

Covert Channels

Waarom technische filterhulpmiddelen niet werken

Systeem- en netwerkbeheerders zullen vaak geneigd zijn om het communicatie gedeelte van het beveiligingsbeleid van een organisatie af te dwingen door het gebruik van technische hulpmiddelen. Dit artikel maakt duidelijk dat dit in veel gevallen onvoldoende blijkt te zijn. Door het gebruik van zogenaamde covert channels glippen gebruikers dwars door alle filterapparatuur en –programmatuur heen. Het opstellen en handhaven van een goed gebalanceerd beveiligingsbeleid zal vaak betere resultaten geven. In de meeste gevallen blijft beveiliging dan ook gewoon mensenwerk.

*Auteur: **Hans (J.C.G) Van de Looy** is een van de oprichters en senior security consultant van Madison Gurkha BV. Voor reacties: hans@madison-gurkha.com.*

Inleiding

Natuurlijk heeft u als security officer, systeem- of netwerkbeheerder uw beleid prima op orde. Duidelijke richtlijnen zijn vastgelegd over wat wel en niet is toegestaan. Dat geldt ook voor de communicatie vanuit en naar het internet en andere netwerken waar uw IT-infrastructuur aan gekoppeld is. Maar zijn de genomen maatregelen voldoende? Indien u uitsluitend vertrouwt op uw firewalls en proxysystemen dan zou u wel eens bedr ogen uit kunnen komen. Ik hoop met dit artikel weer eens het belang van ‘awareness’ en ‘opvoeding van gebruikers’ aan te geven als het gaat om het kunnen handhaven van het vastgestelde beleid.

Technische filtermogelijkheden

Zoals in de inleiding al is aangegeven zullen veel systeem- en netwerkbeheerders het communicatiebeleid van een organisatie proberen af te dwingen door het gebruik van firewalls en proxyservers, om zodoende misbruik tegen te gaan. In dat soort omgevingen zal surfen vaak uitsluitend worden toegestaan via de eigen HTTP-proxy die veelal ook nog beperkingen zal opleggen welke URL's bezocht mogen worden. Helaas voor de meeste bedrijven wordt op deze proxies vaak black-listfiltering gebruikt, waardoor zeker niet al het ongewenste surfgedrag wordt tegengegaan. Ook voor FTP-communicatie wordt vaak gebruik gemaakt van een proxy en peer-to-peerprotocollen zullen zeker worden tegengegaan aangezien deze, volgens vele experts, een grote bron van malware zijn en slechts zelden een valide business-case voor het gebruik ervan kan worden aangehaald. Als laatste veel geïmplementeerde voorbeeld noem ik het verbod voor het personeel om bij de eigen provider de persoonlijke e-mail te lezen via het POP3 (of het IMAP) protocol. Dit laatste vooral om te voorkomen dat er virussen op het netwerk komen, omdat deze communicatie natuurlijk niet via de centrale antivirusmailscanner wordt geleid.

Filters omzeilen

Al jaren zijn er technieken bekend die bovengenoemde technische restricties met eenvoud en verve kunnen omzeilen. De meest eenvoudige mogelijkheid om de simpelste filtertechnieken te doorbreken is natuurlijk door een service aan te bieden op een poort die wel bereikbaar is. Als voorbeeld hiervan kan een bekende internetprovider dienen die als extra service voor haar leden op een paar shell-servers SSH toegang biedt op poort 80. Indien alleen gebruik wordt gemaakt van een filtering firewall die uitsluitend kijkt naar de TCP-header informatie zal op basis van die informatie beslist worden dat het hier om de opbouw van een HTTP-sessie gaat vanuit een client naar een server op het internet. Het feit dat er totaal geen HTTP-verkeer plaatsvindt, kan uitsluitend worden gezien door een applicatie-level firewall of door gebruik te maken van een proxy en de firewall uitsluitend HTTP-verkeer toe te laten van en naar die proxy. Hoewel proxies ook via een andere techniek gebruikt kunnen worden. Die techniek heet 'connect method' en wordt bijvoorbeeld gebruikt door het `ssh-http-proxy-connect`¹ commando. Origineel ontwikkeld om SSL versleuteld HTTP-verkeer tussen client en server ongestoord te laten plaatsvinden wordt het door deze en vergelijkbare commando's gebruikt om niet-tekst gebaseerde gegevens door de proxy heen te transporteren.² Hierdoor hoeft de informatie na het opbouwen van de verbinding niet eens meer in HTTP-verkeer verstopt te worden maar kan zonder verdere problemen door de proxy en firewall heen gaan. Het resultaat hiervan is dat veel klanten van de betreffende provider wel versleuteld met deze systemen kunnen communiceren om zodoende bijvoorbeeld hun persoonlijke e-mail te kunnen lezen of via IRC met anderen te kunnen chatten. Deze functie blijkt in de praktijk zo vaak te gebruiken dat de maker van putty, een SSH-client voor windows, dit standaard in hun client hebben ingebouwd.³

Tunnels en covert channels

Om meer flexibele en krachtiger tunnels mogelijk te maken worden de applicatieprotocollen die wel door de proxy en firewall heen mogen, gebruikt om bijvoorbeeld IP naar een speciale (tunnel)server op het internet te transporteren, om van daaruit doorgestuurd te worden (routing) naar het uiteindelijke doel, hierbij zal de server zichzelf als afzender opnemen en bijhouden van welke client de originele gegevens werden ontvangen. Omgekeerd worden ook de pakketten die terugkomen, door die server weer ingepakt in het applicatieprotocol om zonder problemen door de proxies en firewalls heen de originele client te bereiken. Protocollen zoals HTTP, FTP en DNS zijn allemaal al gebruikt om dergelijke tunnels te bouwen, maar zelfs protocollen van een lager niveau zoals ICMP, bekend van het ping-commando, zijn ervoor gebruikt. In het vervolg van dit artikel zullen deze allemaal behandeld worden.

Deze tunnels kunnen gezien worden als een vorm van wat in de literatuur ook wel een 'Covert Channel' (verscholen kanaal) wordt genoemd. Hierbij wordt uitgegaan van de volgende veel gebruikte beschrijving:⁴

'A covert channel is described as any communication channel that can be exploited by a process to transfer information in a manner that violates the system security policy.'

Hoewel bij deze definitie uit TCSEC nooit sprake is geweest van filtering policies als deel van een security policy lijkt deze beschrijving toch verbazend goed te passen bij wat hier gebeurt. Hoewel, als men goed kijkt naar de inhoud van de communicatie, het bestaan van de tunnel wel duidelijk aangetoond kan worden en op het moment dat deze communicatie niet versleuteld is met een sterk cryptografisch algoritme, ook de berichten zelf meegelezen kunnen worden. Een

echt verscholen kanaal (covert channel) zal zelfs deze aspecten van de communicatie proberen te verbergen, zodat het voor een buitenstaander moeilijk te detecteren is of er communicatie plaatsvindt en nog lastiger om na te gaan wat dan de betekenis van deze communicatie is.

HTTP tunnel

Het GNU `httptunnel` pakket biedt de gebruiker een open-source implementatie van een client en een server om een bidirectionele virtuele dataverbinding te creëren door middel van tunneling in HTTP request pakketten. Deze HTTP requests kunnen ook via een HTTP proxy verzonden worden. De `httptunnel` kan zodoende gebruikt worden om middels andere protocollen, zoals bijvoorbeeld SSH, TELNET of PPP, verbinding te maken met een andere computer buiten de firewall. Het is zowel in een UNIX-variant als voor Windows beschikbaar. Op dit moment ondersteunt het pakket slechts een gelijktijdige verbinding en kan dus niet door meerdere personen gelijktijdig gebruikt worden.

LOKI2

LOKI2⁶ is een proof-of-concept informatie-tunneling programma en is gebaseerd op een eerdere variant.⁷ Met behulp van deze client-server set kunnen eenvoudige shell-commando's in ICMP_ECHO / ICMP_ECHOREPLY en DNS namelookup query / reply verkeer getunneld worden. Het verkeer zal voor een netwerk protocol analyzer geen vreemd gedrag vertonen, maar de informatie in de pakketten kan ook niet zonder meer gelezen worden (mits het pakket gebruik maakt van de encryptiemogelijkheid die geboden wordt). Op dit moment is LOKI2 beschikbaar voor verschillende UNIX-varianten. Het is, zoals uit de beschrijving mag blijken, minder flexibel dan `httptunnel` aangezien er uitsluitend shell-commando's mee getransporteerd kunnen worden, maar soms is dit net voldoende.

Skeeve

Ook Skeeve⁸ is een proof-of-concept tool dat gebruik maakt van ICMP_ECHO en ICMP_ECHOREPLY om informatie te tunnelen. In tegenstelling tot LOKI2 werkt Skeeve door het converteren van de IP header. De protocolvlag wordt van TCP omgezet in ICMP_ECHO of ICMP_ECHOREPLY en er vinden nog wat kleine wijzigingen plaats. Er kunnen dus uitsluitend op TCP gebaseerde protocollen mee getunneld worden. Verder maakt de Skeeve client gebruik van een zogenaamde bounce server om de informatie uiteindelijk bij de Skeeve server te krijgen en omgekeerd. Om dit te laten werken wordt gebruik gemaakt van basis IP spoofing technieken. Het pakket dat naar de bounce server verstuurd wordt, heeft het adres van de uiteindelijke ontvanger als bron IP-adres staan. Er wordt van de bounce server dus verwacht dat deze zal antwoorden op het pakket en het zodoende doorstuurt naar de uiteindelijke ontvanger.

Het probleem met deze proof-of-concept code is natuurlijk dat voor elke tunnel er een nieuw gecompileerde client en server moet worden gemaakt, omdat de adressen van de client, de bouncer en de server hard gecodeerd in de code moeten worden opgenomen. Door de bovenstaande IP spoofing techniek gaat namelijk het originele source IP-adres verloren. Om het proof-of-concept tool dus werkelijk nuttig te maken zal de bounce op een zodanige manier moeten worden geïmplementeerd dat die informatie wel ergens in het verstuurd pakket bewaard blijft of moet gewoon van het bounce concept worden afgeweken.

Andere zogenaamde firewall bypassing tools zoals Active Port Forwarder, Covert Channel Tunneling Tool, Firepass, MsnShell en Web Shell kunnen ook op de website van het Gray World.Net Team⁹ gevonden worden, maar zullen hier niet verder besproken worden.

icmptunnel

Met behulp van icmptunnel¹⁰ kan een machine achter een firewall die bepaalde ICMP-pakketten wel doorlaat communiceren met een server op een systeem buiten de firewall. De meest interessante feature van deze software is dat het ICMP-pakket dat uiteindelijk gebruikt wordt om het TCP/IP-pakket in te pakken, volledig door de gebruiker te bepalen is (ICMP_ECHO, ICMP_ECHOREPLY, ICMP_TIMESTAMP, etc.). Door eerst na te gaan welke ICMP-pakketten door de firewall, zowel van binnen naar buiten als andersom, worden doorgelaten kan de gebruiker er uiteindelijk voor zorgen dat de tunnel ongestoord het werk kan doen.

NSTX

Met behulp van het NameServer Transmit Protocol (NSTX)¹¹ kan men IP tunnels maken die gebruik maken van DNS-query-en-reply-pakketten om de originele informatie in te pakken en uit te wisselen tussen de NSTX client en server. Hierdoor wordt algemeen IP-verkeer mogelijk op plaatsen waar alleen DNS-verkeer wordt toegestaan. Het protocol wordt hierbij zodanig geïmplementeerd dat er alleen RFC-valide pakketten worden gebruikt, hoewel deze wel herkenbaar zijn omdat ze meer informatie (moeten) bevatten dan normaal UDP DNS-verkeer.

De werking van NSTX is ongeveer gelijk aan die van de hierboven genoemde HTTP tunnel, maar gebruikt wel meer trucs. Dit is noodzakelijk omdat DNS UDP-pakketten normaal niet gebruikt worden om grote hoeveelheden data te transporteren. Ook bij deze implementatie wordt gebruik gemaakt van het client-server model. De NSTX server moet als een authoritative nameserver geregistreerd zijn voor een bepaald (sub)domein. Verkeer naar deze NSTX server moet via de lokale nameserver doorgegeven worden. Dit maakt de implementatie van deze tunnel een stuk lastiger dan de vergelijkbare HTTP tunnel. Toch kan de werkwijze eenvoudig beschreven worden.

De client die zich op het afgeschermd netwerk bevindt, vraagt een internetadres van een bepaalde server op het internet aan de lokale DNS server. Aangezien deze nameserver niet verantwoordelijk is voor dat externe domein zal deze de vraag moeten doorgeven aan de nameserver die wel verantwoordelijk is voor dat domein. In ons geval zal deze laatste server voorzien zijn van de NSTX server. De NSTX client zorgt ervoor dat de data die gecommuniceerd moet worden met de NSTX server in de hostnaam van het opgevraagde systeem is gecodeerd. Om volledig aan RFC 1035¹² te blijven voldoen mag deze hostnaam uitsluitend bestaan uit letters (zowel hoofd als kleine letters), cijfers en de hyphen ('-') en maximaal 63 karakters lang zijn, terwijl de totale lengte van de naam (full qualified, dus inclusief het domein) maximaal 255 karakters lang mag zijn. Als laatste restrictie mag het UDP-pakket waarin deze gegevens verzonden worden maximaal 512 bytes groot zijn. De NSTX server heeft het aanzienlijk eenvoudiger. Deze gebruikt een TXT resource record om het antwoord in te verpakken en terug te sturen naar de server. Een dergelijk record kan willekeurige (gecodeerde) informatie bevatten, waarbij de enige restrictie is dat de lengte van het totale UDP-pakket maximaal 512 bytes kan zijn. Dit laatste betekent dat er vrijwel zeker fragmentatie over deze tunnel moet plaatsvinden, waardoor de performance waarschijnlijk suboptimaal zal zijn, maar daar zal de gebruiker hoogstwaarschijnlijk geen probleem mee hebben.

Gevolgen voor WiFi hotspots

We hebben in het voorgaande de nadruk gelegd op het doorbreken van policies die zijn opgelegd door middel van technische filters in de vorm van firewalls en proxyservers. Maar juist het gebruik van een DNS tunnel heeft ook gevolgen voor commerciële aanbieders van zogenoemde hotspots zoals een onderzoek¹³ recentelijk heeft aangetoond. DNS-verkeer op deze netwerken, en vandaar ook naar en van het internet, wordt al toegestaan voordat de gebruiker betaald heeft. Dus voordat autorisatie heeft plaatsgevonden wordt het clients al toegestaan om DNS-vragen te stellen en informatie te ontvangen. Niet alleen van het LAN (wireless netwerk) maar ook vanaf het internet. Hierdoor wordt het dus mogelijk om een NSTX tunnel op te zetten en zodoende gratis van de betreffende faciliteiten gebruik te maken.

Malware

Natuurlijk worden vergelijkbare technieken ook toegepast in de latere fasen van een inbraak. De cracker kan door middel van dit soort tunnels uiteindelijk controle houden over de systemen waarop is ingebroken en een voldoende hoog niveau van toegang is verkregen (in de meeste gevallen root of administrator-permissies). Verschillende backdoor tools en Trojans hebben in ieder geval de mogelijkheid om een covert channel te gebruiken voor communicatie. Niet in alle gevallen gaat het hier om een tunnel. Een interessant voorbeeld is de 'Q' Trojan¹⁴ die gebruik maakt van 'raw IP'. In plaats van te luisteren naar commando's die binnenkomen over een bepaalde poort, snift 'Q' op het netwerksegment en reageert op IP-pakketten die een speciale parameter hebben. De data van deze pakketten bevatten dan de versleutelde instructies die door de 'Q' daemon zullen worden uitgevoerd.

Forensisch onderzoek

Er zijn dus vele verschillende mogelijkheden om informatie uit te wisselen door verschillende beveiligingssystemen waar dagelijks op vertrouwd wordt. Firewalls en proxyservers bieden meestal slechts beveiliging tegen de buitenwacht. Natuurlijk komt dat ook overeen met de originele ontwerpcriteria van deze apparatuur. Om na te gaan of er gebruik gemaakt wordt van 'verscholen kanalen' om te communiceren met systemen buiten de firewall zal niet vertrouwd kunnen worden op de header-informatie van de pakketten, maar zal diep in de inhoud van deze pakketten gekeken moeten worden of een analyse van het verkeer moeten plaatsvinden. Het zal bijvoorbeeld normaal nooit voorkomen dat er heel veel communicatie plaatsvindt tussen een systeem en een externe DNS server. Mocht dat wel het geval zijn dan hebben we hoogstwaarschijnlijk te maken met een tunnel. Ook de inhoud van pakketten kan een indicatie geven of het valide verkeer is of dat er sprake is van een tunnel. Maar om daarvan gebruik te mogen maken moet van te voren het personeel (of andere gebruikers van het betreffende netwerk) op de hoogte worden gebracht dat de eigenaar van het netwerk het verkeer over dat netwerk kan monitoren. Verwacht trouwens niet dat er al te veel geleerd kan worden uit deze gegevens. Zodra gebruikers het concept van een tunnel hebben begrepen, nemen ze al snel de stap om de gegevens die over deze tunnel worden verstuurd ook te versleutelen met een sterk cryptografisch algoritme (al dan niet ingebouwd in het betreffende tunnelproduct).

Conclusies

Natuurlijk blijft het altijd lastig om medewerkers te controleren. De vraag is ook of het werkelijk noodzakelijk is om deze controle door middel van apparatuur en software hard af te dwingen. In

veel gevallen zal de apparatuur dit ontduiken van het informatiebeveiligingsbeleid alleen aantonen maar niet voorkomen. Natuurlijk is het transporteren van informatie over een netwerk eenvoudiger dan deze informatie op te slaan op een of ander medium en dit mee te nemen. Maar onderschat nooit de bandbreedte van een attachékoffertje gevuld met cd-roms of dvd's. Welk bedrijf controleert medewerkers regelmatig of ze geen bedrijfseigendommen meenemen door de poort? Ik ben dit uitsluitend tegengekomen bij onderzoeksinstellingen en bedrijven die producten en diensten leveren aan defensie. Maar zelfs in die gevallen werd er alleen oppervlakkig gekeken naar de inhoud van de rugzak en werd de laptop in veel gevallen niet eens aangezet, laat staan dat de inhoud van de dvd-speler werd gecontroleerd of het filesysteem werd doorzocht.

Ik denk dan ook dat het een valide vraag is of dat we in alle gevallen medewerkers moeten verbieden om even wat e-mail te lezen bij hun eigen ISP. Natuurlijk zal misbruik en sabotage aangepakt moeten worden maar dat gaat meestal beter door een goed informatiebeveiligingsbeleid vast te leggen en op basis hiervan regelmatig onaangekondigde audits uit te (laten) voeren dan erg veel euro's te spenderen aan de aanschaf van nieuwe apparatuur om uw medewerkers te controleren. Mensen blijven in ieder geval op dit moment en ook in de voorzienbare toekomst de zwakste schakel. Machines kunnen heel veel controleren, maar zullen altijd achterlopen bij de laatste ontwikkelingen.

Noten

1. Informatie over ssh-http-proxy-connect: <http://mirrors.ccs.neu.edu/cgi-bin/unixhelp/man-cgi?ssh-http-proxy-connect+1>
2. Informatie over CONNECT Method: <http://www.kb.cert.org/vuls/id/150227>
3. Informatie over Putty:
<http://the.earth.li/~sgtatham/putty/0.54/html/doc/Chapter4.html#4.14>
4. Deze definitie is overgenomen uit het zogenaamde Orange Book. Een on-line referentie kan gevonden worden via deze URL:
<http://kernel.us.themoes.org/pub/linux/libs/security/Orange-Linux/refs/Orange/OrangeI-II-8.html>
5. httptunnel is beschikbaar via: <http://www.nocrew.org/software/httptunnel.html>
6. LOKI2 is beschikbaar via: <http://www.phrack.org/show.php?p=51&a=06>
7. Informatie over Project Loki: <http://www.phrack.org/show.php?p=49&a=06>
8. Skeeve is beschikbaar via: http://gray-world.net/poc_skeeve.shtml
9. <http://gray-world.net/index.shtml>
10. icmptunnel kan o.a gevonden worden via: <http://www.securityfocus.com/tools/1310>
11. NSTX kan gevonden worden via: <http://nstx.dereference.de/nstx/>
12. RFC 1035 kan gevonden worden via: <http://www.ietf.org/rfc/rfc1035.txt>
13. Zie <http://www.computable.nl/artikels/archief4/d23rr4ki.htm>
14. 'Q' kan gevonden worden via: <http://mixter.void.ru/code.html>