typy útokov svedčia nedávne udalosti, keď boli týmto spôsobom útoku vyradené servery (väčšinou e-mailový a webový) vládnych inštitúcií. Od tej chvíle sa tieto spôsoby útokov stali známe širokej verejnosti, čo malo za následok rozšírenie a vznik nových kriminálnych organizácií využívajúcich útoky typu DoS.

V následujúcej kapitole 2 sa zameriam na teoretickú časť práce, budú nastolené základné informácie týkajúce sa počítačových sietí, služby DNS, spôsobom monitorovania spravovaných sietí a samozrejme aj útokom typu DoS. Kapitola 3 sa zaoberá analýzou útoku DNS amplification a možnosťami detekcie. V kapitolách 4 a 5 sa zaoberám návrhom a implementáciou detektora útoku DNS amplification. Kapitola 6 je určená pre analyzovanie výsledkov imlementovaného detekčného nástroja.

Kapitola 2

Rozbor problematiky

Cieľom tejto kapitoly je nastolenie základnej teórie pre pochopenie následujúceho textu, ktorý je zameraný na útok DNS amplification. V úvode sa zameriam na počítačové siete a následne na službu DNS. Záver tejto kapitoly bude venovaný útokom typu DoS a možnostiam monitorovania siete – NetFlow.

2.1 Počítačové siete

Počítačová sieť je prepojenie dvoch a viacerých zariadení za účelom zdielania informácií alebo zdrojov. Spojenie je uskutočnené buď káblom alebo bezdrôtovo. Zdielanými informáciami, či zdrojmi môžu byť súbory, programy, tlačiarne, modemy alebo iné hardwarové zariadenia.

História počítačových sietí siaha do 60. rokov 20. storočia, keď sa Ministerstvo obrany USA snažilo vymyslieť nový alebo jednoduchší spôsob komunikácie. Do tohoto programu boli zapojené aj významné americké univerzity a to konkrétne University of California a MIT. Toto snaženie a podpora ministerstva vyústilo v roku 1968 k vytvoreniu siete *ARPAnet*.

Sedemdesiate a osemdesiate roky minulého storočia znamenali výrazný posun, keď sa ARPAnet rozvíjal a bolo vytvorené mnoho nových prístupových bodov v USA, ale aj v Európe.

V 90. rokoch minulého storočia bol zaznamenaný ešte väčší rozmach ako v rokoch osemdesiatych. ARPAnet bol premenovaný na *Internet*. V týchto rokoch sa Internet stal komerčným nástrojom. K Internetu bolo postupne pripojených, čím viac študentov, škôl, domácností, tým sa rozvíjal aj obchodný svet na Internete. V druhej polovici 90. rokov sa postupne zvyšovala rýchlosť pripojenia a klesali náklady na prevádzku. To dalo impulz k vzniku napríklad online prehrávania audia, či videa. Na konci roku 1999 bolo na Internete viac ako miliarda stránok.

V súčastnosti sú počítače a siete súčasťou takmer každej časti ľudského života. Máme počítače v domácnosti, v práci, a dokonca aj prenosné zariadenia. Dá sa povedať, že existencia väčšiny týchto zariadení je podmienená práve pripojením na Internet.

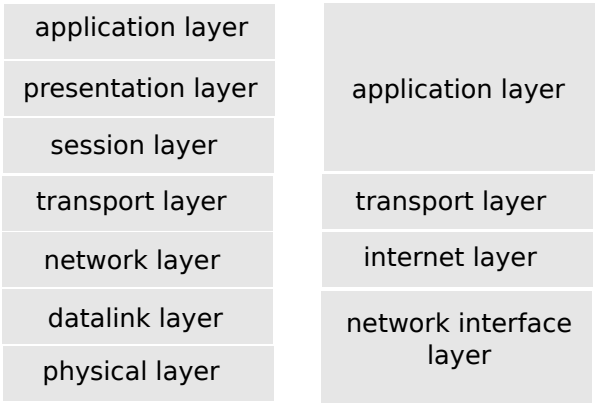
2.1.1 Modely ISO/OSI a TCP/IP

Z historického hľadiska vznikalo veľa modelov architektúry počítačovích sietí. V súčastnosti sa používajú prevažne dva modely – ISO/OSI, ktorý slúži ako referenčný model. Druhý z týchto modelov je model TCP/IP, ktorý je štandardom Internetu a v dnešnej dobe je

implementovaný ako prenosová vrstva u väčšiny počítačových sietí. V následujúcich odstavcoch popíšem tieto modely podrobnejšie.

Model OSI, zobrazený na obrázku 2.1, je tvorený siedmimy vrstvami. Každá vrstva reprezentuje určitý krok v komunikačnom procese. Taktiež v každej vrstve pracuje iný sieťový protokol. Keď si vrstva splní svoju úlohu, ponechá data ďalšej vrstve.

Model TCP/IP v súčastnosti tvorí základ komunikácie na Internete. Oproti predchádzajúcemu modelu má len štyri vrstvy, štruktúra tohto modelu je na obrázku 2.1. Smerovanie datagramov prebieha práve na *internetovej vrstve* (anglicky IP Layer alebo internet layer). Pre smerovanie využíva IP adresy k identifikácii sieťových zariadení a prenosový protokol sa nazýva IP protokol (Internet Protocol).



Obrázok 2.1: Model OSI a TCP/IP

2.1.2 IPv4 a IPv6

Ako som už uviedol vyššie, protokol TCP aj UDP používa na svojej Internetovej vrstve IP protokol. Identifikácia zariadenia v sieti je nutnosťou. K identifikácii zariadenia na internetovej vrstve sa používa IP adresa. V súčastnosti sa používajú dva typy IP protokolov – verzie 4 a verzie 6. U verzie štyri je IP adresa 32-bitová, avšak u verzie šesť je adresa 128-bitová.

V súčastnosti je stále najpoužívanejší IP protokol verzie 4, avšak všetky adresy IPv4 sú v súčastnosti vyčerpané, respektíve už nie sú dostatočné. Preto sa postupne rozvíja IPv6, ktorý dokáže adresovať viac zariadení.

V tejto podkapitole som vychádzal prevažne z [13].

2.2 Služba DNS

V tejto podkapitole som vychádzal najmä z [4]. V predchádzajúcej podkapitole som uviedol, že v sieti je potrebné jedinečné identifikovanie sieťové rozhrania. Takýmto identifikárom je IP adresa, či už verzie štyri alebo šesť. Pre človeka je jednoduchšie si zapamätať meno nie súbor čísiel. Primárne z tohoto dôvodu vznikla služba DNS, anglicky *Domain Name System*.

Pred zavedením DNS, v 70. rokoch 20. storočia, ARPAnet bola malá komunita s malým počtom pripojených zariadení, rádovo stovky. V tejto dobe sa pre preklad doménových mien používal súbor *HOST.TXT*. Súbor bol spravovaný organizáciou NIC (Network Information

Center), bol periodicky aktualizovaný. S pribúdajúcim počtom zariadení v ARPAnet súbor narastal a bol už ťažko spravovateľný.

A tak v roku 1983 Paul Mockapetris vyvinul koncept distribuovanej databáze doménových mien. Systém bol navrhnutý tak, aby bol dostupný širokej škále protokolov a aplikácií a zajistil jednoduchý prístup k informáciam uloženým v databáze.

Primárnou úlohou služby DNS je prevod doménových mien na IP adresy a opačne. Jedná sa o distibuovanú databázu, to umožnuje kontrolu, či spravovanie celej databázy a taktiež umožnuje prístup k datám cez celú sieť prostredníctvom klient-server modelu. Programy takzvané menné servery, anglicky *name servers*, tvoria v tejto schéme serverovú časť a obsahujú informácie o niektorých segmentoch tejto databázy. Druhú časť – klienta tohto modelu reprezentujú resolvery, anglicky *resolvers*, ktoré sa dotazujú na informácie, ktoré obsahujú menné servery.

Služba alebo systém DNS sa skladá z troch základných častí a to:

- priestoru doménových adries, anglicky *Domain Name Space*,
- serverov DNS a
- resolveru.

2.2.1 Infraštruktúra DNS

Ako už bolo spomenuté na začiatku tejto kapitoly, že služba DNS sa skladá z priestoru doménových mien, serverov DNS a resolveru, v tejto podkapitole tieto časti DNS podrobnejšie rozpíšem.

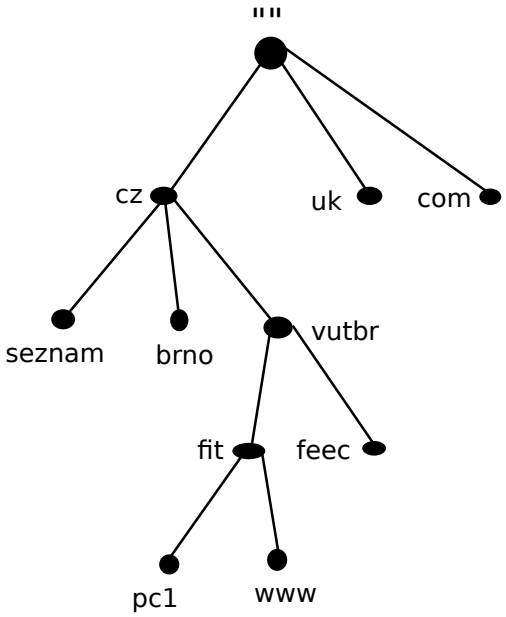
Priestor doménových mien je hierarchicky usporiadaná databáza, spôsobom koreňového stromu (obrázok 2.2), to zaručí efektívnosť a rýchlosť vyhľadávania. Strom má na vrchole jeden koreň (anglicky *root*), ktorého názov je reťazec nulovej dĺžky. Každý uzol je pomenovaný textovými reťazcami (bez bodiek) maximálnej dĺžky 63 znakov. Uloženie a vyhľadávanie doménového mena prebieha cestou od listu ku koreňu stromu, z toho vyplýva, že *doménové meno* je potom cesta od uzla ku koreňu stromu. Doména je podstrom priestoru doménových mien. Doménové meno domény je rovnaké ako doménové meno uzla na vrchole domény. Napríklad na obrázku 2.2 vrchol domény vutbr.cz je uzol pomenovaný vutbr.cz. Každé doménové meno v podstrome je časťou domény.

Doména je vzdialenosti jeden od koreňa stromu sa nazýva doména prvej úrovne (anglicky *TLD – Top Level Domain*), doména vzdialená dva od koreňa stromu sa nazýva doména druhej úrovňe, alebo tiež subdoména atď.. Plné, absolútne doménove meno, anglicky *FQDN – Fully Qualified Domain Name*, je postupnosť názvov uzlov oddelených bodkami, ako príklad uvediem „www.fit.vutbr.cz.“. V prípade stromu na obrázku 2.2, domény prvej úrovne sú *cz*, *uk* a *com*.

Je potrebné zdôrazniť, že doménové mená sú len indexy do DNS databáze.

Servery DNS (anglicky *name servers*) uchovávajú informácie o priestore doménových mien, a tiež majú zodpovednosť za odpovede na dotazy smerujúce na DNS databázu. DNS servery majú kompletné informácie o časti priestoru doménových mien, táto časť sa tiež nazýva *zóna*. Tieto informácie, datá získavajú z lokálneho súboru alebo ich načítavajú z iného serveru. Ako som vyššie uviedol, že server je zodpovedný za určitú zónu, takýto server sa tiež nazýva *autoritatívný* pre túto zónu.

Špecifikácia DNS [10] definuje dva typy serverov a to primárny a sekundárny. Obidva primárny a sekundárny sú autoritatívne pre danú zónu. Ďalším druhom môže byť záložný alebo caching-only server.



Obrázok 2.2: Príklad stromovej štruktúry DNS

Primárny server obsahuje úplné záznamy o doménach, ktoré spravuje, záznamy sú uložené v lokálnom v súbore. Na tomto serveri samotné data vznikajú a tiež v prípade zmeny doménového mena sa dáta musia editovať na primárnom servery. Každá doména má práve jeden primárny server.

Sekundárny server sa tiež nazýva *slave* a je kópiou primárneho. Čiže dáta získava, aktualizuje z primárneho DNS servera a tiež plní úlohu jeho zálohy. Databáza, ktorá obsahuje konkrétne domény/subdomény je tiež uložená v súbore, ale má názov zónový súbor. Proces prenosu súborov z primárneho na sekundárny server sa nazýva prenos zón alebo anglicky zone transfer. Server je tiež autoritatívny server pre danú doménu. Každá doména má minimálne jeden sekundárny DNS server.

Caching-only server pracuje ako proxy server – sprostredkovateľ medzi klientom a cieľovým DNS serverom. Jeho úlohou je prijať dotaz od klienta a preposlať dotaz na ďalšie DNS servery. Odpoveď si rozumne uloží pre dalšie použite v budúcnosti, takže slúži ako vyrovnávacia pamäť (cache memory). Odpovede od záložného DNS serveru nemusia byť aktuálne alebo úplné, hovoríme že poskytuje neautoritatívnu odpoveď, na druhú stranu urýchluje celý proces rezolúcie doménového mena.

Vyššie som uviedol, že DNS systém funguje na schéme klient-server, zatiaľ, čo predchádzajúce časti boli súčasťou servera v tejto schéme, **resolver** je na strane klienta, je to klientský program. Uživateľské programy, ktoré požadujú určité informácie z priestoru doménových mien používajú práve resolvery, lepšie povedané pristupujú k týmto datám pomocou resolverov. Resolvery sa starajú o následujúce úlohy a to:

- posielať dotazy na DNS servery,
- interpretovať odpovede od servera a
- odovzdať získané informácie aplikácii, ktorá ich požadovala.

Pre resolver je nutnosť mať prístup aspoň k jednému DNS serveru. V prípade, že server nepozná odpoveď na dotaz, pošle mu odkaz na ďalší DNS server. Z predchádzajúcich viet

je zrejmé, že resolver môže posielať dotazy aj na iné DNS servery.

Proces hľadania odpovedí v systéme DNS sa nazýva *rezolúcia*. Technicky stačí aby každý server poznal adresu a doménové meno koreňového DNS servera na to aby sa dostal k ľubovolnému DNS serveru (už spomenutá stromová štruktúra).

Koreňové servery DNS (anglicky *root nameservers*) sú autoritatívne servery DNS pre každú doménu najvyššej úrovne TLD spomenuté v podkapitole 2.2.1. V prípade dotazu na koreňový server DNS ohľadne akéhokoľvek doménového mena odpovie buď priamo hľadanou odpoveďou alebo vráti adresu servera DNS, na ktorom sa daná informácia vyskytuje.

Koreňové servery sú nutné pre fungujúci systém DNS, pretože zväčša rezolúcia začína práve u nich. V prípade výpadku všetkých koreňových serverov na dlhšiu dobu by celý proces rezolúcie zlyhal. Preto je vo svete až 13 koreňových serverov rozmiestnených po celom svete. Záťaž každého koreňového servera je rozložená na viacej strojov.

2.2.2 Priebeh rezolúcie

Existujú dva typy dotazov – *rekurzívny* a *iteratívny*. V následujúcich odstavcoch sa im budem podrobnejšie venovať.

U *rekurzívneho dotazu* resolver pošle dotaz na konkrétny DNS server, ten musí odpovedať požadovanou informáciou alebo chybou. V prípade, že tento server nie je autoritatívny pre danú doménu a pozná požadovanú informáciu odpovie neautoritatívnou odpoveďou. Avšak, keď požadovanú informáciu neobsahuje, server sa musí spýtať ďalších autoritatívnych serverov na odpoveď. V hľadaní odpovede dodržuje stromovú hierarchiu DNS – od koreňa po požadovaný uzol. Server hľadajúci odpoveď môže poslať na ďalšie servery znovu iteratívny alebo rekurzívny dotaz.

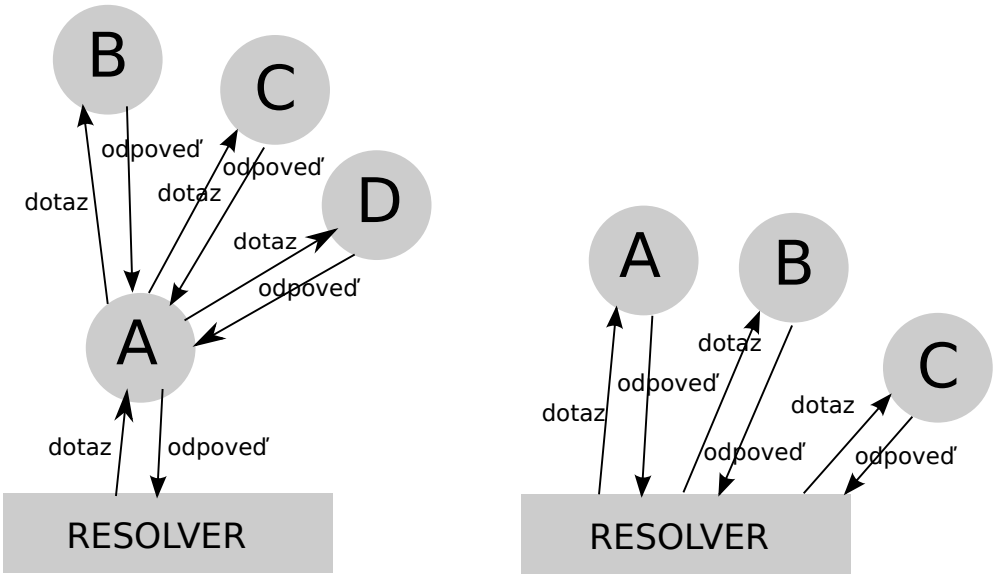
U *iteratívneho dotazovania* na server, server poskytne najlepšiu odpoveď akú môže. To znamená, že sa pozrie do lokálnej databázy, a keď pozná odpoveď tak ju vráti, inak vráti adresy serverov, ktoré sú danej hľadanej adrese najbližšie.

Príklad rekurzívneho a iteravného dotazu je zobrazený na obrázku 2.3, kde na prvom obrázku server *A* obdrží rekurzívny dotaz od resolveru, následne sa dotazuje iteratívne na odpoveď serverov *B, C, D*, kým nezíska odpoveď. Na druhom obrázku sa resolver pýta postupne serverov na odpoveď.

Protokol DNS

Komunikácia s DNS systémom je popísaná štandartom RFC 1035 [11]. Pre komunikáciu sa používa *protokol DNS*, ktorý používa transportný protokol UDP a port 53. Veľkosť paketu je obmedzená na 512 bytov. V prípade väčšej správy je potrebné ju rozdeliť na viacero paketov a nastavit bit TC(TrunCation) v hlavičke protokolu DNS. Paket DNS sa zkladá z piatich zložiek a to:

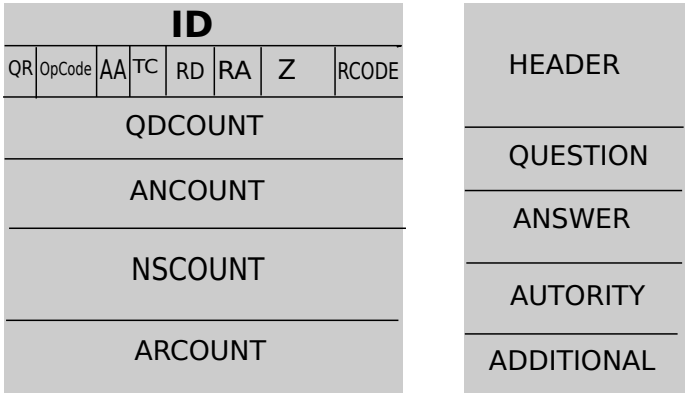
- hlavičky (anglicky *header*),
- časti obsahujúcej dotaz (anglicky *question*),
- časti pre odpoveď (anglicky *answer*),
- informácie o autoritatívnych serveroch – záznam NS (anglicky *autority*) a
- časti pre dodatočné informácie (anglicky *additional*).



Obrázok 2.3: Príklad rekúrzivného a iteratívneho dotazovania

Grafické zobrazenie štruktúry paketu (zprávy) DNS a hlavičky je zobrazené na obrázku 2.4.

Hlavička je povinná časť v DNS pakete a má pevnú dĺžku, obsahuje informácie o ostatných častiach paketu, ktoré su v ňom zahrnuté ako napríklad, keď sa jedná o dotaz na DNS tak nebude zahrnuté časti paketu pre odpoveď, pre autoritatívne servery a pre dodatočné informácie. Z tohoto vyplýva, veľkosť paketu pre dotaz je menšia ako veľkosť paketu pre odpoveď. Taktiež hlavička obsahuje informácie o aký typ paketu sa jedná – dotaz alebo odpoveď a iné.



Obrázok 2.4: Štruktúra hlavičky a zprávy protokolu DNS

2.3 Útoky typu DoS a služba DNS

Už od počiatkov internetu sa bolo treba zaoberať ochranou pred rôznymi druhmi útokov, či sa jednalo o útoky zamerané na zisk určitých informácií, hesiel a podobne, alebo útoky zamerané na vyradenie určitej služby, či systému. Existuje široké spektrum možných ky-

bernetických útokov, avšak táto práca je zameraná na jeden špecifický z nich a to na útok DNS amplification. Tento útok je práve útokom typu DoS alebo DDoS s využitím služby DNS.

2.3.1 Útoky DoS a DDoS

Hoci DoS a DDoS útoky sú v existencii v podstate od počiatku internetu, známe, lepšie povedané populárne sa stali pár rokov späť, a to konkrétne v roku 2010/2011, keď sa stali primárnou útočiacou metódou známej hackerskej skupiny Anonymous. Tento fakt vyústil v zistenie, že väčšina spoločností a organizácií nebola pripravená na tento druh útoku. V tejto podkapitole bude spomenutý základný popis týchto dvoch typov útokov, informácie som čerpal z [14].

Útok *DoS – Denial of Service*[1] je typ útoku, ktorého cieľom je zabezpečiť nedostupnosť služieb na počítači, ktoré by inak boli dostupné. Najčastejšie je tento útok zameraný na šírku pásma alebo na konektivitu. Útok zameraný na šírku pásma zaplavuje (flood) sieť velkým počtom paketov, a tým spotrebuje všetky dostupné zdroje, čo spôsobí, že uživateľská požiadavka nemože byť vybavená. Na druhú stranu, útok zameraný na konektivitu zaplaví počítač veľkým počtom požiadavkov na pripojenie, čo spôsobí spotrebovanie všetkých zdrojov operačného systému a nemožnosť spracovávať ďalšie požadavky od uživateľa.

Útok *DDoS – Distributed Denial of Service*, už podľa názvu napovedá, že sa jedná o distribuovaný útok. Tento typ útoku používa viacero počítačov pri vytváraní koordinovaného DoS útoku na jeden, či viacero cieľov. DDoS útok môže využívať technológiu klient-server k inicializácii tohto typu útoku. Master (arbiter) program beží na jednom počítači a v požadovanom čase komunikuje s rôzným počtom agentských programov, ktoré bežia na rôznych počítačoch kdekoľvek v rámci internetu. Akonáhle agent obdrží príkaz na inicializáciu útoku zaútočí. Takýto koncept združených škodlivých uzlov v internete riadených arbitrom sa nazýva *botnet*.

Princíp oboch typov útoku je znázornený na obrázku 2.5.

2.3.2 Útok DNS amplification

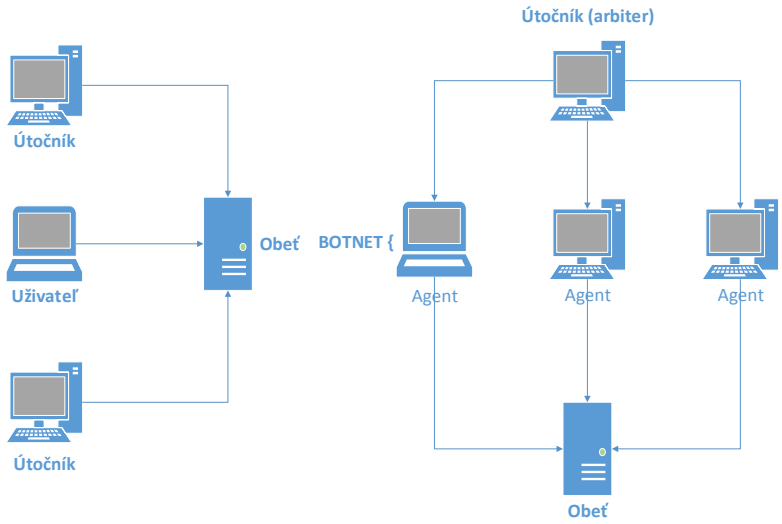
Útok DNS amplification patrí do skupiny DoS/DDoS útokov. Na rozdiel od klasických DoS útokov našel priestor pre zlepšenie, zefektívnenie tohto typu útoku. Týmto zlepšením je práve zneužitie služby DNS. Avšak tento typ útoku je hlavným zameraním tejto práce, je mu venovaná celá samostatná následujúca kapitola 3.

2.4 NetFlow

Nakoľko táto práca je zameraná na detekciu útoku DNS amplification z pasívnej analýzy DNS prevádzky, je potrebné vybrať nástroj pre monitorovanie siete, na ktorom by následná detekcia bola založená. Asi najpoužívanejším nástrojom tohto typu je práve **NetFlow**.

NetFlow bol vyvynutý firmou Cisco Systems pre možnosť monitorovania siete. Táto služba ponúka administrátorom informácie o IP toku v ich sieťach v podobe tokových (flow) dát. Tieto datá zachytávajú sieťové prvky ako prepínače (switches) a smerovače (routers) a exportujú ich do kolektorov. Tok (flow) je definovaný ako jednosmerná sekvencia paketov zdieľajúcich niektoré spoločné vlastnosti, ktoré prejdú cez sieťový prvok za určitý časový interval. Práve tieto prvky ich zachytávajú a exportujú do externého zariadenia nazývaného

[1] voľný slovenský preklad odoprenie služby



Obrázok 2.5: Princíp útokov DoS a DDoS

NetFlow kolektor. Záznamy typu NetFlow obsahujú rôzne informácie, napríklad IP adresy, počet paketov a bytov, porty, časové pečiatky a mnohé iné [5]. Príklad flow záznamu je na obrázku 2.6. Niektoré časti z NetFlow záznamov sú pre detekciu útoku nápomocné, konkrétne ide o tieto:

- IP adresy – zdrojové aj cieľové,
- časy DNS dotazov a časy DNS odpovedí,
- veľkosti DNS dotazov a DNS odpovedí
- a zdrojové aj ceľové porty.

V tejto sekcii som čerpal z [5].

Date flow start	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Packets	Bytes	Flows
2013-12-01 13:59:27.671	0.000	UDP	156.104.71.60:17984	->	50.174.134.35:53	1	69	1
2013-12-01 13:59:26.302	0.000	UDP	d5ff:91..:ff:137.21012	->	20fe:71..44:c205.53	1	91	1
2013-12-01 13:59:31.298	0.000	UDP	194.99.194.19:61035	->	218.14.245.250:53	1	80	1
2013-12-01 13:59:32.187	0.000	UDP	200.95.212.218:19828	->	192.108.140.20:53	1	73	1
2013-12-01 13:59:26.542	0.000	UDP	40.154.12.198:50519	->	145.194.176.30:53	1	57	1
2013-12-01 13:59:05.737	24.986	UDP	156.225.51.245:32768	->	192.32.193.225:53	3	292	1

Obrázok 2.6: Príklad NetFlow záznamu zobrazeným nástrojom nfdump

Kapitola 3

Analýza útoku DNS amplification

Cieľom tejto kapitoly je oboznámiť čitateľa s útokom DNS amplification, ktorý je známy tým, že sa jedná o efektívnejšiu a sofistikovanejšiu verziu klasického DDoS (Distributed Denial of Service) alebo DoS (Denial of Service) útoku. Hovorím sofistikovanejším a efektívnejším, pretože využíva DNS službu na *zosilnenie (anglicky amplification) útoku*. Je známe, a už bolo spomenuté v predchádzajúcich kapitolách, že dotaz na DNS je mnohonásobne menší ako odpoveď (relatívne malý dotaz môže generovať veľkú odpoveď) a práve tento fakt je pri tomto útoku zneužitý.

3.1 Útok DNS amplification

Útok *DNS amplification* je útok typu DDoS alebo DoS. Narozdiel od klasického DDoS (DoS) útoku využíva určitú službu na internete – v tomto prípade je zneužitou službou práve DNS. Útoky, ktoré využívajú služby dostupné na internete sa nazývajú *reflection útoky*[1], útok DNS amplification nie je výnimkou. Možnost reflekcie je dosiahnuteľná pomerne jednoducho, stačí zamaskovať (spoof) zdrojovú IP adresu adresou obete v pakete, ktorý sa dotazuje na službu v rámci internetu, v tomto prípade DNS. Táto metóda sa anglicky nazýva *IP spoofing* [12, 15, 8].

K zamaskovaniu alebo podvrhnutiu (spoofing) zdrojovej IP adresy sa útočník dostane pomerne jednoducho a to z dôvodu použitia protokolu UDP, ktorý neoveruje IP adresu ako protokol TCP. Táto metóda napomáha útočníkovi aj v tom, že je ťažko vystopovateľný, nakoľko je extrémne náročné získať jeho skutočnú zdrojovú IP adresu [9].

Avšak útok DNS amplification využíva ďalšiu možnosť pre zlepšenie, zefektívnenie útoku, a tou je *zosilnenie (anglicky amplification)*. V podkapitole 2.2.2 zameranej na komunikáciu s DNS som už naznačil, že veľkosť DNS paketu je premenlivá, presne toho tento typ útoku využíva. Tento zosiľňujúci faktor (amplification factor) sa vyznačuje tým, že DNS odpoveď (response) je mnohonásobne väčšia ako DNS dotaz (query, request), to je dané tým, že v prípade dotazu na DNS nebudú zahrnuté tieto časti DNS paketu: pre odpoveď, pre autoritatívne servery a pre dodatočné informácie, ktoré by boli zahrnuté v prípade odpoveďe viac v kapitole 2.2.2. Vzťah medzi dotazom a následnou odpoveďou sa nazýva *zosilňujúci faktor (amplification factor)*, ktorý je možno definovať následujúcou formulou [12, 15, 8]:

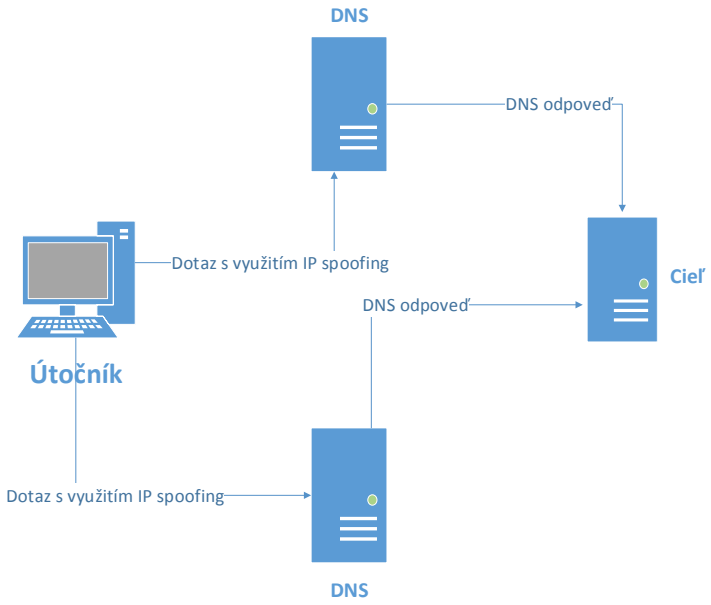
$$ZF = \frac{veľkosť(odpovede)}{veľkosť(dotazu)}$$

[1] voľne preložené z angličtiny do slovenčiny útoky odrazom

Služba DNS je skopná vygenerovať až 4096 bajtov ako odpoveď, je to dané rôznymi rozšíreniami tejto služby ako napríklad DNSSEC. V takom prípade útočník môže poslať iba relatívne malý a cielený dotaz na DNS a odpovede dosiahnu mnohonásobnú veľkosť dotazu. Útočník má viacero možností ako ovplyvniť výsledný zosilňujúci faktor, to však záleží na schopnostiach aké odpovede dokáže využiť:

- **Bežné DNS odpovede** – väčšinou 3 až 4 krát väčšie ako dotazy. Jedná sa o legitímne často používané dotazy na DNS, ktorých odpoveď je najčastejšie uložená v cache servery.
- **Vyskúmané DNS odpovede** – útočník sa snaží dopátrať k legitímnym odpovediam, ktoré mu prinesú najlepší efekt.
- **Vytvorené DNS odpovede** – v prípade slabo zabezpečeného DNS servera, ktorý môže útočník zneužiť, a tak si zaistiť maximálnu veľkosť odpovede, čiže 4096 bajtov.

V tomto odstavci som čerpal z [9].



Obrázok 3.1: Útok DNS amplification

Možný scenár útoku je následovný: útočník pošle reťazec malých dotazov z infikovaných počítačov na autoritatívne servery alebo len DNS server. Normálne by DNS odpovedal naspäť počítaču resp. resolveru, ktorý vytvoril daný dotaz, avšak IP adresa je zamaskovaná adresou obete, tým pádom všetky odpovede smerujú na obeť.

3.2 Príznaky útoku

Pri tomto type útoku je poslané veľké množstvo paketov z určitého počítača alebo počítačov s rôznymi zdrojovými portami na DNS server (port 53). Tieto pakety sú obvykle malej

veľkosti, avšak vedúce k výrazne veľkej odpovedi. Keď už boli poslané dotazy je zrejmé, že bude rovnaké množstvo odpovedí smerujúcich práve na cieľ, konkrétne na porty, z ktorých bol dotaz poslaný. Tieto pakety sú tiež výrazne veľké. Hoci väčšinou útok je smerovaný z rôznych zdrojových portov môže sa stať, že útok bude smerovaný z jedného fixného portu.

Predchádzajúci popis bol najoptimálnejší a najefektívnejší. Avšak útok od útoku sa môže meniť. Útočník nemusí vždy poslať taký dotaz, ktorý generuje veľkú odpoveď, môže nastať situácia, že dotaz bude smerovaný práve k malej odpovedi alebo k premenlivej veľkosti odpovede. Tak isto pre útok môže byť použité malé množstvo DNS serverov, radovo jednotky až desiatky, avšak je tu možnosť použitia tisíce serverov. Ako som už spomenul, každý útok môže byť špecifický a výrazný rôznymi vlastnosťami [7].

Ďalším z príznakov je relatívne veľká podobnosť veľkostí dotazov, respektíve odpovedí. Týmto myslím, že veľkosti jednotlivých dotazov, respektíve odpovedí budú takmer nemenné a každý s týchto dotazov si bude podobný. Príklad takýchto podozrivých dotazov je v tabuľke 3.1 a podozrivých odpovedí je v tabuľke 3.2

```
Date flow start     Duration Proto Src IP Addr:Port Dst IP Addr:Port  P B  F
2013-10-08 06:17:43.847 0.000 UDP 177.189.225.3:43619 -> 4.62.5.23:53 1 71 1
2013-10-08 06:17:46.283 0.000 UDP 177.189.225.3:6544  -> 4.62.5.23:53 1 71 1
2013-10-08 06:17:46.318 0.000 UDP 177.189.225.3:16299 -> 4.62.5.23:53 1 71 1
2013-10-08 06:17:52.670 0.000 UDP 177.189.225.3:53376 -> 4.62.5.23:53 1 71 1
2013-10-08 06:17:52.892 0.000 UDP 177.189.225.3:32304 -> 4.62.5.23:53 1 71 1
```

Tabuľka 3.1: Príklad podozrivých dotazov

```
Date flow start     Duration Proto Src IP Addr:Port Dst IP Addr:Port  P   B  F
2013-10-08 06:17:46.401 0.000 UDP 4.62.5.23:53 -> 177.189.225.3:53401 3 4031 1
2013-10-08 06:17:46.431 0.000 UDP 4.62.5.23:53 -> 177.189.225.3:53401 3 4031 1
2013-10-08 06:17:47.123 0.000 UDP 4.62.5.23:53 -> 177.189.225.3:50218 3 4031 1
2013-10-08 06:17:47.302 0.000 UDP 4.62.5.23:53 -> 177.189.225.3:57488 3 4031 1
2013-10-08 06:17:47.361 0.000 UDP 4.62.5.23:53 -> 177.189.225.3:54591 3 4031 1
```

Tabuľka 3.2: Príklad podozrivých odpovedí

Dáta v tabuľkách 3.1 a 3.2 pochádzajú z tokových dát a sú zobrazené nástrojom `nfdump`[2]. Je treba podotknúť, že tieto útoky nepozostávali len z piatich tokov, ale takýchto podobných tokov tam bolo viacero.

3.3 Scenáre útoku

V tejto podkapitole sa zameriam na možné scénare útoku v prípade použitia detekcie na základe tokových dát. V takejto situácii vznikajú tri stupne dostupnosti informácií a to:

- k dispozícii sú informácie (dáta) o dotaze i odpovedi,
- k dispozícii je len informácia o dotaze alebo odpovedi a
- k dispozícii nie sú informácie ani o dotaze, ani o odpovedi.

[2] http://nfdump.sourceforge.net/

V prvom prípade, čiže ak sú k dispozícii obidve informácie, čo znamená, že dotazovaný DNS server(y) ležia buď v sieti, na ktorej hranici sa toky zachytávajú, alebo mimo nej. Avšak, aby boli zachytené dotazy aj odpovede je potrebné situovať aj pozíciu útočníka a cieľa, v prvom prípade (DNS server leží v sieti) sa útočník aj cieľ nachádzaju mimo sledovanej siete, v druhom prípade (DNS server je mimo siete) sa útočník aj cieľ nachádzajú v sledovanej sieti. Na obrázku 3.2 sú to prípady D a G.

V druhom prípade sú možnosti mať informácie o dotaze alebo o odpovedi. Ak sú dostupné informácie len o dotaze, čo v skutočnosti znamená situovanie útočníka v pozorovanej sieti alebo mimo nej, ale táto skutočnosť nie je postačujúca pre zachytenie len dotazu. Je potrebné určiť polohu DNS servera a cieľa – ak útoční leží v sieti, DNS server a cieľ musia byť mimo tejto siete; ak útočník je mimo siete, DNS server a cieľ musia ležať v monitorovanej sieti. Na obrázku 3.2 je to možnosť E, respektíve C.

Dostupnosť iba odpovede znamená polohu útočníka a DNS servera mimo sieť a cieľ situuje do pozorovanej siete (obrázok 3.2 prípad A) – dotaz nikdy neprekročí hranice siete. Avšak existuje aj pravý opak tohto scénara – útočník a DNS server sa nachádzajú v sieti a cieľ je mimo nej, taktiež dotaz nikdy neprekročí monitorovanú sieť (obrázok 3.2 prípad F).

Posledný prípad, kde nie je možné zaznamenať ani dotaz, ani odpoveď, naznačuje polohu všetkých troch komponent mimo, respektíve vrámci siete (obrázok 3.2 prípad B, respektíve H).

Za zmienku stojí fakt, že podľa doporučenia BCP 38 (Best Current Practice)[6] by sieť nemala povolovať pakety zo zamaskovaných (spoofed) IP adries, ktoré vchádzajú alebo odchádzajú zo siete. To znamená, že napríklad paket so zamaskovanou zdrojovou IP adresou vrámci danej siete by nemal vchádzať do takej siete, a tak isto ak zdrojová IP adresa je zamaskovaná adresou, ktorá nie je v rámcie danej siete by nemala takúto sieť opustiť. Technicky to znamená, že scénare B a E by sa nemali vyskytovať. Avšak, je to len doporučenie, a to neznamená, že všetky siete sú implementované podľa BCP 38.

Na záver tejto podkapitoly je potrebné zdôrazniť, že uvedené scénare sa môžu kombidovať. Útoky môžu byť veľmi rozmanité, napríklad útok používa viacero DNS serverov, z ktorých niektore ležia v sieti a niektoré sú mimo siete. Uvedené scenáre vychádzaju z práce [7].

3.4 Možnosti detekcie

V tejto podkapitole bude čitateľovy ponúknutý pohľad na čo je potrebné sa zamerať v prípade detekcie útoku DNS amplification.

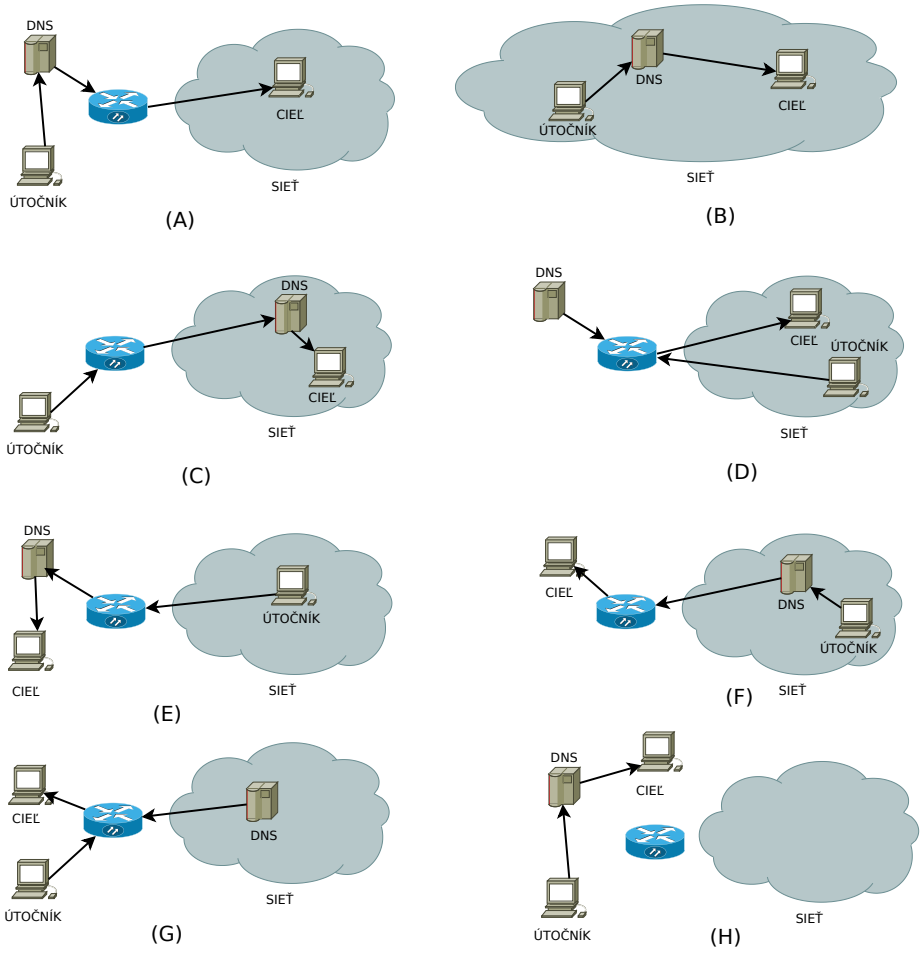
Priestor pre detekciu samozrejme existuje, dokonca sa útok dá detekovat pomocou tokových dát, čiže nie je potreba kompletné pakety a postačia iba tokové informácie. V predchádzajúcich podkapitolách tejto kapitoly som sa zaoberal príznakmi a scénarmi útokov DNS amplification a z informácií v nich uvedených vyplýva, že je dobré podrobnejšie zamerať sa hlavne na:

1. **Počet dotazov, respektíve odpovedí** poslaných z jednej IP adresy za určitý časový interval je jeden z hlavných prvkov pre možnú detekciu. Z tejto informácie je možné zistiť, či sa jedná o legitímnu komunikáciu alebo o minimálne podozrivú. Samozrejme, ak týchto dotazov (odpovedí) je poslaných málo pravdepodobne sa nebude jednať o útok, avšak ak ich bude priveľmi na pomery danej siete ide minimálne o podozrivú komunikáciu, ale bez ďalšieho preskúmania by sa nemal urobiť definitívny záver, či sa jedná o útok.

2. **Veľkosť týchto dotazov, respektíve odpovedí** poslaných z jednej adresy je jeden z faktorov, které napomáhajú pri detekcii. V prípade veľkostí odpovedí od DNS ide o veľmi dôležitý faktor, a to z dôvodu, že cieľom útoku DNS amplification je využiť čo možno najväčšiu možnú odpoveď od DNS. Akonáhle veľkosti odpovedí budú neobvykle veľké, je veľká pravdepodobnosť, že sa nejedná o legitímnu komunikáciu a stáva sa podozrivou. Jednotlivá veľkosť dotazov nie je pri detekcii až tak potrebná, síce dotazy by mali byť malé aby zosilňujúci faktor bol čo najväčší. U dotazov je potrebné sa viacej zamerať na následujúce informácie.

3. **Podobnosť veľkostí týchto paketov**. Táto informácia je užitočná pre skúmanie hlavne dotazov na DNS. Automaticky generované dotazy budú poväčšinou veľmi podobnej veľkosti alebo sa nebudú líšiť vôbec. Táto skutočnosť sa dá využiť pre detekciu *reflection* časti útoku. U odpovedí môže mať útok za následok podobnú situáciu, avšak tam sa javí toto skúmanie pomerne zbytočné a postačujúca je informácie o veľkosti odpovedí.

Na informácie uvedené v tejto kapitole naväzuje aj výsledný návrh algoritmov detekcie útoku DNS amplification v následujúcej kapitole.



Obrázok 3.2: Rôzne scénare útoku

Kapitola 4

Návrh detektora útoku DNS amplification

Táto kapitola je zameraná na návrh aplikácie detektora a požiadavkov na túto aplikáciu. V následujúcej podkapitole nastolím a zhrniem požiadavky na výslednú aplikáciu. Zbytok tejto kapitoly bude už venovaný práve návrhu aplikácie samotnej.

4.1 Požiadavky na program

Pred samotným návrhom aplikácie je potrebné sa zamerať na samotné požiadavky na aplikáciu, z ktorých práve návrh vychádza.

Ako takmer u každej aplikácie sa kladie dôraz na kvalitnú, intuitívnu a jednoduchú interakciu aplikácie s uživateľom. Nakoľko výsledný program tejto práce nebude klásť až taký veľký nárok na interakciu a vstupy uživateľa, zvolil som typ *konzolovej aplikácie*, ciže bez grafického uživateľského rozhrania (GUI – Graphics User Interface). Avšak minimálná interakcia s používateľom bude v aplikácii potrebná, konkrétne v podobe argumentov samotného programu, z toho dôvodu je potrebné zvoliť argumenty, ktoré budú pre uživateľa intuitívne.

Medzi ďalšie požiadavky patrí aj prehľadný a názorný výstup somotného programu. Výstup programu by mal obsahovať samostatný výstup respektíve *názorne oddelený výstup* detekovaného útoku pre reflection časť a amplification časť útoku. Taktiež výstup by sa mal prispôsobovať argumentom príkazovej riadky zadanými používateľom. Výstup by taktiež mal obsahovať informácie o podozrivom toku ako sú: zdrojová a cieľová adresa, porty (hlavne u DNS), smer toku, veľkosť v bajtoch, počet paketov a čas trvania, tieto informácie by mali byť znázornené rádoby tabuľkou, respektíve textom, ktorý je prehľadne formátovaný.

Zrejme do najväčších požiadavkov na program je možné zaradiť požiadavok na čo najpresnejšie *označenie toku za podozrivý*, ciže sa vyhnúť false-positive detekcii.

4.2 Návrh programu

Požiadavky na aplikáciu boli zhrnuté v predchádzajúcej podkapitole. V tejto podkapitole pozvolna prejdem k návrhu programu, taktiež spomeniem problémy, ktoré sa budú riešiť v implementácii detektora.

Pre implemntáciu detektora som sa rozhodol použiť metody *procedurálneho* programovania, nakoľko som zhodnotil, že použitie objektovo orientovaného modelu je zbytočne zložité, keď sa v podstate jedná iba o nástroj pre detekciu útoku DNS amplification.

V prvej časti tejto podkapitoly som sa rozhodol venovat jednoduchšej časti návrhu, a to návrhu vhodných výstupov nástroja, v druhej polovici sa zameriam na zložitejšie veci, konkrétne na vnútornú reprezentáciu tokov a na návrh algoritmov detekcie.

4.2.1 Vizualizácia výstupu detektora

Požiadavky na výstup programu boli jasne stanovené na začiatku tejto kapitoly. V tejto podkapitole sa budem venovať práve výstupom.

Aby bol výstup jasný, zrozumiteľný a prehľadný je potreba ho sformátovať do takých formátov, ktoré tieto vlastnosti spĺňajú. V nástroji pre detekciu som sa rozhodol podporovať dva typy výstupov a to:

- formátovaným textom na štandartný výstup a
- výstupom do HTML súboru.

Klasický formátovaný text ma veľa nevýhod, ako sú nemožnosť zvýrazniť dôležité časti tohto výstupu, čo sposobuje neprehľadnosť v prípade veľkých a dlhých výstupoch. Na druhú stranu má aj veľkú výhodu, ktorou je jednoduchosť ďalšieho spracovania obyčajného formátovaného textu, z toho dôvodu je dobré takýto výstup v nástroji podporovať.

Ďalším možným výstupom je už spomenutý výstu do HTML[1]. HTML narozdiel od klasického formátovaného textového výstupu má výhodu v natívnej podpore zvýrazňovať prvky, tvorbu tabuliek, nadpisov a mnoho iných prvkov. Kaskádové štýly CSS posúvajú HTML dokument na inú úroveň. Kaskádové štýly ponúkajú rôzne možnosti štýlovania HTML dokumentu ako sú zmeny typov, farby a veľkostí písma (fontov), zmena pozadia rôznych prvkov a mnoho ďalších rôznych možností.

4.2.2 Nastavenie programu používateľom

Navrhovaný nástroj nebude patriť medzi programy alebo aplikácie, ktoré sú zamerané na uživateľské vstupy. Nástroj bude typu konzolovej aplikácie, čo bolo viackrát spomínané. Od uživateľa bude požadované základné nastavenie aplikácie formou argumentov programu.

Medzi nastaveniami by určite mala patriť možnosť nastavenia vstupných NetFlow súborov, ktoré by mali byť päť minútové, pretože algoritmus detekcie natívne pracuje práve s päť mínutovými NetFlow záznamami, túto skutočnosť som uviedol v závere podkapitoly 4.2.4, zemeranej práve na algoritmy detekcie. Súbory by mali taktiež na seba naväzovať, aby nenastávali situácie typu, že sa na vstupu objaví súbor s piatimi minútami z jedného dňa a následne súbor s piatimi minútami z druhého dňa. Najoptimálnejším riešením by bolo dať možnosť uživateľovi zadať začiatočný súbor a koncový súbor pre detekciu a očakávať, že má k dispozícii všetky potrebné data, čo by zabránilo potrebe spúšťať program pre každý súbor zvlášť. Riešenie tohto problému ale nepatrí do návrhu nástroja ale až do samotnej implementácie (následujúca kapitola).

Ďalšou z možností programu by mala byť voľba medzi hore uvedenými typmi výstupov programu. A v neposlednej rade by program mohol poskytovať možnosť zmeny respektíve

[1] HTML je hypertextový značkovací jazych (HyperText Markup Language) určený pre vytváranie internetových stránok[1].

nastavenia časového okna pre detekciu. Časovým oknom myslím, nastavenie dĺžky záznamov pre jednorázovu detekciu, napríklad pri zadaní časového okna na desať minút by sa zobrali dva päť-minútové súbory, ktoré na seba naväzujú a spustila by sa jednorázová detekcia pre agregované toky ktorých dĺžka je väčšia ako desať minút.

4.2.3 Vnútorna reprezentácia tokov

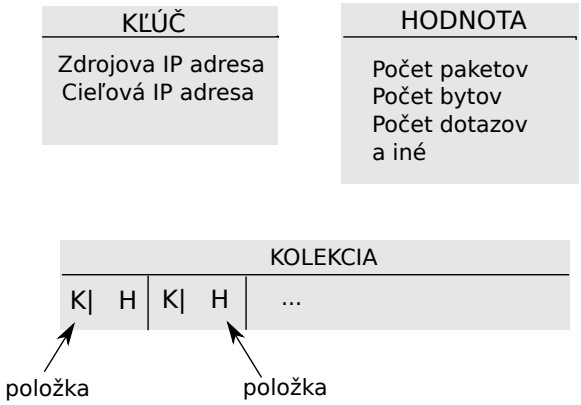
V tejto sekcii sa zameriam na návrh vnútornej reprezentácie NetFlow tokov. Je dobrým zvykom sa venovať návrhu vnútornej reprezentácie dát v programe ešte pred samotnou implementáciou daného problému. Jedná sa pomerne o dôležitú časť návrhu, preto som sa rozhodol tejto problematike venovať hneď zo začiatku samotného návrhu a implementácie nástroja pre detekciu útoku DNS amplification.

Je známe, že NetFlow dáta sú dosť rosiahle a veľké, z toho dôvodu je potrebné tieto dáta skladovať a reprezentovať v programe čo najjednoduchšie, najefektívnejšie a nejprehlednejšie (z pohľadu zdrojového kódu). Intuitívny nápad by bol použiť pre reprezentáciu toku v programe *štruktúru* alebo *objekt*, keďže NetFlow záznam pripomína štruktúru alebo objekt (obsahuje napríklad zdrojovú a cieľovú adresu, porty a ďalšie dáta viac v podkapitole 2.4 kapitoly 2).

Avšak je treba si uvedomiť, že tieto štruktúry tvorené z informácií z konkrétnych tokov, je nutné skladovat v pamäti programu. Tu by sa naskytla možnosť použiť určitú *kolekciu* dát, do ktorej by sa dané štruktúry uložili. Týmto by sa vyriešilo skladovanie záznamov o tokoch v programe ale z navrhnutých algoritmov uvedených v podkapitole 4.2.4 kapitoly 3.4 vyplýva, že NetFlow toky je potrebné pred detekciou agregovať pomocou zdrojovej a cieľovej adresy. No, ak by bola použitá kolekcia štruktúr NetFlow záznamov, by v prípade potreby vyhladania toku a agregácie s novým načítaným bolo neefektívne a neprakticke, v podstate by sa muselo prechádzať celou kolekciou a porovnávať IP adresy.

Z predchádzajúceho odstavca je zrejmé, že je potreba nájsť lepší spôsob uloženia záznamov o tokoch do pamäti, hlavne z dôvodu ľahšieho a efektívnejšieho vyhľadávania. Presne pre takýto prípad by sa hodila kolekcia, ktorá má možnosť uložiť dáta pod určitým kľúčom pre prípad ďalšieho vyhľadávania. Táto vlastnosť by náležala napríklad *hashovacej tabuľke* alebo *mapy* známej z programovacieho jazyka C++. V takomto prípade by bola zdrojová a cieľová adresa *kľúčom* a zbytok informácií o toku by bolo *hodnotou* v takejto kolekcii.

Na obrázku 4.1 je znázornená navrhovaná reprezentácia záznamov o toku a následná kolekcia týchto záznamov, kde každý záznam je tvorený kľúčom a hodnotou.



Obrázok 4.1: Názorný príklad reprezentácie záznamu o toku

4.2.4 Algoritmy detekcie

V tejto podkapitole rozpíšem návrh algoritmov respektíve návrh implementácie a možné problémy pri implementácii týchto algoritmov. Algoritmy detekcie sú založené na *tokových dátach* (NetFlow dátach)

Použité matematické konštrukcie

Obidva algoritmy využívajú *Exponenciálny kĺzavý váhový priemer (EWMA)*. Práve túto sekciu považujem za vhodnú pre oboznámenie čitateľa s týmto priemerom. Exponenciálny kĺzavý priemer alebo tiež EWMA (z anglického Exponentially Weighted Moving Average) jedná sa o štatistiku pre skúmanie trendov v závislosti na histórii. EWMA zahrnuje všetky namerané historické dáta a dáva im určitú váhu, tak isto dáva váhu aktualne nameranej hodnote. Vzorec pre výpočet tohto priemeru je v rovnici 4.1, viacej informácií je možné nájsť v [3].

$$EWMA_t = \lambda Y_t + (1 - \lambda) EWMA_{t-1} \quad \text{pre} \quad t = 1, 2, \ldots, n. \tag{4.1}$$

Kde

- Y_t je aktuálne nameraná hodnota v čase t,
- λ je EWMA faktor ležiaci $0 < \lambda \leq 1$, znamenajúci váhu historických dát,
- n je celkový počet nameraných hodnôt.

Pri implementácii tohto priemeru vzniká problém, pre správnu funkčnosť je potrebné určiť $EWMA_0$. Väčšinou ako počiatočná hodnota sa určí priemer meraných hodnôt, za určitý počiatočný úsek prebiehajúceho merania. V mojom prípade by to znamenalo, že napríklad by sa v prvých piatich minútach počítal priemer počtu paketov v agregovaných tokoch, za následok by to malo nemožnosť detekcie pre tento časový úsek alebo znovu prejdenie agregovaných tokov pre detekciu s už vypočítaným kĺzavým priemerom. Z pohľadu efektívnosti sa to práve nejaví ako najlepšie riešenie.

Ďalšie z možneho riešenia predošlého problému ako inicializovať počiatočnú hodnotu kĺzavého priemeru sa javí nastaviť $EWMA_0 = 0$. V takomto prípade by zo začiatku hodnota tohto priemeru nebola až tak presná, avšak tento nedostatok by sa vyriešil po načítaní menšieho množstva dát. Tento prístup netrpí nedostatkom vyššie uvedeného prístupu ohľadne nedetekovania útoku, respektíve dvojitému prechádzaniu tokov v inicializačnej fáze. Avšak tento prístup nie je bezchybný je tu možnosť menej presného určovania meranej hodnoty.

Taktiež v prípade kĺzavého priemeru je potrebné určiť EWMA faktor λ, ktorý určuje akú budú mať váhu historicky namerané hodnoty. Doporučuje sa použit hodnoty v rozsahu $\langle 0{,}2, 0{,}3 \rangle$.

Jeden z algoritmov, konkrétne algoritmus pre detekciu podozrivých DNS dotazov používa *smerodajnú odchylku (standard deviation)*. Smerodajná odchýlka je využívaná v teórii pravdepodobnosti a štatistiky. V podstate táto odchýlka vypovedá o tom, ako sa od seba skúmané veličiny lýšia, to znamená, čím je odchýlka menšia, tým je súbor skúmaných čísiel podobnejší. Vzorec pre výpočet smerodajnej odchylky σ je v následujúcej rovnici 4.2.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})^2} \tag{4.2}$$

Kde:

- N je celkový počet nameraných hodnôt,
- x_i je i-tá hodnota v súbore nameraných hodnôt a
- $\overline{x}$ je priemerná hodnota hodnôt v danom súbore.

Hoci rovnica smerodajnej odchylky 4.2 je úplná a funkčná, no z pohľadu efektívnosti a implementácie nie je vhodná. A to prevažne z dôvodu potreby aritmetického priemeru po celý čas výpočtu smerodajnej odchylky. Táto skutočnosť by znamenala, že v prípade postupnom načítavaní a agregácii tokov zo súbora, dvojité prechádzanie týchto tokov – prvý krát pre výpočet artimetického priemeru a po druhý krát pre výpočet danej smerodajnej odchylky.

Po skutočnostiach uvedených v predchádzajúcom odstavci bude potrebné sa zamyslieť nad lepším riešením. Tu sa naskytá možnosť vužitia úpravy rovnice 4.2 na rovnicu 4.3[1].

$$\sigma = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right) - \overline{x}^2} \tag{4.3}$$

Kde:

- N je celkový počet nameraných hodnôt,
- x_i je i-tá hodnota v súbore nameraných hodnôt a
- $\overline{x}$ je priemerná hodnota hodnôt v danom súbore.

U vzorca uvedeného vyššie je zrejmé, že z pohľadu efektivity nemá tento vzorec nedostatok, ktorý nadobudol vzorec 4.2. To znamená, že nie je potreba mať vypočítaný priemer hneď na začiatku, ale stačí ho vypočítať až na konci výpočtu smerodatnej odchylky. Samozrejme je nutné počítať časť výpočtu ale ten už je dobre realizovateľný pri postupnom načítaní a agregovaní tokov.

V tejto podkapitole som čerpal z následujúcich zdrojov – u exponenciálneho váhového kĺzavého priemeru z [3] a u smerodajnej odchylky z [2].

Algoritmus detekcie

V tejto sekcii sa zameriam na algoritmus detekcie útoku. Už som viackrát spomenul, že útok DNS amplification sa skladá z dvoch častí – *reflection* a *amplification*. Taktiež by som rád vytkol skutočnosť, že nie vždy je možné detekovať obidve časti, tento fakt som už popísal v predchádzajúcich kapitolách prevažne v 3.3.

Pri tvorbe týchto algoritmov som sa inšpiroval prácou [7]. Autor použil možnosti detekcie pomocou pevne daných prahov počtu paketov v toku. Taktiež stanovil a použil pevne daný prah, ktorý označí tok ako podozrivý hneď, keď je prekročený bez ďalšej kontoroly alebo skúmania toku. V tomto pôvodnom algoritme bola použitá agregácia NetFlow tokov, avšak len podľa jednej adresy a to:

- v algoritme zameranom na detekciu podozrivých dotazov agregoval podľa zdrojovej IP adresy a
- v algoritme pre detekciu podozrivých odpovedí agregoval toky podľa cieľovej IP adresy.

[1] Dôkaz tvrdenia, že rovnica 4.3 je upravená rovnica 4.2 je v [2]

Táto skutočnosť má za následok nemožnosť zistenia zdrojovej, respektíve cieľovej adresy.

Anlýzou tohto algoritmu som zistil, že vykazuje veľké množstvo false-positve[2]detekcií. Práve false-positive detekciam som sa chcel vyhnuť alebo ich aspoň zminimalizovať.

Algoritmus 4.1: Algoritmus detekcie podozrivých dotazov

```
1  double tresh;
2  bool attack = false;
3  if(ewma >= treshold2)
4          tresh = ewma;
5  else
6          tresh = treshold2;
7  if(flow.packets > tresh){
8          if(flow.num_request > (flow.packets/2)){
9                  if(standard_deviation() < 1){
10                         attack = true;
11                 }
12         }
13 }
```

Obidva tieto algoritmy očakávaju na vtupe tok ktorý je agregovaný pomocou zdrojovej a cieľovej adresy.

Algoritmus 4.1 sa zameriava na skúmanie podozrivých dotazov na DNS, ciže sa jedná o *reflection* časť celkovej detekcie. Ak má byť útok efektívny musí útočník generovať velké množstvo paketov, avšak to nie je podmienkov je potrebné sa zamerať aj na menšie útoky alebo na útoky, ktoré sú distribuované. Samozrejme, ak je útok menší nemá až takú efektivitu ale stále sa jedná o anomáliu, ktorá by nemala byť nezachytená. Práve z toho dôvodu na treťom až šiestom riadku zisťujem *prah* (treshold) počtu paketov. Práve tento práh je definovaný *exponenciálnym váhovým kĺzavým priemerom*, čo spôsobí adaptívnosť tohto práhu. Avšak, tento priemer môže byť za určitých okolností veľmi malý, čo by spôsobilo veľa *false-positive* detekcií, ktorým som sa chcel vyhnúť. Práve preto som definoval minimálny práh `treshold2`, ktorý je veľkosti 100.

V dalšej časti tohto algoritmu sa odohráva samotná detekcia. V tejto časti sa pracuje s *tokom* (flow), ktorý je agregovaný pomocou zdrojovej a cieľovej IP adresy. Najprv sa skontroluje počet paketov v danom toku ak ten počet paketov prekročí daný prah, prekročí sa k ďalšej kontrole. NetFlow dáta, ako už bolo viackrát spomenuté, sa združujú podľa určitých vlastností do jednoho toku, väčšinou sa jedná o zdrojovú adresu, port a cieľovú adresu, port. V takom prípade sa stratia informácie o jednotlivích dotazoch na DNS. Presne pre takýto prípad sa na riadku 8 zisťuje, či sú dotazy poslané z rôznych portov.

Ak v tomto pakete boli dotazy posielané z rôznych portov, prejde sa na ďalšiu časť tohto algoritmu pre detekciu. Táto časť je zameraná na veľkosti daných dotazov, respektíve na rozdiely veľkostí dotazov – je vypočítana *smerodajná odchýlka* (standard deviation) veľkostí dotazov na DNS. V prípade, že vypočítaná smerodajná odchýlka je menšia ako 1, čo znamená, že veľkosti dotazov sa takmer nemenia, je takýto tok označený ako *podozrivý*. Na rozdiel od pôvodného algoritmu som nepristúpil na detekciu tokov, ktoré boli poslané z fixného portu, a to z dôvodu nemožnosti preskúmať jednotlivé dotazy, a tým pádom určiť s väčšou presnosťou existenciu útoku.

[2]Detekovaný *false-positive* tok je taký tok, ktorý prejde detektorom ako podozrivý ale v skutočnosti sa o útok nejedná.

Algoritmus 4.2: Algoritmus detekcie podozrivých odpovedí

```
1  double tresh;
2  bool attack = false;
3  if(ewma >= treshold2)
4          tresh = ewma;
5  else
6          tresh = treshold2;
7  if(flow.packets > tresh &&
8          (flow.bytes/flow.packets) > treshold_avg_size){
9          attack = true;
10 }
```

Algoritmus detekcie podozrivých odpovedí 4.2 je už od pohľadu jednoduchší ako predchádzajúci algoritmus. Pravdepodobne je zrejmé, že ide o algoritmus pre detekciu amplification časti celkovej detekcie. Tento algoritmus taktiež využíva exponenciálny váhový priemer priemer pre definovanie prahu počtu paketov v toku, pri ktorom sa spustí detekcia. Na začiatku sa podobne ako v predchádzajúcom algoritme zistí, či hodnota vypočítaného priemeru nie je príliš malá, čo by mohlo spôsobiť nedetekovanie určitých podozrivých tokov. Minimálna hodnota pre práh je 100, táto hodnota sa nachádza v prementej `treshold2`.

Aby bol útok efektívny a využil čo najväčší *zosilňujúci faktor*(amplification factor) musí byť odpoveď od DNS čo najväčšia. Z tohoto dôvodu je ďalšia kontrola pred označením toku za podozrivý venovaná práve tejto skutočnosti. Ak *priemerná veľkost* paketu (v bajtoch) v tomto agregovanom toku prekročí práh `treshold_avg_size` je označený ako podozrivý. Po viacerých experimetoch sa osvedčilo tento práh stanovit na veľkost 900 bajtov.

Avšak veľkosť tohto práhu maximálnej priemernej veľkosti paketu súvisí aj so skutočnosťou, že maximálna prenosová jednotka (MTU – Maximum Transmission Unit) po sieti (Ethernet) je 1500 bajtov na paket. Ak by tento práh bol príliš malý vznikalo by veľa *false-positive* detekcií, ktoré nie sú žiadané. Na druhú stranu, ak by som práh stanovil príliš veľký nedetekovali by sa stredne silné alebo slabšie útoky. Na druhú stranu je potrebné dodať, že tento algortimus nemusí detekovať slabé útoky, ktorých veľkosť odpovede je menšia ako stanovených 900 bajtov.

Ná záver je potrebné uviesť, že uvedené práhy počtu paketov v NetFlow tokoch sú stanovené pre päť minútové NetFlow dáta. Nakoľko v predchádzajúcich podkapitolách návrhu nástroja bolo uvedené, že detektor by mal povolovať uživateľom stanoviť potrebnú dĺžku pre jednorázovú detekciu. V takom prípade je potrebné prahy prepočítať, ale je nutné podotknúť, že výsledky následnej detekcie sa môžu nakloniť na lepšiu aj horšiu stranu.

Kapitola 5

Implementácia výsledného nástroja pre detekciu

V predchádzajúcej kapitole 4 som sa venoval požiadavkom na program a návrhu programu, taktiež som uviedol problémy, ktoré sa budú musieť riešiť v následnej implementácii.

Práve táto kapitola je venovaná implementácii nástroja pre detekciu útoku DNS amplification. Nástroj som sa rozhodol implementovať v jazyku C++ a to z viacerých dôvodov. Jedným a veľmi významným dôvodom pre vybranie tohto jazyka je podpora alebo existencia knižníc pre prácu s NetFlow dátami. Druhým dôvodom je skutočnosť, že s týmto jazykom mám asi najviac zkušeností. V následujúcej podkapitole sa budem venovať vnútorným datovým štruktúram programu.

5.1 Vnútorné datové štruktúry programu

Už v návrhu programu bolo uvedené, že štruktúry alebo objekty sa hodia pre reprezentáciu NetFlow dát, čiže tokov. Avšak táto štrukturá nie je jedinou, ktorú som sa rozhodol použiť.

Štruktúru, ktorú som sa rozhodol implementovať ako prvú je určená pre *udržiavanie stavu aplikácie*. Táto štruktúra nesie názov `main` a je zobrazená v kóde 5.1.

Kód 5.1: Štruktúra určená pre udržiavanie stavu aplikácie

```
1 struct main {
2     int e_code;
3     char* start;
4     char* end;
5     unsigned int max_min;
6     bool details;
7     bool htmlout;
8 };
```

Kde:

- `e_code` je integer držiaci aktuálnu chybu v programe,
- `start` je reťazec, ktorý značí počiatočný NetFlow súbor pre načítanie,
- `end` je reťazec, ktorý značí posledný NetFlow súbor pre načítanie dát,

- `max_min` je bezznamienkový integer naznačujúci po koľkých načítaných minutách sa spustí detekcia,
- `details` je boolean hodnota naznačujúca, či uživateľ požaduje zobrazenie detailov útokov a
- `htmlout` je boolean hodnota naznačujúca, či uživateľ chce výstup do HTML súbora.

Ďalej je potreba implementovať štruktúru, ktorá reprezentuje toky. V návrhu programu boli uvedené, že toky je potrebné pre dobré vyhľadávanie ukladať do kolekcie, ktorá má tú vlastnosť, že je možné v nej vyhľadávať s pomocou kľuča. Takouto kolekciou v jazyku C++ je `map`[1].

Do takejto mapy je možné ukľadať páry kľuč – hodnota, v mojom prípade kľučom je objekt triedy `Flow_key_t`, ktorý má dva členy (members). Prvým z nich je zdrojová adresa (`srcadd`) a druhým je cieľová adresa (`dstadd`). Pomocou týchto dvoh hodnôť je možné vyhľadať v danej mape ktorýkoľvek tok. Pre hodnotu v tejto dvojici používam štruktúru `Map_value`, táto štruktúra je znázornená v kóde 5.2.

Kód 5.2: Štruktúra určená pre hodnotu v mape

```
1  typedef struct {
2      uint16_t srcport;
3      uint64_t packets;
4      uint64_t bytes;
5      uint32_t first;
6      uint32_t last;
7      unsigned long long num_request;
8      unsigned long long sum_size_sq;
9      int type;
10 } Map_value;
```

Kde jednotlivé položky štruktúry znamenajú následovné:

- `srcport` je 16 bitový bezznamienkový integer určujúci zdrojový port toku
- `packets` je 64 bitový bezznamienkový integer, ktorý zastáva celkový počet paketov v toku
- `bytes` je 64 bitový bezznamienkový integer reprezentujúci celkovú veľkosť paketov v bajtoch
- `first`, respektíve `last` určujúci čas prvého, respektíve posledného toku
- `num_request` reprezentuje počet dotazov v agregovanom toku
- `sum_size_sq` značí *mocninu* súm veľkostí jednotlivých tokov v bajtoch pre vypočítanie smerodajnej odchylky
- `type` značí typ útoku (reflextion alebo amplification) určený pre následný detekčný algoritmus

[1] http://www.cplusplus.com/reference/map/map/

5.2 Načítanie NetFlow dát alias funkcia `fill_flowmap`

Po implementovaní dátovej vrstvy som pokračoval v implementovaní načítania NetFlow dát. Pre načítanie dát som sa rozhodol použiť knižnicu `nfreader`. Táto knižnica je súčasťou nástroja `nfdump` určeného pre prácu s uloženými NetFlow dátami.

Z návrhu algoritmov pre detekciu je zrejmé použitie dvoch typov algoritmov – jeden pre detekciu reflection a druhý pre detekciu amplification časti útoku. Z vyššie uvedeného dôvodu som sa rozhodol pre existenciu dvoch typov máp pre uloženie tokových dát a to:

1. pre uloženie dotazov
2. pre uloženie odpovedí

Pre samotné načítanie/uloženie tokov zo súboru slúži funkcia `fill_flowmap`, ktorá má parametry hore uvedené mapy pre uloženie a dopredu otvorený NetFlow súbor. Pre iterovanie cez toky v súbore slúži funkcia `nf_next_record` zo spomenutej knižnice, ktorá naplní záznam o danom toku do štruktúry typu `master_record_t`, ktorá obsahuje prvky (informácie) o toku.

Ďalej je potrebné oddeliť toky súvisiace s DNS, a taktiež oddeliť dotazy na DNS a odpovede od DNS. Implementácia tohto problému je pomerne jednoduchá. V predchádzajúcich kapitolách som uvedol, že komunikácia s DNS prebieha na protokole UDP. Avšak táto skutočnosť nie je dostačujúca pre odhalenie DNS komunikácie. Tento problém sa dá vyriešiť pomerne jednoducho, jeden zo portov (zdrojový alebo cieľový) bude 53, pretože tento port je rezervovaný pre službu DNS. Čo sa týka oddelenia dotazov a odpovedí som vyriešil následovne:

- **Oddelenie dotazov** zabezpečuje kontrola cieľového portu, ktorý musí byť 53
- **Oddelenie odpovedí** je možné dosiahnuť opakom predchádzajúceho, čiže zdrojový port bude 53

Taktiež som sa rozhodol v tejto funkcii implementovať agregáciu tokov podľa zdrojovej a cieľovej IP adresy, nakoľko navrhnuté algoritmy túto agregáciu vyžadujú. Pri imlementácii tejto časti ma napadli dve riešenia – agregovať až načítane toky v mape alebo agregovať už pri načítavaní zo súboru. Nakoniec som sa rozhodol túto agregáciu vyriešiť druhým z týchto možností, pretože sa mi javil menej pamäťovo náročný.

5.3 Algoritmy detekcie

Táto podkapitola vychádza z návrhu algoritmov nachádzajúcich sa v predchádzajúcej kapitole. Navrhnuté algoritmy detekcie využívajú dva podporné výpočty – *exponenciálny váhový kĺzavý priemer (EWMA)* a *smerodajnú odchyľku (standard deviation)*. Obidva výpočty potrebujú určité podporné priebežné výpočty. Na účel udržiavania prebežných výpočtov sa hodia *globálne premenné*.

Túto technikou (globalnými premennými) som sa rozhodol implementovať vypočet exponenciálneho váhového kĺzavého priemeru. Pre tento priemer som určil dve globálne premenné:

1. `ewma1` slúži ako priemer pre reflection časť algoritmu a
2. `ewma2` slúži ako priemer pre amplification časť algoritmu.

V návrhu som uvedol, že je potrebné určiť EWMA faktor λ, ktorý je doporučený držať v rozmezí $\langle 0{,}2, 0{,}3\rangle$, v mojom prípade som zvolil 0,3. Tieto dva priemery sa počítajú až nad agregovanými tokmi za uživateľom definovaný časový interval.

Čo sa týka smerodajnej odchýlky, je potreba ju počítať pre každý tok individuálne. Z toho dôvodu štruktúra reprezentujúca informácie o toku (kód 5.2) obsahuje položku `sum_size_sq`, ktorá reprezentuje sumu mocnín veľkostí dotazov. Táto informácie je potrebná pre výpočet smerodajnej odchýlky pre daný agregovaný tok dotazov. Celková smerodajná ochýlka je počítaná funkciou `get_deviation`, ktorá má parametry: celkový počet dotazov, hore uvedenú sumu mocnín veľkostí dotazov a sumu dotazov.

5.3.1 Funkcia `check_flow`

Samotná implementácia algoritmov detekcie je riešená funkciou `check_flow`. Parametrami tejto funkcie sú štruktúry reprezentujúce agregovaný tok (položka z mapy načítaných tokov): `Flow_key_t` ako kľúč a `Flow_record_t` ako hodnota. Táto funkcia k svojej správnej funkcionalite potrebuje dve ďalšie globálne definované mapy tokov:

1. `susp_map1` určená pre podozrivé dotazy a
2. `susp_map2` určená pre podozrivé odpovede.

Tieto mapy sú určené pre uloženie podozrivých tokov, čiže útokov.

Uvedená funkcia pracuje ako jednorázová detekcia, čo znamená, že sa skontroluje *jeden* tok, ktorý bol načítaný v mape a už je určený pre detekciu. Avšak útok s rovnakými IP adresami už mohol byť zaznamenaný v predchádzajúcich detekciách a je o nich vedený záznam v mapách na to určených. Z tohto dôvodu na začiatku funkcie je potrebné zistiť, či nový tok nenaväzuje už na predchádzajúci a to z dôvodu sčítaní metrík útoku.

Ďalej je potrebné zistiť o aký tok sa jedná, respektíve pre akú detekciu je určený. Daný tok môže byť určený pre algoritmus podozrivých dotazov (reflection časť) alebo pre algoritmus podozrivých odpovedí (amplification časť). Následne v tejto funkcii sú implementované algoritmy detekcie, ktoré už boli uvedené v 4.2.4.

5.4 Spracovanie uživateľských volieb

Pri návrhu programu som uviedol, že je potrebné sa venovať aj uživateľským volbám. Tieto volby alebo nastavenia som sa rozhodol implementovať ako argumenty (parametry) daného detektora. Pri imlementácii som sa rozhodol použiť následujúce parametry:

- `-s` parameter znamenajúci počiatočný súbor,
- `-e` parameter znamenajúci koncový súbor s NetFlow dátmi,
- `-t` je čas v minutách reprezentujúci potrebný čas pre jednorázovú detekciu, hodnota musí byť väčšia ako 5 minút,
- `-d` volba pre zobrazenie detailov o útoku,
- `-w` argument, ktorý značí výstup do HTML súbora a
- `-h` zobrazenie nápovedy.

Z môjho pohľadu sú parametry určené intuitívne a sú skratkami anglických slov, ktoré daný parameter vystihuje (napríklad `-w` je odpovodený od slova web). Zisťovanie parametrov sa odohráva vo funkcii `parse_arguments` s využitím funckie `getopt`[2] z knižnice GNU[3] jazyka C. Následne je zistenými informáciami naplnená štruktúra na to určená (kód 5.1).

Taktiež v návrhu programu sa počítalo s podporou, lepšie povedané s potrebou použiť NetFlow súbory, ktoré majú dĺžku najviac 5 minút. Takéto súbory taktiež na seba musia naväzovať. Z týchto dôvodov je možné zadať počiatočný a koncový súbor. Pre samotnú implementáciu tejto časti je nutné poznať názvy súborov medzi začiatočným a koncovým NetFlow súborom. Túto situáciu som vyriešil nutnosťou, aby súbory boli pomenované ako *časové pečiatky*. V praxi to znamená, že ak je k dispozícii päť minútový súbor z 20. januára 2014 o 6 hodine, jeho názov by bol `201401200600` a naväzujúci súbor by mal názov `201401200605`. Samotná implementácia otvárania týchto súborov sa odohráva v hlavnej funkcii `main`. Ide o cyklus od počiatočného súboru po koncový a s inkrementáciou nazvu súboru o 5. V každom priechode cyklom sa naplní mapa tokov (funkcia `fill_flowmap` spomenutá v predchádzajúcich podkapitolách). Ďalej sa skontroluje táto mapa, či obsahuje toky, ktoré majú potrebnú dĺžku pre detekčný algoritmus, a následne sa pošlú do funckie pre detekciu `check_flow` a sú vymazané z mapy. Tie, ktoré túto vlastnosť nemajú zostávajú v mape pre následnú agregáciou v ďalších priechodoch cyklom.

5.5 Výstupné mechanizmy

Je potrebné implemtovať aj navrhnuté mechanizmy pre výstupy detektora. V návrhu som uviedol potrebu použiť dva typy výstupov: *formátovaný text na štandartný výstup* a *HTML súbor*.

V implementácii tieto dva výstupy zabezpečujú následujúce funkcie:

- `print_result` a
- `print_html`.

Prvá z uvedených funkcií zabezpečuje klasický textový formátovaný výstup. Aby bolo zobrazenie prehľadné rozhodol som sa implementovať mechanizmus, ktorý zabezpečí zobrazenie IP adresy, ktorá posiela, respektíve príma podozrivé toky. Implementáciu tohto mechanizmu som urobil jednoduchou agregáciou tokov podľa zdrojovej, respektíve cieľovej adresy. Tieto podozrivé toky sa nachádzajú po detekcii v globálnych mapách `susp_map1` a `susp_map2`.

Samotný formátovaný výpis jednotlivých tokov do ľudskej podoby implementuje funckia `print_record`. Táto funckia musí riešiť problém bajtových poradí (big endian/little endian) medzi sieťovým poradím bajtov a poradím bajtov na hostujúcej stanici. Tento problém sa týka IP adries a vyriešil som ho pomocou funckie `htonl`[4] z knižnice `arpa/inet.h`. Avšak tieto IP adresy su stále v binárnej podobe je treba ich previesť do čitateľnejšej podoby – textovej. Pre tento prípad slúži funckia `inet_ntop`[5] taktiež z knižnice uvedenej vyššie.

Avšak je nutné vziať do úvahy možnosť používateľa vynútiť si zobrazenie detailov[6]. Zobrazenie detailov znamená, že u každého v predchádzajúcom odstavci definovaného agre-

[2] `http://man7.org/linux/man-pages/man3/getopt.3.html`
[3] `http://www.gnu.org/`
[4] `man htonl`
[5] `man inet_ntop`
[6] parameter `-w` v podkalitole 5.4

govaného toku sa vypíšu jednotlivé toky, z ktorých bol ten pôvodný agregovaný. Príklad výstupu typom formátovano textu s detailom je zobrazený na obrázku 5.1.

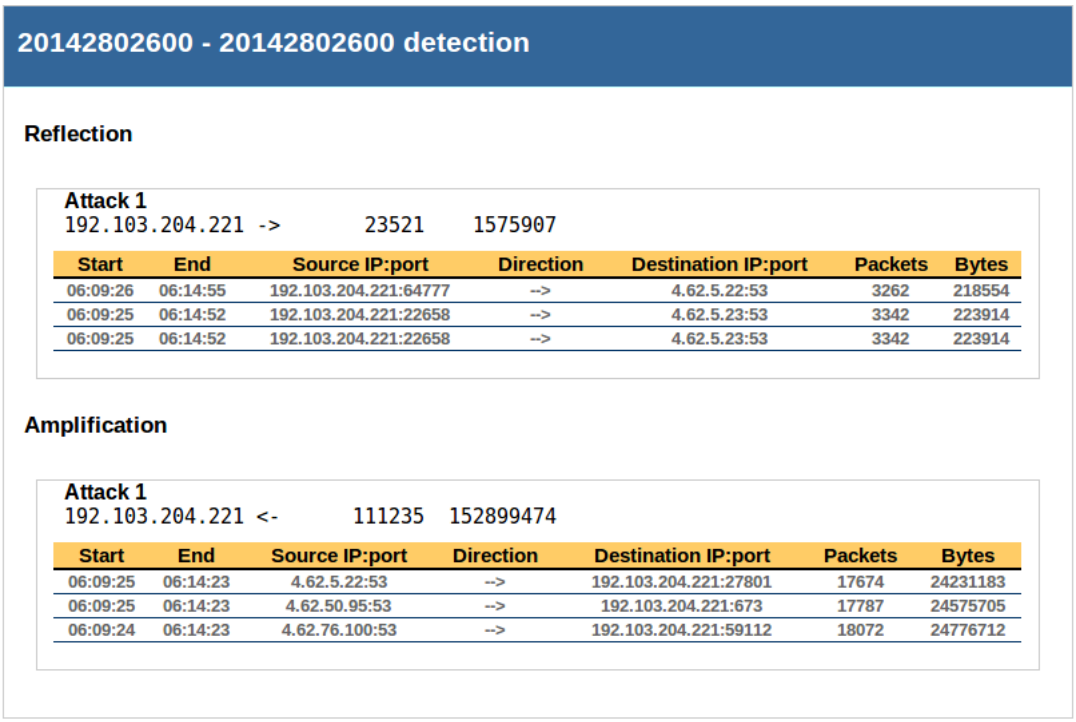
Druhá z funkcií je určená pre výstup do HTML súboru. Výhody tohto výstupu boli uvedené v návrhu. U implementácie tohto prístupu som vytvoril HTML šablonu, ktorú som naštýloval pomocou kaskádových štýlov, aby boli všetky dôležité prvky zobrazené prehľadne. Ďalej som sa pri implementácii uvažoval, či je potrebné aby uživateľ mal možnosť špecifikovať výsledný súbor HTML. Nakoniec som sa rozhodol pre programom generovaný nazov súboru podľa následujúceho predpisu:

`počiatočný_NetFlow_súbor-koncový_NetFlow_súbor.html`

```
AMPLIFICATION
 192.103.204.221 <-      139330  191459575
******************DETAIL**************************
[2014-02-13 05:59:19] - [2014-02-13 06:04:22]        4.62.5.22:53    ->  192.103.204.221:10606      18865   25864044
[2014-02-13 05:59:19] - [2014-02-13 06:04:22]        4.62.5.23:53    ->  192.103.204.221:16109      18732   25681572
[2014-02-13 05:59:19] - [2014-02-13 06:04:22]       4.62.50.95:53    ->  192.103.204.221:22583      18531   25603665
[2014-02-13 05:59:19] - [2014-02-13 06:04:22]      4.62.76.100:53    ->  192.103.204.221:42827      18981   26022951
[2014-02-13 05:59:19] - [2014-02-13 06:04:22]     4.62.107.194:53    ->  192.103.204.221:12810       4119    5647149
[2014-02-13 05:59:19] - [2014-02-13 06:04:21]      97.61.64.28:53    ->  192.103.204.221:47954      18237   25197455
[2014-02-13 05:59:19] - [2014-02-13 06:04:21]   106.191.17.250:53    ->  192.103.204.221:42690       1353    1869395
[2014-02-13 05:59:19] - [2014-02-13 06:04:21]   106.191.64.183:53    ->  192.103.204.221:38880      18846   25837866
[2014-02-13 05:59:19] - [2014-02-13 06:04:21]   106.191.189.98:53    ->  192.103.204.221:56972       2943    4066245
[2014-02-13 05:59:19] - [2014-02-13 06:04:21]  106.191.248.149:53    ->  192.103.204.221:22597      18723   25669233
------------------------------------------------------------------
```

Obrázok 5.1: Textový výpis aj s detailom

Na príklad, ak uživateľom definovaný počiatočný súbor nesie názov `2014022000` a koncový uživateľom definovaný súbor má názov `2014022020` výstupný HTML súbor bude mať názov `2014022000-2014022020.html`. Príklad výstupu do HTML súbora je na obrázku 5.2.



Obrázok 5.2: Ukázka HTML výstupu detektora

Kapitola 6

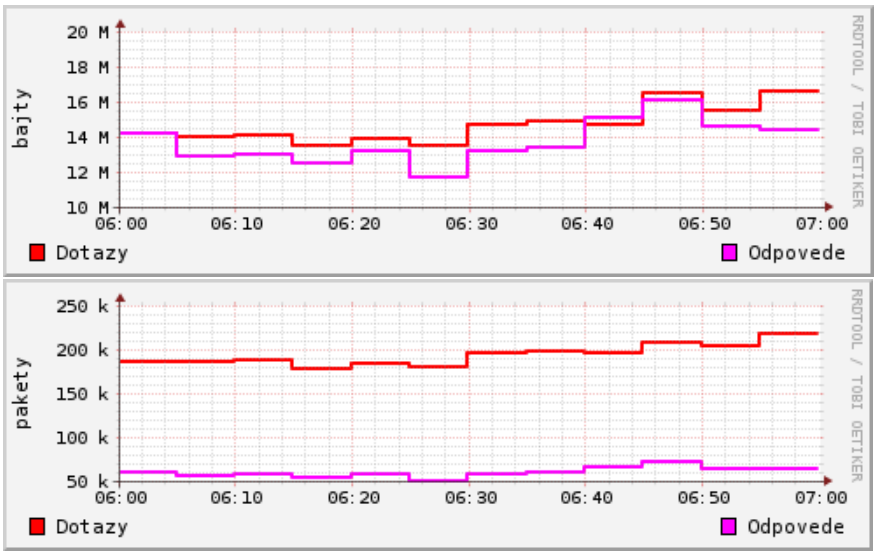
Analýza výsledkov detekovania

V tejto kapitole sa budem zaoberať rozborom výsledkov (testovaním) implementovaného detektora útoku DNS amplification. Tento rozbor bude založený na troch reálnych jednohodinových NetFlow dátach[1]. A to konkrétne:

1. prevažne čisté dáta,
2. dáta s jedným prevažujúcim útokom a
3. dáta s viacerími útokmi.

6.1 Dáta bez útoku

Dáta, na ktorých je táto podkapitola založená pochádzajú z 25. februára 2014 od 6:00 do 7:00 a *nemali by* obsahovať žiadne útoky DNS amplification alebo by mali obsahovať slabý útok.



Obrázok 6.1: Graf počtu bajtov a paketov súvisiace s DNS v závislosti na čase

[1]Všetky uvedené IP adresy v tejto práci boli anonymizované.

Na oboch grafoch (obrázok 6.1) je zobrazená závisloť počtu paketov a bajtov v závislosti na možnosti, či sa jedná o *dotaz* alebo *odpoveď*.

Výsledky detekovania ukázali, že daný tok obsahoval štyri podozrivé IP adresy, ktoré generovali nezvyklé dotazy na DNS (reflection časť útoku). Avšak, DNS odpovede na tieto dotazy neboli zachytené, a to mohlo nastať z dvoch dôvodov:

1. nejavili známky podozrivej odpovede pre algoritmus alebo
2. nemohli byť zachytené, pretože neprekročili hranice pozorovanej siete (viac podkapitola 3.3).

Po manuálnom preskúmaní týchto tokov sa ukázalo, že sa jedná o druhý z uvedených prípadov. Je treba poznamenať, že tieto podozrivé toky boli veľmi malé.

Čo sa týka druhej časti detekcie – podozrivých odpovedí (amplification časti útoku), bol detekovaný jeden podozrivý tok a to z adresy `97.56.26.150` na adresu `167.99.248.128`. Tento podozrivý tok trval 27 sekúnd a pozostával z 122 paketov celkovej veľkosti 129 405 bajtov. Tento útok je taktiež veľmi malý. Taktiež v tomto toku je vidieť, že nebol zaznamenaný dotaz, ktorý patrí k tejto odpovedi. Existujú zasa dve možnosti, ktoré sú uvedené vyššie. No, v tomto prípade sa jedná o druhú možnosť. Po preskúmaní bolo zistené, že daný tok mal smerodajnú ochyľku väčšiu ako 1, a tým pádom nebol daný tok detekovaný.

Všetky detektorom určené podozrivé IP adresy je možné nájsť v prílohe B v tabuľke B.1.

6.2 Dáta obsahujúce jeden prevyšujúci útok

Pri testovaní tejto situácie mi poslúžili dáta z 13. Februára 2014 od 6:00 do 7:00. Tieto dáta majú tú vlastnosť, že obsahujú viacero útokov, avšak iba jeden je prevyšujúci a vyčnieva nad ostatnými. Týmto myslím skutočnosť, že je oveľa silnejší ako iné podozrivé toky v týchto dátach.

Už z pohľadu na následujúci graf (obrázok 6.2) je viditeľný rozdiel od predchádzajúceho prípadu (graf na obrázku 6.1). Výrazný rozdiel je hlavne v počte bajtov, ktorý je od predchádzajúceho príkladu mnohonásobne väčší.

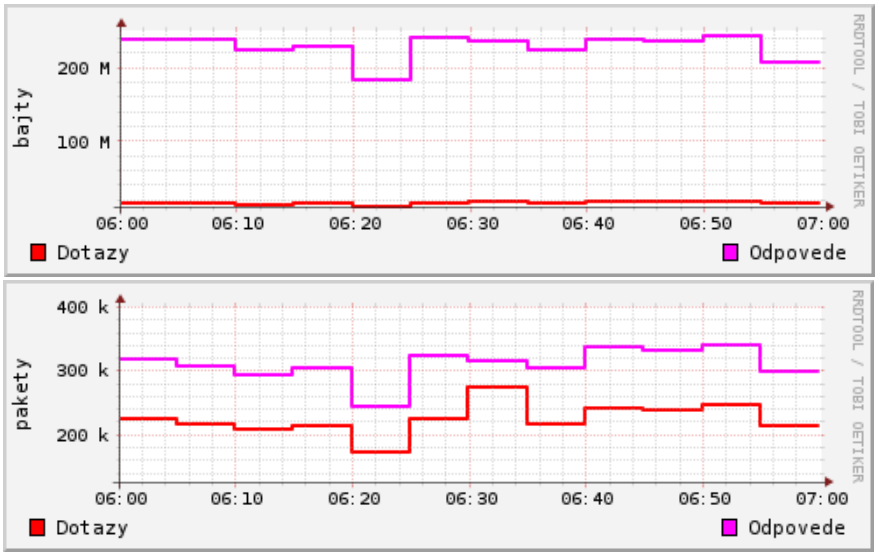
V týchto dátach sa vyskytlo päť IP adries, ktoré generovali neobvyklé DNS dotazy (reflection časť útoku). Z týchto piatich bola navýraznejšia práve `192.103.204.221`, z ktorej bolo poslaných približne 370 000 paketov v celkovej veľkosťi 25 miliónov bajtov, ako podozrivé DNS dotazy. Tieto dotazy smerovali na 13 rôznych DNS serverov.

Čo sa týka podozrivých DNS odpovedí (amplification časti útoku) v daných NetFlow dátach bolo detekovaných osem IP adries, ktoré prímali podozrivé odpovede. Z týchto ôsmich je jedna IP adresa totžná s hore uvedenou, čo znamená, že bol detekovaný útok ako celok (amplification aj reflection časť útoku). Hore uvedená adresa, `192.103.204.221`, prijala približne 1,6 milióna paketov o celkovej veľkosti približne 2 066 MB.

Hoci na uvedená adresa odoslala dotazy na 13 DNS serverov, detekovaných odpovedí ako podozrivých bolo len desať. Je to z dôvodu, že odpovede od troch DNS serverov mali menšiu priemernú veľkosť ako 900 bajtov, čo znamená, že boli veľmi malé. Manuálna kontrola to taktiež potvrdila.

V tejto situácii, keď sú zachytené aj dotazy aj odpovede je možné vypočítať približný celkový *zosiľnujúci faktor (amplification factor)*:

$$ZF = \frac{veľkosť(odpovede)}{veľkosť(dotazu)} = \frac{2\,191\,631\,149}{24\,731\,978} \doteq 88{,}6$$



Obrázok 6.2: Graf počtu bajtov a paketov súvisiace s DNS v závislosti na čase

Ku zbytku IP adries, ktoré boli detekované, ako prímajúce podozrivé DNS odpovede, neboli zachytené DNS dotazy. Zaujímavosťou je skutočnosť, že na tieto adresy boli odpovede poslané z jedného DNS servera s IP adresou `4.62.5.22`. Všetky detektorom určené podozrivé IP adresy je možné nájsť v prílohe B v tabuľke B.2.

6.3 Dáta s viacerými útokmi

Dáta u tohto druhu testu pochádzajú z 8. októbra 2013, trvajúce jednu hodinu – od 6:00 do 7:00. Tieto dáta obsahovali viacero podozrivých tokov, či už väčších alebo menších. Z pohľadu na následujúci graf (obrázok 6.3) je možno spozorovať výrazný rozdiel od predchádzajúcich prípadov (obrázok 6.1 a 6.2).

Obrázok 6.3: Graf počtu bajtov a paketov súvisiace s DNS v závislosti na čase

Po detekcii sa ukázalo, že dáta obsahovali 39 IP adries, ktoré generovali neobvyklé, či podozrivé DNS dotazy. Z týchto adries generovala najvačší počet dotazov práve adresa `7.47.136.230` a to 21 142 celkovej veľkosti 1,5 MB. Tieto dotazy boli smerovane na 51 rôznych DNS serverov. Aj odpovede od DNS serverov na uvedenú IP adresu boli dostupné a detekované ako podozrivé odpovede.

Druhou adresou, ktorá generovala najväčší počet podozrivých dotazov je `177.189.225.3`. Uvedená adresa generovala 19281 celkovej veľkosti približne 1,3 MB. Dotazy boli smerované na 27 rôznych DNS serverov. K týmto dotazom boli zachytené a detekované aj odpovede ako u predchádzajúceho prípadu.

Z celkových 39 IP adries podozrivých z posielania nezvyčajných dotazov na DNS u 30 z nich nie sú k dispozícii informácie o DNS odpovediach z dôvodu neprekročenia hranice pozorovanej siete (viac v podkapitole 3.3). Hoci o zbytku adries sú k dispozícii informácie o odpovediach, no ako podozrivé odpovede sa javilo len 6 odpovedí. Zbytok odpovedí na dotazy sa nejavilo ako podozrivé a to z dôvodu malej priemernej veľkosti odpovedí (do 900 bajtov).

Čo sa týka amplification časti detekcie – detekcia podozrivých DNS odpovedí bolo detekovaných 30 IP adries, ktoré takéto podozrivé odpovede prímali. Najväčšie (najsilnejšie) množstvo týchto opovedí prijala adresa `177.189.225.30`. Jednalo sa o približne 800 000 paketov celkovej veľkosti 1 000 MB. Druhou najviac prímajúcou adresou je `177.189.225.33`, ktorá prijala približne 350 000 paketov celkovej veľkosti 450 MB. K obom odpovediam boli k dispozícii aj DNS dotazy, avšak neboli detekované ako podozrivé, pretože nejavili klasické známky útoku časti reflection. Hlavným dôvodom prečo neboli detekované je skutočnosť, že jednotlivé dotazy sa od seba veľmi odlišovali, a tým pádom ich smerodajná odchýlka bola väčšia ako jedna. Je potrebné podotknúť, že daný detektor útok detekoval a skutočnosť, že detektor je navrhnutý tak aby zaznamenal obidve adresy, čo zabezpečí jednoduchosť dohľadania dotazov, ktoré generovali tieto nezvyklé odpovede.

Všetky detektorom určené podozrivé IP adresy je možné nájsť v prílohe B v tabuľke B.3.

Kapitola 7

Záver

Táto odborná práca sa zaoberá útokom DNS amplification a jeho detekciou. Súčasťou práce je teoretická časť, ktorá je zameraná na základný popis počítačových sietí, služby DNS a iných dôležitých častí pre daľší vývoj práce.

Pre tvorbu práce bolo nutné podrobnejšie naštudovať službu DNS, NetFlow a rôzne druhy útokov typu DoS a hlavne útok DNS amplification. Taktiež bolo potrebné zistiť informácie o možnosti detekovania uvedeného útoku. Ďalej sa bolo treba venovať návrhu a implementácii detektora. Pre implementáciu bolo potrebné vybrať a naštudovať nástroj pre prácu s NetFlow dátami v jazyku C/C++.

Výsledkom práce je funkčný detektor, ktorý je schopný z NetFlow dát detekovať DNS amplification útoky a snaží sa čo najviac vyhýbať false-positive detekciám. Použiteľnosť výsledného detektora je možná prevažne v obore správy sietí, kde by napomáhal sieťovým administrátorom v analyzovaní spravovanej siete a možnej detekcii a prevencii tohto typu útoku.

Priestor pre ďalší vývoj, či vylepšenie detektora samozrejme existuje. Budúci vývoj by mohol súvisieť s rozšírením, ktoré by detekovalo aj iné útoky ako DNS amplification, čím by sa stal nástroj všeobecnejším. Testy preukázali, že detektor nemal problém s detekciou útokov, ktoré mali stredne veľký až veľký zosiľnujúci faktor. Problém s detekciou mal práve u podozrivých tokov, ktoré mali priemernú veľkosť odpovedí menšiu ako 900 bajtov, čo je dané použitím priemernej veľkosti odpovedí ako určujúci faktor útoku. Tento problém by sa dal pravdepodobne vyriešiť kontrolou pomocou štandardnej odchýlky, v prípade zlyhania detekcie pomocou priemernej veľkosti, nakoľko aj veľkosti podozrivých DNS odpovedí sú si podobné. Otázne je, či je možné identifikovať tok, ktorého zosiľnujúci faktor je veľmi malý, za útok DNS amplification.

Čo sa týka reflection časti detekcie, detektor nemal takmer žiadny problém s detekciou typicky podozrivých tokov. Hoci, problém vznikal v prípade, že jednotlivé dotazy mali veľkú odlišnosť – smerodajná odchýlka bola väčšia ako 1. Tento problém je ľahko riešiteľný zvýšením hranice smerodajnej odchýlky, čo na druhú stranu môže spôsobiť veľa false-positive detekcií, ktorým som chcel predchádzať.

Literatúra

[1] HTML 4.01 Specification [online]. `http://www.w3.org/TR/html4/`, 18. December 1997 [cit. Apríl 2014].

[2] Směrodatná odchylka [online]. `http://cs.wikipedia.org/wiki/Směrodatná_odchylka#cite_note-odvozeni-1`, 25. Január 2013 [cit. Apríl 2014].

[3] EWMA Control Charts [online]. `http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm`, [cit. Apríl 2014].

[4] Albitz, P.; Liu, C.: *DNS and BIND*. O'Reilly Media, 2001, ISBN 0-596-00158-4.

[5] Claise, B.: *Cisco Systems NetFlow Services Export Version 9*. RFC 3954, Október 2004.

[6] Ferguson, P.: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. BCP 38, Máj 2000.

[7] Huistra, D.: *Detecting Reflection Attacks in DNS Flows*. University of Twente, 2013.

[8] Kambourakis, G.; Moschos, T.; Geneiatakis, D.; aj.: *Detecting DNS amplification attacks*. University of Aegean, Február 2013.

[9] Kováčik, M.: *Detekce síťových anomálií a bezpečnostních incidentů s využitím DNS dat*. VUT Brno, 2014.

[10] Mockapetris, P.: *Domain names - concepts and facilities*. RFC 1034, November 1987.

[11] Mockapetris, P.: *Domain names - Implementation and Specification*. RFC 1034, November 1987.

[12] Rozenkrans, T.; Koning, J. D.: *Defending against DNS reflection amplification attacks*. University of Amsterdam, Február 2013.

[13] Shinder, D. L.: *Počítačové sítě*. Softpress, 2003, ISBN 80-86497-55-0.

[14] Stein, L. D.; Stewart, J. N.: The World Wide Web Security FAQ. `http://www.w3.org/Security/Faq/wwwsf6.html`.

[15] Vaughn, R.; Evron, G.: *DNS Amlification attacks*. 17. Marec 2006.

Príloha A

Obsah DVD

Priložené DVD obsahuje zdrojové kódy detekčného nástroja, NetFlow dáta na testovanie, zdrojový kód tejto práce a návod na spustenie detekčného nástroja.

Príloha B

Tabuľky podozrivých IP adries určených detektorom

IP posielajúce podozrivé dotazy			IP obdržujúce podozrivé odpovede		
IP adresa	**pakety**	**bajty**	**IP adresa**	**pakety**	**bajty**
4.62.73.93	156	9 048	167.99.248.128	122	129 405
4.62.123.207	102	7 038			
4.62.190.131	181	10 136			
107.249.67.165	132	9 108			

Tabuľka B.1: Podozrivé IP adresy, ktoré sú výsledkom detekcie z 25. februára 2014 od 6:00 do 7:00

IP posielajúce podozrivé dotazy			IP obdržujúce podozrivé odpovede		
IP adresa	**pakety**	**bajty**	**IP adresa**	**pakety**	**bajty**
6.106.200.186	128	9 216	0.1.30.34	1 197	1 261 638
92.87.22.227	520	43 680	21.67.67.136	4 173	4 398 342
106.36.255.86	112	9 408	24.50.25.166	777	818 958
179.215.104.38	400	33 600	81.13.52.228	6 844	7 213 576
192.103.204.221	369 134	24 731 978	134.66.190.197	2 898	3 054 492
			187.185.30.171	199	209 746
			188.105.226.186	650	685 100
			192.103.204.221	1 576 608	2 166 635 153

Tabuľka B.2: Podozrivé IP adresy, ktoré sú výsledkom detekcie z 13. februára 2014 od 6:00 do 7:00

IP posielajúce podozrivé dotazy			IP obdržujúce podozrivé odpovede		
IP adresa	**pakety**	**bajty**	**IP adresa**	**pakety**	**bajty**
4.62.107.5	595	36 295	1.175.191.5	792	992 434
4.62.229.220	2 366	132 496	4.21.216.147	122	142 100
7.47.136.230	21 142	1 501 082	4.62.103.88	1 053	1 432 212
8.61.90.235	1 248	88 608	4.62.107.218	3 874	5 280 658
10.32.111.172	1 311	95 703	7.47.136.230	68 007	91 611 436
24.229.233.174	1 883	137 459	7.239.184.85	51 211	69 144 942
31.15.65.145	4 403	321 419	7.252.231.42	15 993	21 567 105
72.44.80.241	524	37 204	21.139.168.129	765	1 029 857
78.180.63.58	467	32 223	25.135.191.108	8 202	1 1082 086
78.182.216.118	6 367	452 057	78.198.35.104	169 777	228 873 597
98.215.66.156	4 087	298 351	78.248.190.179	150 810	203 162 542
106.191.64.157	1 643	134 726	92.32.40.99	948	1 151 664
106.191.87.85	1 666	136 612	93.126.119.218	591	799 079
106.191.87.86	1 607	131 774	106.191.64.157	250	340 082
109.217.69.131	1 321	96 433	107.39.19.6	124 026	167 273 243
130.84.19.167	127	9 017	126.255.210.162	11 269	14 159 427
130.178.52.154	3 456	245 376	162.255.138.3	35 930	45 100 544
131.95.167.251	5 610	398 310	166.138.189.182	1 596	2 154 216
140.63.179.78	369	25 461	174.19.20.95	11 658	14 661 917
143.115.219.59	2 467	180 091	174.224.5.224	16 182	21 818 150
154.42.225.54	3 269	238 637	176.76.33.162	13 842	18 623 886
154.46.244.113	1 248	91 104	177.189.225.3	240 944	324 715 002
159.92.214.104	1 035	75 555	177.189.225.29	5 283	7 122 303
161.189.103.161	135	9 585	177.189.225.30	764 384	1 031 509 782
168.255.184.14	1 100	75 900	177.189.225.33	337 197	454 697 235
176.76.33.162	1 986	141 006	183.146.99.117	193 225	260 515 622
177.189.225.3	19 281	1 368 951	189.59.182.188	1 209	1 630 637
180.180.47.137	191	13 561	205.116.241.208	945	1 278 277
183.36.221.223	1 022	70 518	222.137.140.98	10 884	14 653 251
183.109.249.204	1 661	117 931	243.211.79.49	6 636	8 958 048
190.214.217.173	276	19 596			
210.96.153.129	3 120	221 520			
220.229.190.96	1 644	120 012			
220.241.248.1	1 098	80 154			
222.137.140.98	2 262	160 602			
231.166.232.25	311	21 459			
231.254.200.195	1 995	141 645			
232.224.90.199	8 889	631 119			
243.211.79.49	2 925	207 675			

Tabuľka B.3: Podozrivé IP adresy, ktoré sú výsledkom detekcie z 8. októbra 2013 od 6:00 do 7:00