DoS spoofing-attacker

Hur stor är möjligheten till att implementera spoofing-filter i routrar över Internet?

Marcus Löf **malof@kth.se**

CDATE12 **KTH**

DD1395 **VT 2015**

Sammanfattning.

Idag utgör *Denial Of Service*-attacker ett stort hot mot aktörer på Internet. Uppsatsen introducerar först vad DoS-attacker är samt dess konsekvenser, därefter förklaras den mer specifika DoS-attackskategorin, *spoofing.* Olika typer av spoofing-attacker behandlas följt av skydd mot dessa, där fokuset hamnar på spoofing-filter. Spoofing-filters funktionalitet samt nackdelar och olika implementationer utreds. Därefter diskuteras möjligheterna att implementera dessa filter i routrar över hela Internet, och en slutsats dras utifrån materialet som behandlats. Uppsatsen är skriven för målgruppen civilingenjörsstudenter.

Innehållsförteckning.

Introduktion.

Teknologin utvecklas konstant och når hela tiden nya höjder, och en stor del av samhället bygger på att diverse tekniska system ska fungera. Då dessa systems funktioner kan vara väldigt vitala, innebär det en sårbarhet för attacker som förhindrar de från att fungera. Det talas ofta om attacktypen *Denial Of Service(DoS)*-attack. Attacken går ut på att försöka få ett datasystem att inte kunna utföra sina tänkta arbetsuppgifter, det kan dels handla om att få det att bli långsamt eller att helt och hållet att få det att sluta fungera.[1] Attackerna kan vara mot en CPU(Central Processing Units), mot minnen, disk utrymmen eller bandbredd. Ofta faller DoS-attackerna i den senaste kategorin och är nätbaserade. Detta innebär att det är nätverksfunktioner som får ta emot så mycket trafik att dessa inte klarar av att hantera mer, som då kan få som konsekvens att de attackerade nätverksfunktionerna stängs ner.[2] Detta kan få digra påföljder, nätverkssystemet som attackeras kan exempelvis vara en server där en webbsida körs, vilket då innebär att det inte längre går att ansluta till den hemsidan. Om hemsidan i fråga är en bank, kan inte längre kunder använda sin internetbank.

DoS-attacker kan alltså på olika sätt förhindra normal användning av diverse tekniska system, men det är trots detta svårt att skydda sig mot, då dessa attacker kan se ut på många olika sätt. En av attacktyperna som används är spoofing. Spoofing går ut på att en attackerare skickar internetpaket med en förfalskad IP-adress som källadress. Detta innebär att mottagaren sedan tror att paketet kommer från den IP-adressen som står som avsändare, när det inte är sanningen. Med det följer att anfallaren kan se till att sitt mål får ta emot väldigt många paket och anslutningar från olika IP-adresser i syfte att överbelasta. Detta samtidigt som den skyldige med stor sannolikhet inte blir upptäckt. Det går givetvis att lista ut vem som attackerade genom att titta på flödena av paketen, men det är en betydligt mer tidskrävande och svårare uppgift än att bara titta på källadressen för paketen.[2] För att förhindra någon från att skicka paket med förfalskad IP finns så kallade spoofing-filter, vilka filtrerar baserat på avsändaradresser och kan därför förkasta paket med ogiltiga avsändaradresser.[3]

Syfte.

Med de växande säkerhetskraven som ställs på samhället idag, är DoS spoofing-attacker något som kan hota mycket av funktionaliteten på Internet. Uppsatsens syfte är att behandla DoS spoofing-attacker och de skydd som finns, där tyngdpunkten hamnar på spoofing-filter. Det slutgiltiga målet är att, med hjälp av materialet som behandlats, sedan besvara frågeställningen *Hur stor är möjligheten till att implementera spoofing-filter i routrar över Internet?*

Metod.

Författaren har med en grundkunskap om DoS, fördjupat sig i DoS spoofing-attacker genom att läsa djupare i diverse kurslitteratur samt kombinera det med att hitta utomstående information i vetenskapliga och tekniska skrifter(ex. RFCs) samt skrifter som går att läsa hos kända aktörer på marknaden. Även undersökningsresultat används från *MIT spoofer project* och *Cmand spoofer project*. De slutsatser som dras baseras på det underliggande materialet, även om författaren själv står för analys och tolkning av det.

Bakgrund och analys.

När en grundförståelse för DoS har etablerats, finns möjlighet att gräva djupare i hur DoS spoofing-attacker fungerar i olika skepnader. Detta avsnitt i uppsatsen behandlar först olika typer av DoS spoofing-attacker, följt av hur skydd mot dessa kan se ut där tyngdpunkten hamnar på spoofing-filter.

Spoofing DoS-attacker.

Det är problematiskt att IP-spoofing fungerar, flertalet olika DoS-attacker kan utföras med hjälp av detta. Om exempelvis den förfalskade IP-adressen som står som avsändare på ett internetpaket faktiskt tillhör en verklig IP, finns en risk att mer än en maskin drabbas av en DoS spoofing-attack.[4] Attackeraren skickar då internetpaket till ett av sina mål med en verklig förfalskad avsändaradress, paketet kan ofta vara en *ICMP-request* (ex. ping) vilket i sin tur leder till att målet kommer att svara på frågan med en *ICMP-response*. Responsen skickas då tillbaka till den angivna källadressen. Adressen tillhör då en verklig host, vilken inte förväntar sig att få detta paket och kan därför skicka tillbaka ett felmeddelande. Detta innebär alltså att det direkta målet har fått ta emot två paket, dels en förfrågan och dels ett felmeddelande, och har dessutom fått bearbeta dessa. Det indirekta målet har i sin tur fått ta emot ett paket och bearbeta det. Sker detta i större skala belastas både det direkta och indirekta målet med betydligt mer trafik, vilket kan bli så mycket att dessa datasystem inte fungerar som de ska, DoS. Även i de fall avsändaradressen inte går att nå med ICMP-response, får det direkta målet med stor sannolikhet ändå ett ytterligare paket, felmeddelandet *ICMP destination unreachable*.[5]

Det går alltså inte bara att direkt attackera ett mål med hjälp av spoofing, även indirekta attacker är möjliga. Attacker som ser ut på det viset kallas för *reflection-attacker*. Dessa går ut på att skicka trafik till diverse datasystem med spoofad avsändaradress, vilket kommer att leda till att dessa datasystem försöker svara och skickar därför respons på trafiken till den givna avsändaradressen.[6] Den givna avsändaradressen i detta fallet är alltså målet för attacken, även om det är indirekt. Attackeraren kan, förutom att skicka trafik till massvis med hosts för svar, också se till att svaren som ges innehåller så mycket data som möjligt för att göra attacken effektiv. Anfallaren kan därför tänkas använda sig av DNS-frågor, då svaren för dessa innehåller mycket data. Just att förvärra volymen och tyngden av svaren i en reflection-attack kallas för amplification.[7]

Reflection-attacker kan också handla om att skicka SYN-paket, vilket är det första paketet i en initiering till en anslutning via TCP. Dessa skickas återigen till diverse datasystem följt av att systemen enligt protokoll svarar med SYN-ACK, vilket är det andra paketet för att sätta upp en anslutning. Paketen som fås som svar är sedan tänkta att överbelasta målet.[8] Att använda sig av SYN-paket vid DoS med spoofing är inte unikt i bara reflection-attacker, SYN-flooding med spoofing är också möjligt. Attacken innebär till skillnad från reflection att målet direkt attackeras. Detta sker genom att anfallaren genererar SYN-paket med diverse förfalskade IP-adresser och skickar dessa paket till sitt mål. Datasystemet som anfalls, får då ta emot massor med anslutningar, där tanken är att fylla upp målets kö med anslutningar. När denna är full kan inga nya förbindelser till datasystemet göras, vilket gör att attacken har åstadkommit DoS.[9]

Skydd mot DoS spoofing-attacker.

Det finns flertalet skydd mot DoS spoofing-attacker, även om attacktypen är svår att skydda sig mot. Den senaste attacken som nämndes ovan, SYN-flooding med spoofing, är dock något som kan bemötas. Attacken går som bekant ut på att med många anslutningar försöka överbelasta målets anslutningskö, så att inga nya förbindelser kan göras. Ett framgångsrikt skydd mot detta är att, om kön är full, slumpmässigt stänga anslutningar i listan. Detta kan med otur innebära att riktiga anslutningar ersätts av spoofade, men det går att probabilistiskt visa att metoden är lyckosam.[9]

Det finns även andra metoder att bemöta SYN-spoofing på. Anslutningar har en keepalive-timer, vilket innebär att för att behålla anslutningen, behöver de anslutna datasystemen få någon signal av varandra att kommunikation faktiskt fortfarande pågår innan dess att timern räknar ut. Om ingen trafik utbyts räknar denna timer ner till noll, och anslutningen stängs ner. Varje gång data fås, återställd timern och börjar återigen räkna ner.[10] Det en anfallare då kan göra är att se till att skicka ett hjärtslag, en signal som bara ser till att väcka anslutningen, varje gång timern håller på att räkna slut, och på så vis se till att anslutningen inte stängs. Detta fungerar givetvis om keepalive-timern har ett kostant värde, och anfallaren har vetskap om detta. Motmedlet är då att ha ett mer slumpmässigt keepalive-värde. På så vis kan inte anfallaren veta hur denne ska hantera hjärtslagen för att hålla igång anslutningen, och spoofade anslutningar kan då stängas. Dock förhindrar ingen av de två metoderna som beskrivits ovan en SYN-spoofing attack, däremot försvåras attacken.[11]

Många anfallare som använder sig av spoofade IP-adresser, använder sig av adresser som ingen annan har överhuvudtaget. Dessa kan, av målsystemet, detekteras med hjälp av ett så kallat *Bogon address filter*. Filtret som ofta implementeras i den yttre delen av ett nätverk, har helt enkelt information om vilka adressblock som inte är allokerade. Detta innebär att filtret kan upptäcka om inkommande paket har en källadress som tillhör dessa ickeallokerade adresser, och på så vis kan dessa paket direkt förkastas. På detta sätt kan många spoofing-attacker undvikas, enligt mätningar tillhör upp till 60% av de spoofade adresserna vid attacker bogon-kategorin.[12]

De skydden som nämnts ovan förhindrar givetvis till viss grad DoS spoofing-attacker. Författaren menar dock att det egentligen går att se på det som så att om paketen kommer fram till det tänkta målet, har attacken redan delvis lyckats. Även om datasystemet i sig har skydd, exempelvis filter eller försvårande timers, är det fortfarande data som måste bearbetas av datasystemet vilket är resurskrävande. Det enda sättet som faktiskt till fullo förhindrar en attack på ett målsystem, är att se till att paket med förfalskade adresser överhuvudtaget inte kan ta sig fram till målet, detta kan uppnås med spoofing-filter.

Ett spoofing-filter har som uppgift att titta på ett inkommande internetpakets avsändaradress, och bestämma sig för om det är giltigt eller bör förkastas. Filtret implementeras bäst i routrar över hela Internet, då en router har som uppgift att titta på inkommande paket följt av att se till att dessa vidarebefordras rätt. Detta gör att det är naturligt att paketens avsändaradress också inspekteras av routrarna. Förhoppningen med att ha sådana filter är att det så nära som möjligt till avsändaren ska kunna gå att verifiera om IP-adressen är korrekt, och om inte, förkasta paketet.[13] Det handlar om att routern i fråga kan förvänta sig avsändaradresser inom ett visst adresspann. Om adressen i fråga inte tillhör det spannet, inses det att det inkommande paketet inte har en korrekt avsändaradress, och tillhör därför sannolikt någon form av spoofing-attack. Desto närmare en router är en host, desto färre möjligheter att förfalska en avsändaradress finns det eftersom det förväntade adresspannet är

mindre.[14] En router lokaliserad vid KTH:s nätverk bör exempelvis förvänta sig utgående paket med en avsändaradress likt 130.xxx.xxx.xxx, om så inte är fallet bör misstankar väckas och paketet förkastas. När ett paket förkastas av en router med anledning av IP-spoofing, bör detta loggas för att på så vis kunna upptäcka avvikande beteende.[14] En router kan också innehålla ett Bogon address filter.[12]

Spoofing-filtret kan fungera på olika vis. En metod som används är att ha en lista med giltiga IP-adresser, och jämföra de inkommande paketens avsändaradress mot denna lista för att se om den är giltig. Nackdelen med detta tillvägagångssätt är att adresserna måste vitlistas manuellt, vilket gör att mänskliga misstag kan innebära att avsändaradresser som är tillåtna anses vara otillåtna och vice versa. Det finns då andra infallsvinklar. När ett meddelande fås tar routern avsändaradressen och undersöker på vilket fysiskt medium, som routern är inkopplad på, som ett paket skulle skickas om det var på väg till avsändaradressen. Om detta fysiska medium motsvarar det där paketet kom ifrån, accepteras paketet och kan då vidarebefordras, annars förkastas det. Detta är dock inte vattentätt, det finns fallgropar eftersom Internet inte är symmetriskt, paket som går mellan två noder på Internet går inte alltid samma väg åt båda hållen vilket gör den senare infallsvinkeln problematisk. Det finns då möjlighet till förbättring av tillvägagångssättet, där det handlar om att även alternativa vägar ses över, eller till och med om det över huvudtaget finns en väg.[13] Trots dessa förbättringar är det fortfarande svårt att säkert kunna avgöra om ett paket ska få passera en router baserat på om det finns vägar att skicka tillbaka ett paket till avsändaren på samma fysiska enhet. Storleksordningen på Internet gör det problemematiskt och resurskrävande för routrarna att behöva göra dessa beslut. Det bästa och enklaste är att använda den förstnämnda metoden, att vitlista tillåtna IP-adresser, förutsatt att det manuella arbetet är korrekt utfört.[13]

Spoofing-filter har dock begränsningar. Filtret stoppar givetvis inte spoofade adresser som tillhör ett adresspann som tillåts av en router, vilket fortfarande kan möjliggöra spoofade DoS-attacker även om möjligheterna hos en sådan attack minskar.[14] Filtret är också problematiskt för *Mobile IP*, så som den strukturen är implementerad.[15] Mobile IP ska trots att en mobil nod förflyttar sig till andra nätverk, kunna behålla sin IP-adress fastän noden nu ansluts till en annan punkt än förut.[16] Detta innebär en problematik för filtret, då en IP-adress kan vandra mellan olika anslutningspunkter, där routrar med vitlistade adresser eventuellt förkastar paket från den mobila noden.[15] Dock finns en lösning på problemet som innefattar att tunnla trafik från och till den mobila noden, paketen inkapslas och skickas via en så kallad *home agent*[17]. Denna strategi är dock inte effektiv och skapar onödig trafik via Internet eftersom utgående trafik från en mobil nod först måste gå till sin home agent följt av sitt egentliga mål, istället för direkt till destinationen.[18]

Att implementera spoofing-filter i routrar över hela Internet är någonting som ligger en bit bort i tiden[19], menar vissa, medan andra menar på att filtret indirekt faktiskt finns implementerat i viss utsträckning, exempelvis i form av vanliga paketfilter i routrar. Enligt ett projekt vars syfte att undersöka framgången hos spoofade avsändaradresser, lyckades cirka 80% av alla spoofade avsändaradresser från adressrymden filtreras bort.[20] Vidare finns det en mätning *från MIT spoofer project* som menar på att någon form av spoofing-filter finns implementerat på 76,2% av alla autonomous system, AS.[21] Givetvis är målet att förebygga all spoofing, och det specifika spoofing-filtret i sig är ett tillägg som snabbt skulle kunna implementeras.[14] Det finns också redan routrar som har funktionaliteten, exempelvis av Cisco och Juniper.[12]

Det går givetvis att fråga sig varför vissa menar på att en bred implementation av spoofing-filter är något som ligger en bit bort i tiden när det enda som krävs, vilket anslogs ovan, är att implementera filter som vitlistar avsändaradresser. Att det dessutom redan finns routrar med spoofing-filterfunktioner gör det hela ännu märkligare, detta diskuteras i *Diskussion och slutsats*.

Källkritik.

Det finns anledning att kritisera valet att, till en liten del, även inkludera kurslitteratur som källor då dessa får anses vara grundläggande och ej fördjupande. Dock har dessa endast använts som ett skelett att bygga texten kring, och nyckelpoängerna vilar inte på information från dessa källor även om det bör nämnas att trovärdigheten hos dessa källor är hög. Mycket av nyckelinnehållet i uppsatsen bygger på diverse RFCs, vilka torde vara tillförlitliga samt ha en objektiv infallsvinkel. Vissa av dessa är relativt gamla, vilket är en kritik som går att rikta. Dock anser författaren att mycket av materialet är tidlöst. Även två undersökningar(*MIT spoofer project* och *Cmand spoofer project*), vars syfte att titta på hur mycket spoofad trafik som filtreras bort, har använts. Den förstnämnda får anses trovärdig medan den andra möjligtvis inte är lika tillförlitlig. Dock pekar båda dessa oberoende undersökningar på ungefär samma resultat vilket ökar trovärdigheten som helhet.

Diskussion och slutsats.

DoS spoofing-attacker är idag något som kan göra stor skada vid en lyckad attack, samtidigt som det är svårt att komma åt förövaren. Det finns, som nämnts ovan, flertalet skydd mot spoofing-attacker lokaliserade i ett måls datasystem som inte involverar spoofing-filter. Problemet med dessa är att oavsett hur bra paketfilter eller keepalive-timers systemet har, är det fortfarande resurskrävande att bearbeta inkommande anslutningar eller paket. Det är givetvis bra att ha dessa skydd, men det räcker inte för att till fullo vara skyddad mot spoofing formen av DoS-attacker. Det måste alltså handla om att se till att paketen från spoofade adresser aldrig når fram till sitt mål. Det uppenbara sättet att göra det på är att använda spoofing-filter. Filtrerna minskar betydligt en anfallares möjligheter att lyckas med attacker, men också sannolikheten för ett utförande av en sådan, då det är lättare att identifiera den skyldige. Filtrerna kan givetvis inte förhindra alla typer av DoS-attacker, anfallaren kan fortfarande använda sig av exempelvis botnät, men möjligheterna till en attack minskar. Det går inte att förneka att dessa filter är positiva i den bemärkelsen att dessa skyddar mot DoS spoofing-attacker, men det finns dock uppenbarliga problem med filtret. Exempelvis Mobile IP, som nämndes ovan, fungerar endast med hjälp av *tunneling* via en home agent, vilket är ineffektivt. Det handlar alltså om att väga funktionalitet mot säkerhet. Det finns en säkerhetsprincip, *principle of psychological acceptability*, som säger att en säkerhetsmekanism inte ska försvåra möjligheten att nå en resurs.[22] Denna bryter då spoofing-filtret mot, om det gör det svårare för mobila noder att nå resurser på Internet än om filtret inte skulle finnas. Författaren anser dock att spoofing-filter är en central försvarsmekanism som torde fungera för att till fullo kunna skydda sig mot spoofing-attacker, därför bör det anses okej med ett ineffektivt tillvägagångssätt för Mobile IP i utbyte mot fungerande spoofing-filter.

Enligt vissa är spoofing-filtret något som är långt ifrån implementerat, men samtidigt finns det trots allt statistik på att mycket spoofad trafik faktiskt filtreras bort, detta med hjälp av bland annat vanliga paketfilter. Att mycket sådan trafik filtreras är dock inte tillräckligt, det behövs att all trafik med spoofade avsändaradresser förkastas innan säkerhetsexperterna bör vara nöjda anser författaren. Med bakgrund av att spoofing-filterfunktioner i sig inte är särskilt svårimplementerade och att vissa

routrar idag faktiskt stödjer dessa filterfunktioner, finns uppenbarligen goda möjligheter att implementera filtret över hela Internet. Varför detta inte görs tros av författaren bland annat bero på att det är väldigt många routrar som behöver få den nya funktionaliteten installerad och konfigurerad med vitlistade adresser eller dylikt. Detta ligger det, som nämndes i *Bakgrund och analys*, mycket manuellt arbete bakom vilket möjligtvis inte anses vara värt mödan då det ändå inte stoppar en anfallare från att utföra DoS-attacker med andra metoder såsom botnät. En annan faktor som författaren tror spelar in är att som dessa filter är implementerade och fungerar, vilket beskrevs i *Bakgrund och analys*, skyddar de snarare andra än sig själv. Med detta menas att exempelvis ett spoofing-filter i utkanten av KTH:s nätverk lätt skulle kunna avgöra om ett utgående paket kommer från en korrekt adress eller inte medan det är svårare att avgöra om ett inkommande paket har en äkta avsändaradress. Det är alltså viktigt att funktionaliteten finns så nära källan som möjligt, eftersom det är där det bäst går att validera avsändaradressen. Detta gör att dessa filter alltså snarare skyddar andra än sig själv, vilket kan minska incitamenten för en implementation.

Så för att besvara frågeställningen *Hur stor är möjligheten till att implementera spoofing-filter i routrar över Internet?*, menar författaren att möjligheten till att implementera spoofing-filter i routrar över Internet är stor. Det är relativt enkelt, men ändock en tidskrävande process med den kvantiteten av routrar som finns. Funktionaliteten finns trots allt redan, även om bakomliggande orsaker såsom bristande incitament, gör att implementation av dessa filter inte sker i en stor utsträckning. Författarens profetia är att det inte kommer ske några kraftiga aktioner för att få dessa filter implementerade i framtiden heller, hade incitamenten varit tillräckliga hade spoofing-filter redan funnits i hög utsträckning då de inte är svårimplementerade. Däremot tror författaren att filtret över tid kommer att implementeras i takt med att säkerhetsrestriktioner blir allt viktigare. Det kan mycket väl handla om fler restriktiva paketfilter-routrar, vilka redan idag filtrerar bort mycket spoofad trafik. Där tror författaren att datorvärlden kommer se en långsam utbredning av en form av spoofing-filter.

Referenslista.

[1] Forouzan B. *TCP/IP Protocol Suite* , Fourth ed. New York. McGraw-Hill; 2010. P 819.

[2] Stallings W, Brown L. *Computer Security Principles and Practise* , Second ed. USA. Pearson Education; 2012. P 222,225.

[3] Ramachandran V, Choudhary M, Madhusudhana H. *Method and system for filtering spoofed packets in a network*. Novell, Inc; 2008. US7360245 B1(Patent).

[4] Furguson P, Senie D. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. The Internet Society; 2000. P 3. Tillgänglig på: http://tools.ietf.org/html/rfc2827.html (hämtad 25 januari 2015)

[5] Stallings W, Brown L. *Computer Security Principles and Practise* , Second ed. USA. Pearson Education; 2012. P 224-225.

[6] Verdejo Alvarez G. *Denial Of Service Attacks*. Barcelona. edge security; 2004.Tillgänglig på: http://www.fortguard.com/DDOS/Ataques%20DOS-DDOS%20FIST%202004.pdf (hämtad 25 januari 2015)

[7] Kührer M, Hupperich T, Rossow C, Holz T. *Exit from Hell? Reducing the Impact of Amplification DDoS Attacks.* San Diego: usenix ; 2014. P 111. https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-kuhrer.pdf (hämtad 26 januari 2015)

[8] Stallings W, Brown L. *Computer Security Principles and Practise* , Second ed. USA. Pearson Education; 2012. P 235-236.

[9] Dr. Lincoln P, Ricciulli L, Kakkar P, *TCP SYN Flooding Defense*.* USA: SRI International; 1999. P 1-2. Tillgänglig på: http://www.csl.sri.com/papers/cnds99/cnds99.pdf (hämtad 28 januari 2015)

[10] Forouzan B. *TCP/IP Protocol Suite* , Fourth ed. New York. McGraw-Hill; 2010. P 482.

[11] Stallings W, Brown L. *Computer Security Principles and Practise* , Second ed. USA. Pearson Education; 2012. P 242.

[12] Garcia F, Cerezo J. *RIPE Anti-Spoofing Task Force HOW-TO.* : RIPE ; 2008. Tillgänglig på : http://www.ripe.net/ripe/docs/ripe-431 (hämtad 29 januari 2015)

[13] Baker F, Savola P. *Ingress Filtering for Multihomed Networks*. : The Internet Society; 2004. Tillgänglig på: http://www.hjp.at/doc/rfc/rfc3704.html (hämtad 29 januari 2015)

[14] Furguson P, Senie D. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. The Internet Society; 2000. P 4-7. Tillgänglig på: http://tools.ietf.org/html/rfc2827.html (hämtad 25 januari 2015)

[15] Das D, Fitzgibbons P, Hash L. *SECURE ARCHITECTURE FOR EXTENSIBLE MOBILE INTERNET TRANSPORT SYSTEM*. State University of New York Institute of Technology at Utica-Rome. AFRL-IF-RS-TR-2004-139, 2004. P 26, 43

[16] Perkins C. *IP Mobility Support for IPv4, Revised.* : IETF Trust ; 2010. P 5. Tillgänglig på: http://tools.ietf.org/html/rfc5944 (hämtad 29 januari 2015)

[17] Montenegro G. *Reverse Tunneling for Mobile IP*. The Internet Society; 1998. Tillgänglig på: http://tools.ietf.org/html/rfc2344 (hämtad 1 februari 2015)

[18] Tsigas P, Senie D. *Mobile IP*. Chalmers; 2010. P 6. Tillgänglig på: http://www.cse.chalmers.se/~tsigas/Courses/DCDSeminar/Files/Mobile%20IP.pdf (hämtad 1 februari 2015)

[19] Buchegger S. *Denial of Service, Intrusion Detection*. KTH; 2014.Tillgänglig på: https://www.kth.se/social/files/547daabff276547c3f2fcc07/dasak14Lecture11dosids.pdf (hämtad 28 januari 2015)

[20] Beverly R. *Spoofer Project: State of IP Spoofing.* : CMAND ; 2014. Tillgänglig på: http://spoofer.cmand.org/summary.php (hämtad 29 januari 2015)

[21] Yao G, Bi J, Xiao P. VASE: Filtering IP spoofing traffic with agility. *Computer Networks* 2013; 57(1):243-257. Tillgänglig på: http://netarchlab.tsinghua.edu.cn/~junbi/ComputerNetworksVol57No1.pdf (hämtad 30 januari 2015)

[22] Buchegger S. *Introduction, Cryptography*. KTH; 2014.Tillgänglig på: https://www.kth.se/social/files/545ca25bf27654159b8a3f21/dasak14Lecture02introcrypto.pdf (hämtad 1 februari 2015)