Fakulta matematiky, fyziky a informatiky
Mlynská dolina, 842 48 Bratislava

Tunneling

Daniel Adam, Mária Gemeranová, Ladislav Maršík, Peter Sucha

Obsah

Tunneling....1
1. Úvod....4
2. Tunelovanie....5
Využitie tunelovania....5
Podrobnejší princíp....5
Analógia – donášková služba....6
Narušenie OSI referenčného modelu....6
Datagramové a streamové tunelovacie protokoly....6
Port forwarding....7
3. VPN....9
Analógia: ostrovy....9
Princíp....9
Typy VPN....10
Časti VPN....11
4. Tunelovacie protokoly 2. vrstvy....13
4.1. PPTP (Point-to-Point Tunneling Protocol)....13
Zapuzdrenie informácie....13
Vytvorenie tunela....13
Kontrola spojenia....14
Bezpečnosť....15
Plusy a mínusy....15
4.2. L2F (Layer 2 Forwarding)....15
4.3. L2TP....15
Topológia....16
Vysvetlenie niektorých pojmov....16
Ako L2TP pracuje. 16
5. Tunelovacie protokoly 3. vrstvy....20
5.1 GRE (Generic Routing Encapsulation)....20
Univerzálnosť....20
Zapuzdrenie....20
Komunikácia....20
Bezpečnosť....21
5.2. IPSec (Internet Protocol Security)....21
Funkcie....22
IPSec protokoly....22
Zapuzdrenie....22
GRE cez IPSec....23
L2TP cez IPSec....23
6. Tunelovanie a IPv6 24
IPv6 a IPv4....24
Tunneling....24
IP tunneling....26
Protokol 6to4....26
Protokol 6in4....29
Ako sa dostať na IPv6 internet s naším IPv4 pripojením?....29
IPv6 a tunelovanie - záver....32
7. SSH Tunneling....33
SSH (Secure Shell)....33
Ako funguje....33

Tunneling cez SSH....34
8. NAT (Network Address Translation)....35
Úvod do NAT....35
Prečo NAT....35
Adresovania a rôzne druhy NAT....36
Zápory NAT....38
NAT a Tunneling....38
9. Iné tunelovanie....40

1. Úvod

V našom projekte sa pokúsime spraviť hlboký náhľad do techniky zvanej tunelovanie, hojne využívanej v počítačových sieťach. Najprv si podrobne vysvetlíme princípy a charakteristiky a následne sa vrhneme na využitie, nezanedbajúc aj okolitú problematiku a situáciu, v ktorej tunelovanie prichádza na scénu. Cieľom je postupne a v celej kráse ukázať, čoho je táto metóda schopná. Tiež chceme čitateľa oboznámiť s najviac používanými tunelovacími protokolmi, podľa možností tak, aby po prečítaní referátu bolo aj naštudovanie nami nezahrnutých protokolov jednoduchšie.

Využitie tunelovania je veľmi široká téma. Aj o jedinom tunelovacom protokole by sa dal spraviť celý referát. Koncept sme si ale navrhli tak, že sa budeme sústrediť na zopár dnes asi najaktuálnejších využití tunelovania a poskytneme dostatok informácií na predstavu ďalších možných aplikácií. Tým vytvoríme neznalému čitateľovi dostatočný prehľad a skúsenému čitateľovi pohľad z našej strany a dostatok konkrétnych informácií.

2. Tunelovanie

Počítačové siete a hlavne internet dnes ponúkajú spektrum možností, ako získavať informácie, komunikovať, či pracovať aj na obrovské vzdialenosti. So stúpajúcou kvalitou služieb však stúpajú aj požiadavky. Nie vždy sa nám zdá práca dostatočne pohodlná, komunikácia dostatočne bezpečná, ba priam niekedy sa môžeme cítiť veľmi obmedzení používanými zariadeniami či softvérom, alebo sklamaní nekompatibilitou sieťových prvkov. Naplnenie našich predstáv nám ponúka metóda zvaná tunelovanie (tunneling).

Tunelovanie je proces prenášania dát s využitím zaobalenia nákladného protokolu (payload protocol) do prenosného protokolu (delivery protocol). Najčastejšie v prípade, keď je nákladný protokol určený na prenos po sieti iného typu, a teda je bez tunelovania nemožné dostať náklad na požadované miesto.
Známa interpretácia tunelovania je aj: Prenos dát určených pre súkromnú sieť prostredníctvom verejnej siete spôsobom, že prenosové uzly verejnej siete nevedia, že ide o komunikáciu súkromného charakteru. Verejná sieť v tomto prípade spravidla býva internet a pod súkromnou sieťou si môžeme predstaviť lokálnu sieť (LAN), ktorá chce komunikovať so vzdialeným počítačom, takže napr. firemnú sieť.

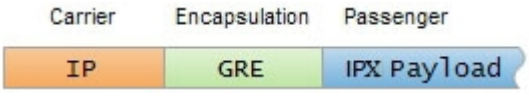
Využitie tunelovania

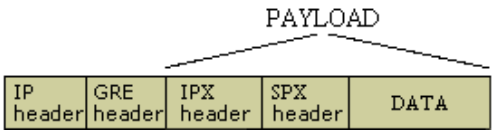
Už sme naznačili viaceré možnosti využitia.
Najznámejšie je spomínané prepojenie počítačov do súkromnej siete na podnikové alebo iné účely na veľké vzdialenosti. Práve vzdialenosť bola najväčším podnetom zavedenia tunelovania. Spoločnosti potrebujú na svoj chod komunikáciu po sieti LAN a tá sa dá zabezpečiť len priamym spojením cez prenajaté telefónne linky alebo optické káble. Takéto prepojenie na veľkú vzdialenosť by bolo pre spoločnosť veľmi nevýhodné, tobôž keby niekedy presťahovala sídlo. Tunelovanie vie ľubovoľný počet počítačov pripojených na internet navzájom prepojiť do siete, kde môžu navzájom komunikovať rovnako, ako by boli na jednom sieťovom segmente. Takáto sieť sa nazýva virtuálna privátna sieť (virtual private network, VPN) a budeme sa ňou zaoberať neskôr.
Tunelovanie sa môže využiť aj na zvýšenie bezpečnosti komunikácie. Protokol (napr. emailový, POP3), ktorý v sebe nezahŕňa dodatočné ochranné prvky možno zaobaliť napr. do protokolu SSH (Secure Shell) a stiahnuť si tak zo servera poštu v podobe zašifrovaných dát (za predpokladu, že server podporuje SSH).
Tunelovanie, ako univerzálny nástroj, môžeme využiť aj pri mnohých ťažkostiach s kompatibilitou. Za všetky spomenieme tú najzávažnejšiu a tou je dlho zamýšľaný prechod z IPv4 sieťového protokolu na IPv6. Popri zavádzaní IPv6 sa komunikácia so staršou verziou protokolu rieši využitím špeciálnych tunelovacích protokolov na zaobalenie paketov IPv4 do IPv6 a naopak.
V neposlednom rade sa tunelovanie využíva na obídenie niektorých reštrikcií, ktoré nám môžu brániť v práci (napr. obmedzenia firewallu alebo proxy servera). Pokiaľ sa dokážeme pripojiť napr. na SSH server, nie je nič jednoduchšie ako zaobaliť „nechcené“ protokoly do SSH a oklamať tak kontrolný orgán.

Podrobnejší princíp

Pre lepšie pochopenie ďalších kapitol uvediem princíp spoločný pre všetky tunelovania.
Majme dáta, ktoré chceme preniesť (napr. textová správa). Ďalej máme protokol, ktorým chceme komunikovať a je podporovaný oboma stranami (napr. IPX, pokiaľ sa napr. pripájame na sieť Novellovskej architektúry) – payload protokol (passenger protocol). Správu sme ale nútení poslať cez inú sieť, obyčajne internet, ktorý využíva TCP/IP architektúru a teda sieťový protokol IP – delivery protokol (carrier protocol).
Tu prichádza na rad tzv. tunelovací protokol (tunneling protocol alebo aj encapsulation protocol) – protokol, ktorý zabezpečí vytvorenie „tunela“, cez ktorý správa môže prejsť. V tomto prípade to

môže byť protokol GRE. Jeho úlohou je správu, ktorá má IPX hlavičku, zaobaliť do IP protokolu. Spraví to tak, že pridá najskôr svoju GRE hlavičku s informáciami o posielaných dátach (IPX) a napokon pridá celému paketu ešte IP hlavičku a ten je tak pripravený cestovať po internete (ako na obrázkoch).





Druhá strana, tiež pripojená na internet, najskôr odstráni IP hlavičku, z GRE hlavičky načíta informácie o dátach, ktoré sú vo vnútri a napokon pracuje s IPX paketom rovnako, ako by mu práve prišiel z lokálnej siete.

Analógia – donášková služba.

Predstavme si donáškovú službu, napr. UPS. Pekná analógia je predstaviť si pod payloadom to, čo hodláme doručiť – napr. televízor s potrebným príslušenstvom. Čo spraví donášková služba? Zabalí televízor do kartónovej škatule, na ktorej sú informácie o produkte (tunelovací protokol), a ten naloží do svojej dodávky, ktorá je schopná televízor preniesť (delivery protokol). Až po dodaní a rozbalení škatule máme čo dočinenia s televízorom, ktorý treba uložiť na správne miesto a manuálom k nemu, ktorý ešte treba spracovať, čo môže byť analógia so spracovaním payload protokolu.

Narušenie OSI referenčného modelu

Jednou z charakteristík tunelovania je, že svojou podstatou narúša klasické rozloženie vrstiev v referenčných modeloch.
Tunelovací protokol často oddeľuje dva protokoly sieťovej vrstvy (napr. IPX a IP), niekedy dokonca podloží pod protokol spojovej vrstvy (napr. PPP) protokol sieťovej vrstvy (IP). Vyplýva to priamo zo zámeru - využiť iné sieťové služby. Teoreticky sa táto nezrovnalosť opisuje tak, že (v prvom prípade) protokol pracuje spolu s delivery a payload protokolmi na sieťovej vrstve (IP-GRE-IPX), alebo (v druhom prípade) pokladáme tunelovací protokol za protokol na spojovej vrstve (napr. L2TP, layer 2 tunneling protocol), pričom ale tajne vieme, že prakticky všetky tunelovacie protokoly pracujú na transportnej, teda štvrtej vrstve, pretože poskytujú isté možnosti a pravidlá na prenos dát a svoje dáta posúvajú sieťovej vrstve.

Datagramové a streamové tunelovacie protokoly

Rovnako ako je to pri klasickej sieťovej komunikácií, aj tunelovacie protokoly môžeme rozlíšiť podľa toho, či sa posielajú jednotlivé fragmenty správy – datagramy, alebo ide o súvislý prúd dát – stream. Princíp vytvorenia tunela je rovnaký.
Treba azda ešte dodať, že datagramový protokol neznamená nutne, že pracuje formou negarantovaného doručenia bez vytvorenia spojenia. Správy môžu byť dodávané rovnako cez connection-oriented TCP/IP ako cez connectionless UDP/IP (podľa konkrétneho tunelovacieho

protokolu). Navyše sa pri vytvorení privátnej siete VPN vytvorí kontrolné spojenie aj na nižšej úrovni, uzla k uzlu (PPP). Slovo datagram si teda v tomto zmysle môžeme podľa potreby zameniť za slovo paket a značí len fragmentáciu posielaných dát.

Medzi datagramové tunelovacie protokoly patria:

GRE (Generic Routing Encapsulation)
IPSec (Internet Protocol Security)
PPTP (Point-to-Point Tunneling Protocol)
PPPoE (Point-to-Point Protocol over Ethernet)
PPPoA (Point-to-Point Protocol over ATM)
L2F (Layer 2 Forwarding)
L2TP (Layer 2 Tunneling Protocol)
6to4 (IPv6 over IPv4 as protocol 41)
Teredo (IPv6 over UDP over IPv4)
MPLS (Multi-Protocol Label Switching)
GTP (GPRS Tunnelling Protocol)
IP in IP
IEEE 802.1Q (Ethernet VLANs)
DLSw (SNA over IP)
XOT (X.25 datagrams over TCP)
AYIYA (Anything In Anything)

Medzi streamové tunelovacie protokoly patria:

SSH (Secure Shell)
TLS (Transport Layer Security)
SSL (Secure Sockets Layers)
SOCKS (Sockets)
HTTP CONNECT command
Proxy protokoly (Microsoft Proxy Server's Winsock Redirection, WinGate Winsock Redirection Service, ...)

Všetky tieto protokoly majú svoje špecifické využitie, niektoré riešia len jeden konkrétny problém, iné možno použiť univerzálnejšie (GRE). Pre svoju rôznu funkcionalitu sa niektoré začali používať v spolupráci s ostatnými a vytvorili tak silné dvojice (GRE + IPSec, PPTP + GRE, L2TP + IPSec, ...). A niektoré, ako sa neskôr dozvieme, vznikli práve ako syntéza alebo nadstavba ostatných a sú dnes najviac používaným štandardom (L2TP).
Z hľadiska funkcionality by sa dali tunelovacie protokoly (nie jednoznačne) rozdeliť na:

- Protokoly na vytvorenie privátnej siete VPN: PPTP, L2F, L2TP, IPSec
- Protokoly na vytvorenie bezpečného spojenia: IPSec, SSH, TLS, SSL
- Protokoly riešiace konkrétny problém: 6to4, PPPoE, PPPoA, GTP, ...

Port forwarding

Na tomto mieste ešte môžeme spomenúť termín úzko sa spájajúci s tunelovaním, a to je port forwarding. Hoci sa niekde vraví, že tunneling a port forwarding odkazuje na to isté, pokladáme port forwarding za technológiu, ktorá sa skôr dá využiť popri tunelovaní. Najmä pri SSH tunelovaní – je možné, aby sme komunikáciu z nášho lokálneho servera na istom porte posunuli priamo na špecifický port na serveri, na ktorý sa pripájame (cez SSH tunel) a naopak. Vytvoríme tak dojem, že máme vzdialený zdroj priamo na našom lokálnom serveri (localhost). Tomuto sa budeme viac venovať v podkapitole SSH tunelovanie. Port forwarding je veľmi využiteľná metóda pri takýchto

pripojeniach, ale priamo sa princípu tunelovania netýka.

V ďalšom opíšeme hlavných zástupcov tunelovacích protokolov, ich funkciu vo svete sietí ako aj spôsoby ich použitia. Našou snahou je výber najpoužívanejších spôsobov tunelovania a podrobnejší náhľad do ich problematiky. Protokoly, ktoré nezahrnieme, sú tiež používané, avšak využívajú sa vo viac konkrétnych prípadoch a ich vysvetlenie by nesplnilo účel našej práce.

K jednotlivým protokolom sa dostaneme cez najdôležitejšie využitia tunelovania. Začneme VPN sieťami a tam najviac využívanými protokolmi. Pokračovať budeme problémom zavedenia IPv6 protokolu. Špeciálnu sekciu vyhradíme aj streamovému SSH tunelovaniu a na záver sa budeme venovať aj problémom s tunelovaním a prekážkam, ktoré pre tunelovanie predstavuje NAT adresácia. Bonusom pre čitateľa bude náhľad do globálneho problému adresovania na internete, keďže sa o ňom budeme rozprávať v časti IPv6 a tunneling, ako aj v časti, kde si podrobnejšie popíšeme NAT adresáciu.

3. VPN

Rozvoj sietí a internetu vnukol spoločnostiam a podnikateľom mnoho možností na to, aby si zväčšili záber. Dostalo sa to až tam, že dnes sa už prakticky neuvažuje nad podnikaním len v úzkom regióne. Požiadavkou veľkých firiem je dnes spoľahlivo, rýchlo a bezpečne komunikovať na diaľku z viacerých sídiel po celom svete.

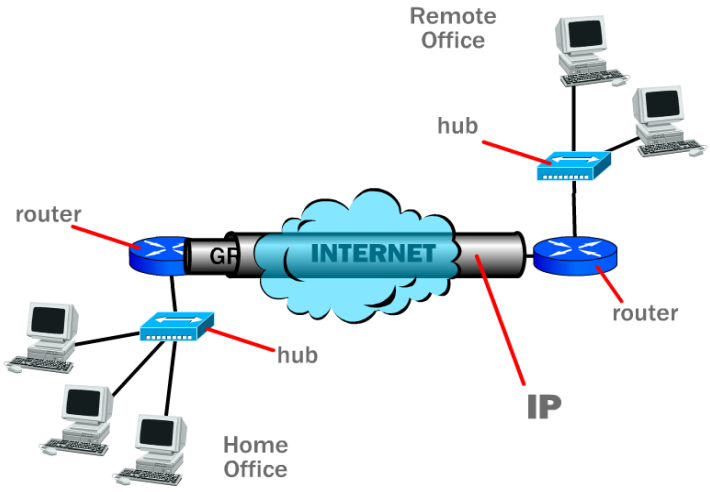
Analógia: ostrovy

Problém so zosieťovaním vzdialených uzlov prenajatými linkami (leased lines) na vytvorenie privátnej siete sme si už popísali vyššie, je to samozrejme krajne nevhodné riešenie. Peknou analógiou je: ostrovy ako lokálne LAN siete a oceán ako internet. Mnohé lode prepravujúce stovky pasažierov predstavujú bežnú komunikáciu po internete a pripájanie sa na jednotlivé servery (prístavy). Takáto loď neposkytuje pocit súkromnej prepravy medzi dvoma ostrovmi, a určite ani nie bezpečnej, pri toľkých ľuďoch naokolo. Pokiaľ si dva susedné ostrovy chcú zabezpečiť bezpečnú, priamu a rýchlu prepravu, môžu si postaviť most. Je to síce veľká investícia, ale výhody sú natoľko žiaduce, že takéto mosty sa stavajú bežne (optické káble, ISDN, ...). Problém však je, keď príde potreba spojiť sa s ostrovom, ktorý je príliš ďaleko.
Postaviť most je veľmi neekonomické, ak nie nemožné. Namiesto toho by sme mohli využiť úplne inú technológiu. Predstavte si, že máte súkromnú ponorku, ktorá je rýchla, prenosná, spoľahlivá a dokáže vás ukryť pred ostatnými ľuďmi. Napriek tomu, že premáva v oceáne ako všetky ostatné prostriedky, môže sa kedykoľvek použiť na prepravu medzi ľubovoľnými ostrovmi, navyše nepozorovane a bezpečne.

Takto by sa dala popísať aj VPN. Je to súkromná sieť, využívajúca verejnú sieť (internet) na prepojenie užívateľov. Tí môžu spolu komunikovať akoby boli v rámci jednej lokálnej siete. Je to možné práve vďaka tunelovaniu, metódou zaobalenia protokolov, ktorú sme si popísali vyššie.

Princíp

Počítače na lokálnej sieti komunikujú na jednom sieťovom segmente, je to tzv. súkromná sieť. Každá informácia môže ísť priamo z počítača na počítač, a využíva sa pritom adresácia typická pre LAN (192.168.x.y). Samozrejme, pred neporiadkom, ako je internet, sú všetky tieto počítače oddelené routrom, ktorý im sprostredkúva pripojenie a elegantne zakrýva ich IP adresy. Napriek tomu, my by sme sa do takejto siete radi pripojili aj cez verejnú sieť, internet.
Predstavme si, že chceme prepojiť dve vzdialené LAN siete, prostredníctvom routrov. Chceme teda posielať IP pakety (alebo prípadne aj informácie z nižšej vrstvy) cez internet na druhú LAN sieť s možnosťou adresovať aj konkrétny počítač alebo zariadenie na tejto sieti. Tunelovacie protokoly komunikujúcich routrov vedia náš IP paket (spolu s presnou informáciou, na ktorý počítač /



zariadenie sa chceme pripojiť, teda typu 192.168.x.y) zaobaliť a pridať mu novú IP hlavičku na prenos cez internet. Inak povedané, vytvoria si fiktívny tunel. Cez tento „tunel“ pošlú pripravený paket. Ostatné uzly na ceste nevedia, že vnútri tohto tunela (pod IP hlavičkou a hlavičkou tunelovacieho protokolu) sa nachádza úplne nová informácia o tom, kam má payload smerovať. Takže, keď sa paket dostane na router na druhej strane, tento ho rozbalí, nájde pod hlavičkou GRE druhú IP hlavičku, takže už len po LAN sieti posunie paket tam, kam má ísť. Na obrázku vidíme takúto VPN sieť využívajúcu GRE tunel.

Výhody teda poznáme. Jej najväčšou doménou je tzv. škálovateľnosť, tj. možnosť pridávania virtuálneho výkonu, v tomto prípade neobmedzená vzdialenosť dvoch uzlov. Ďalšie môžeme zhrnúť slovami: spoľahlivosť, bezpečnosť, manažment siete.

Typy VPN

Môžeme rozlíšiť tri typy VPN podľa toho, prepojením akých uzlov vznikla:

Remote-access VPN

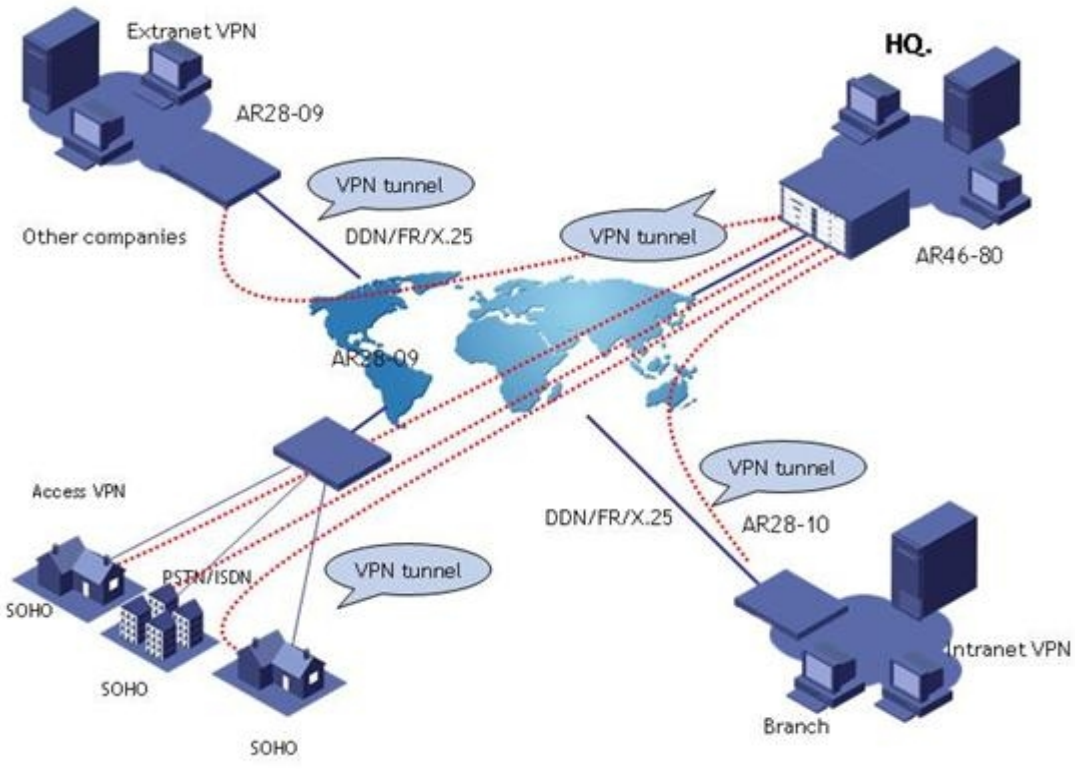
Prepojenie jednotlivého počítača na vzdialené centrum. Vhodné ak sa zamestnanci spoločnosti často vyskytujú mimo jej sídla. Spoločnosť si podľa potreby môže zriadiť aj poskytovateľa služieb na pripojenie do siete (ESP, enterprise service provider), ktorý vytvorí server NAS (network access server, môžeme ho všeobecne nazvať aj POP - point-of-presence). Zároveň dodá zamestnancom príslušný klientsky softvér, pomocou ktorého sa ľahko pripoja na NAS a komunikujú vo VPN.

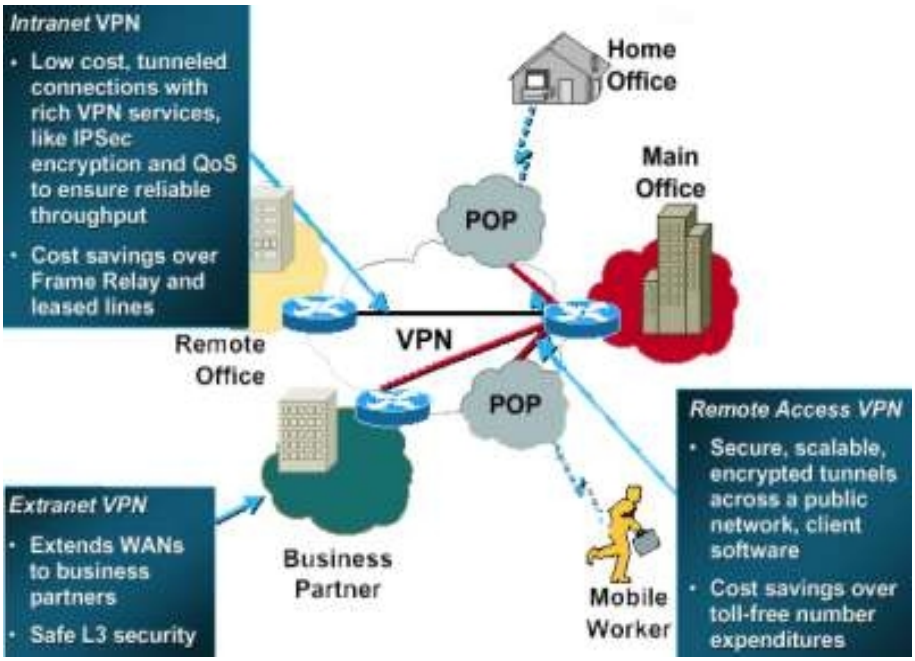
Site-to-site VPN, intranet-based

Prepojenie viacerých sídel spoločnosti. Takzvaný firemný intranet sa vytvorí, pokiaľ sa prepoja viaceré firemné LAN siete. Využívajú sa spoľahlivé VPN s kvalitným šifrovaním. Väčšinou sú koncové body tunela dva routre.

Site-to-site VPN, extranet-based

Prepojenie partnerských spoločností, napr. spoločnosť a jej dodávatelia.





Časti VPN

Pre úplnosť treba ešte spomenúť, aké súčasti využíva virtuálna privátna sieť na svoje fungovanie po verejnej sieti. Bezpečnosť VPN zaručujú tieto komponenty:

Firewall

Rôzne metódy šifrovania

Rovnako ako firewall, aj šifrovanie môže byť riešené hardvérovo, zadovážením si špeciálnych, na to určených routrov a mechanických firewallov. Sú to často riešenia od renomovaných sieťových firiem ako Cisco a poskytujú aj mnoho iných služieb.

AAA server

Dobre navrhnutá VPN môže využiť aj takýto server. Pri pokuse o pripojenie má na starosti: Autentifikáciu (kto sa pripája), autorizáciu (čo smie urobiť), accounting (čo aktuálne užívateľ robí).

Tunelovacie protokoly

Samozrejme, tvoria podstatu celej VPN. Ich vhodný výber môže navyše aj vylepšiť alebo priamo zabezpečiť bezpečnosť v jednej zo spomínaných sfér – šifrovanie, autentifikácia. Pre tieto účely sa dnes spravidla používa IPSec.

Ďalšie dôležité súčasti na výstavbu VPN (záleží od typu) sú: klientský softvér pre individuálnych užívateľov, VPN server, prípadne NAS server poskytovaný providerom na pripojenie užívateľov, centrum manažmentu siete, … Väčšina spomínaných komponentov je však voliteľná a závisí od návrhu siete, či sa využijú.

Toľko ku charakteristikám VPN. Poďme sa teraz pozrieť hlbšie na to, ako fungujú. Myšlienku tvoria tunelovacie protokoly. Podľa charakteru VPN sa používajú konkrétne protokoly.

Na site-to-site VPN, kde základ tvorí prepojenie dvoch routrov, sa obyčajne používa protokol GRE v kombinácií s IPSec.

Na remote-access VPN, kde sa vzdialený užívateľ pripája na server, sa často využíva pripojenie pomocou PPP (Point-to-Point protocol), ktorý je potrebný na volanie na NAS server, a z toho vyplýva využitie tunelovacích protokolov, ktoré zapuzdrujú PPP na 2. vrstve. Ide o protokoly PPTP, L2F, L2TP.

Ale takéto rozdelenie vôbec nie je presné. Využitie konkrétneho protokolu závisí skôr od konkrétnej architektúry a požiadaviek na VPN. Jasné sú len niektoré zákonitosti: ak chceme zapuzdriť protokol

PPP, treba použiť jeden z protokolov PPTP, L2F, L2TP. Ak nám ide o bezpečnosť, treba použiť IPSec. A podobne.

4. Tunelovacie protokoly 2. vrstvy

Najskôr sa vrhnime práve na protokoly zapuzdrujúce PPP, alebo všeobecne, protokoly zapuzdrujúce na 2. vrstve. Na úvod si o týchto protokoloch povedzme, že reprezentujú dva rôzne smery: PPTP od združenia PPTP forum zvučných mien ako Microsoft, a L2F od spoločnosti Cisco, ktoré splynuli v jedno riešenie, vznikom ich kombinácie: L2TP.
L2TP v sebe okrem univerzálneho GRE protokolu využíva aj IPSec na bezpečnosť, a najmä preto sa s obľubou využíva aj na site-to-site spojenie. Skutočnosť je aj tá, že L2TP prakticky vytlačil svojich dvoch predchodcov PPTP a L2F a stal sa štandardom. Preto si na PPTP a L2F vysvetlíme len princíp a zameriame sa na L2TP.

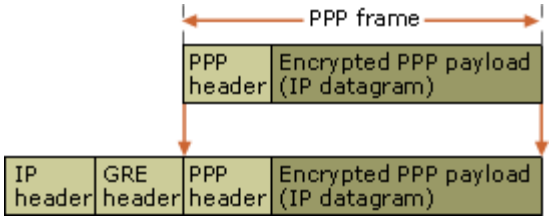
4.1. PPTP (Point-to-Point Tunneling Protocol)

Dnes už nie veľmi perspektívny, ale dosiaľ hojne využívaný protokol PPTP bol vytvorený konzorciom PPTP forum, ktoré sa skladá z organizácií: Ascend Communications, Microsoft Corporation, 3Com/Primary Access, ECI Telematics, a U.S. Robotics. Jeho popularita zrejme pramení v tom, že to bol prvý VPN protokol podporovaný Microsoftom v čase dial-up networking na operačných systémoch Windows (PPTP je podporované od Windows 95 doteraz). Plná podpora v Linuxových jadrách sa zaviedla až nedávno v r. 2005 kvôli opletačkám s právami na použitie.

Tunelovanie za účelom vytvorenia VPN nie je len o zapuzdrení a poslaní informácie. Nemenej dôležitá je aj neustála kontrola dostupnosti komunikujúcich uzlov a tiež bezpečnosť komunikácie (autentifikácia, šifrovanie). Aj preto sa na PPTP pozrieme z týchto troch pohľadov.

Zapuzdrenie informácie

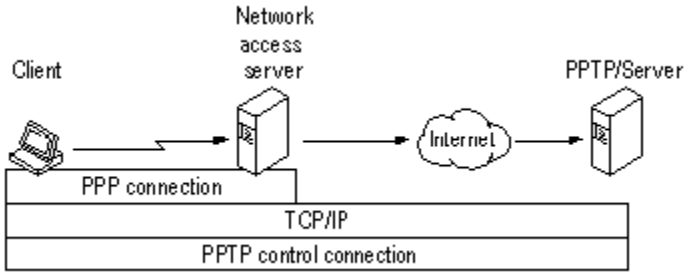
Dáta, ktoré chceme odoslať sú v IP datagrame, ktorý je navyše obalený protokolom PPP, pretože sa ním chceme pripojiť na server, ktorý nám poskytne komunikáciu s LAN sieťou (máme payload s PPP hlavičkou). Je navyše šifrovaný, o čom si povieme neskôr.
PPTP zapuzdrenie nie je vlastne nič iné, ako využitie modifikovanej verzie GRE na vytvorenie nového IP datagramu. Teda je pridaná hlavička GRE a hlavička IP (delivery protokol). Takto pripravené dáta (na obrázku) môžu byť odoslané „tunelom“ a na druhej strane odpuzdrené a poslané po LAN sieti na príslušný uzol.



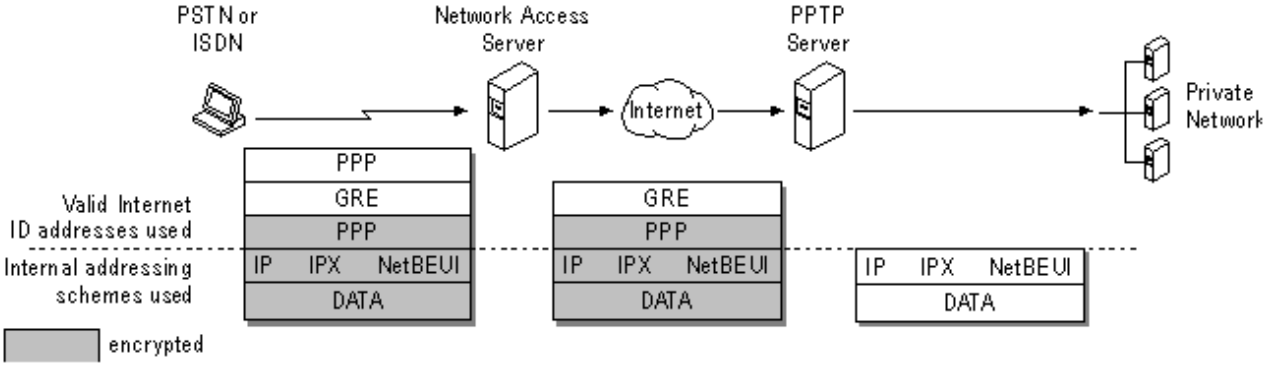
Vytvorenie tunela

Vysvetlíme si to na typickej situácií pripojenia do VPN. Uvažujme PPTP klienta, ktorý sa pripája na internet cez NAS server od svojho ISP (internet service provider), a chce sa spojiť s počítačom na vzdialenej LAN sieti. Spraví tak pripojením sa na PPTP server tejto LAN siete, pripojený na internet (ako sme si spomínali, firmy vytvárajú takéto servery práve pre tento účel – aby sa na ne pripájali ich zamestnanci). Uvedomme si, že teoreticky sa môže tunelovať, aj len v rámci jednej LAN siete, ale situácia s NAS serverom cez internet je názornejšia.
Klient sa najprv pripojí PPP pripojením na NAS svojho ISP. Od toho momentu môže klient cez NAS komunikáciou TCP/IP dostávať pakety z internetu. Hneď potom nasleduje druhé volanie

formou TCP/IP, už cez existujúce spojenie a priamo na PPTP server. Toto volanie je takzvané kontrolné volanie, za účelom vytvorenia spojenia - tunela (o kontrolných volaniach si povieme nižšie) a ešte neprenáša naše dáta (obrázok).

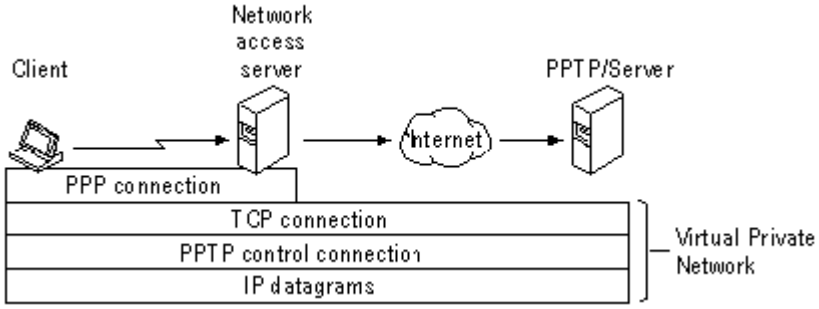


Akonáhle je tunel vytvorený, môže začať posielanie zapuzdrených paketov s IP-GRE hlavičkou (tunelovanie).



Kontrola spojenia

Vytvorenie tunela je len jedným z mnohých kontrolných volaní. V skutočnosti pri komunikovaní PPTP existujú naraz dve pracovne nezávislé spojenia (samozrejme, obe na sebe existenčne závisia).
PPTP kontrolné spojenie
PPTP tunel



Tunel sme si už vysvetlili vyššie. Kontrolné spojenie cez TCP/IP (naznačené na obrázku) pozostáva z preposielania TCP/IP datagramov medzi PPTP klientom a PPTP serverom. Dáta, ktoré sa posielajú, sú špecifické kontrolné správy, najmä:

PPTP_START_SESSION_REQUEST – vytvorenie spojenia
PPTP_START_SESSION_REPLY – odpoveď na vytvorenie spojenia
PPTP_ECHO_REQUEST – udržiavanie spojenie
PPTP_ECHO_REPLY – odpoveď na udržiavanie spojenia
PPTP_SET_LINK_INFO – konfigurácia spojenia
PPTP_WAN_ERROR_NOTIFY – chyba pri PPP spojení
PPTP_STOP_SESSION_REQUEST – ukončenie spojenia

PPTP_STOP_SESSION_REPLY – odpoveď na ukončenie spojenia

Ako u väčšiny podobných spojení, je tu využívaný tzv. keep-alive mechanizmus, teda neustále vymieňanie si ECHO_REQUEST a ECHO_REPLY správ, aby v každom momente uzly vedeli o dostupnosti druhej strany.

Bezpečnosť

PPTP samo o sebe neposkytuje dôveryhodnosť spojenia ani šifrovanie. Spolieha sa v tom na protokol, ktorý zapuzdruje. Protokol PPP však spravidla býva zabezpečený rôznymi technikami šifrovania aj autentifikácie, takže tieto PPTP prostredníctvom PPP vlastne preberá.
Čo sa šifrovania týka, PPP býva najčastejšie chránené MPPE (Microsoft Point-to-Point encryption, šifrovanie), pričom kľúče sa používajú z autentifikačného procesu, ktorý najčastejšie zabezpečuje MS-CHAP-v2 alebo EAP-TLS. Alebo po novom, napr. vo Windows Vista už je to PEAP (Protected Extensible Authentication Protocol), ktorý na autentifikáciu využíva SSL/TLS tunel.
Treba si uvedomiť, ako cieľový uzol číta prichádzajúci paket z tunela. Najskôr odstráni IP a GRE hlavičku a až nakoniec sa dostane k PPP autentifikácií. Tiež ak by sme si predstavili, že chceme odchytiť paket z PPTP tunela, môžeme odstrániť IP a GRE hlavičky, ale následne nájdeme len kvalitne zašifrované dáta privátnej PPP siete.
Ďalšia silná bezpečnostná schopnosť PPTP „zdedená“ po PPP je filtrovanie paketov. Jednoducho povedané, PPTP server môže mať zvolených klientov, ktorým dovolí komunikovať s LAN sieťou a ostatným prístup zamedzí. Tým pádom sa žiadna neautorizovaná komunikácia nedostane dnu ani von zo siete.

Plusy a mínusy

Na jednej strane PPTP poskytuje rýchle a ľahko konfigurovateľné rozhranie pre VPN, ktoré je navyše široko podporované. Avšak, niektoré nevýhody ho predurčili na to, aby sa nestal uznávaným štandardom. Najmä možná nekompatibilita šifrovacích nástrojov (stačí aby mali komunikujúce uzly rôzne šifrovanie PPP), ale tiež fakt, že kontrolné správy nie sú dostatočne autentifikované. Z tohto hľadiska je prechod na, hoci komplikovanejší a pomalší, ale jednoznačne riešený L2TP žiaduci.

4.2. L2F (Layer 2 Forwarding)

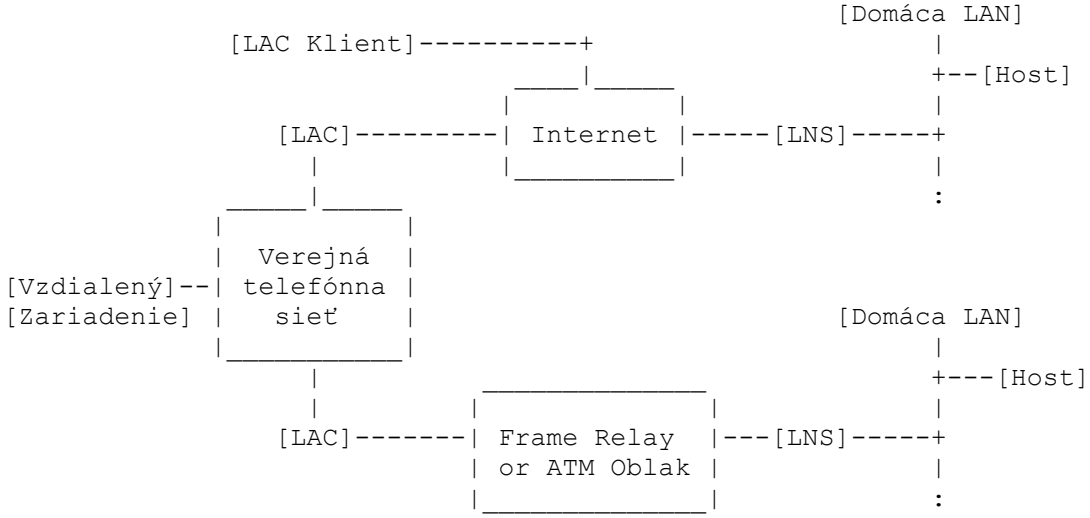
Ku tomuto protokolu len krátko. Vyvinutý Cisco Systems, L2F bol rovnako ako PPTP určený na tunelovanie PPP paketov pre účely VPN. Rovnako ako PPTP sa aj L2F spolieha na šifrovanie a autentifikáciu zapuzdrených protokolov.
Na rozdiel od PPTP využíva inú štruktúru posielaných dát, s použitím vlastnej L2F hlavičky (hoci podobnej ako GRE) a prostredníctvom UDP/IP a iný spôsob nadviazania a kontroly spojenia (nadviazanie pomocou Link Control Protocol LCP, kontrolné správy majú rovnakú hlavičku ako tunelované správy).
Nástupcom L2F je L2TP.

4.3. L2TP

L2TP (Layer 2 Tunneling Protocol) je tunelovací protokol určený na podporu virtuálnych privátnych sietí. Čiže jeho primárna úloha je poskytnúť nástroj na vytvorenie bezpečného spojenia medzi vzdialeným počítačom a prevažne firemnou sieťou, na ktorú potrebuje užívateľ za vzdialeným počítačom prístup. Šifrovanie dát ani dôvernosť však nezabezpečuje, v tom sa spolieha na protokol, ktorý posiela cez svoj tunel.

Protokol L2TP, ktorý je zlúčením a nástupcom Cisco protokolu L2F(Layer 2 Forwarding) a Microsoft PPTP (Point-to-Point Tunneling Protocol), štandardizovaný v 1999 a najnovšia verzia je z 2005 L2TPv3(vylepšená bezpečnosť a encapsulation(obaľovanie), schopnosť prenášať aj iné data linky ako iba PPP), sa správa ako protokol druhej Data-Link vrstvy OSI modelu no v skutočnosti je to až protokol 5. Session (Relačnej) Vrstvy. Presnejšie - umožňuje oddeliť od seba ciele pre Layer 2 (Vrstva 2 – ďalej L2) link, a PPP (Point-to-Point Protokol), ktoré sa zvyčajne nachádzajú na tom istom zariadení. A to tak, že užívateľ získa L2 link s prístupovým koncentrátorom (modem bank a podobne) a tento potom posiela ďalej PPP rámce (frames) serveru pre sieťový prístup (NAS). Toto umožňuje oddeliť spracovanie PPP paketov od ukončenia L2 spojenia. Očividnou výhodou je ukončenie L2 spojenia na koncentrátore, ktorý sa nachádza v blízkosti jeho stanice, miesto na vzdialenom NAS, pričom užívateľ nevidí žiadny rozdiel medzi týmito dvoma.

Topológia



Táto schéma ukazuje typickú L2TP situáciu. Vzdialené zariadenie sa zvyčajne cez verejnú telefónnu sieť pripojí na LAC (L2TP Access Concentrator), ktoré je koncovým bodom L2 spojenia a následne tuneluje PPP rámce ďalej. To sa cez Internet, Frame Relay, alebo ATM oblak, skontaktuje so svojím náprotivkom na strane siete LNS (L2TP Network Server), ktoré je koncovým bodom PPP spojenia, pričom ale v skutočnosti tieto PPP rámce posiela cieľovému zariadeniu na sieti (môže ním byť samo ale nemusí). Pričom vzdialené zariadenie, ktoré ani poriadne nevie, že ide cez L2TP, dostáva priamo potrebné adresy zo súkromnej siete pomocou PPP Network Control protokol negociácie, pričom autentifikáciu a autorizáciu, môže prevádzať správcovská doména siete ako by na ňu vzdialené zariadenie bolo napojené priamo.

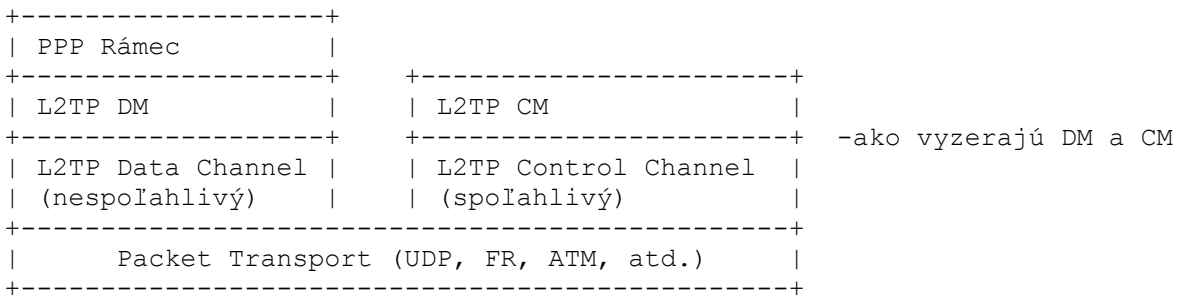
Vysvetlenie niektorých pojmov

Attribute Value Pair (AVP) – Dynamický spôsob zaznamenania dvojice atribút a jeho hodnota. Viaceré AVP dokopy tvoria CM (control messages).
CHAP - Challenge Handshake Authentication Protocol [RFC1994], PPP protokol zabezpečujúci šifrovanú autentifikáciu na výzvu, bez toho aby po linke posielal heslo ako obyčajný text.
Session – *Relácia*, ktorá sa vytvorí medzi LAC a LNS vždy keď je vytvorené PPP spojenie medzi vzdialeným zariadením a LNS. Datagramy súvisiace s PPP spojením sú posielané cez tunel medzi LAC a LNS, toto je jedna relácia. V tunely sa vždy nachádza riadiaca relácia, a okrem nej 0, 1 alebo viacero ďalších. Medzi spojeniami ktoré príma LAC a L2TP reláciami je vzťah 1 : 1.

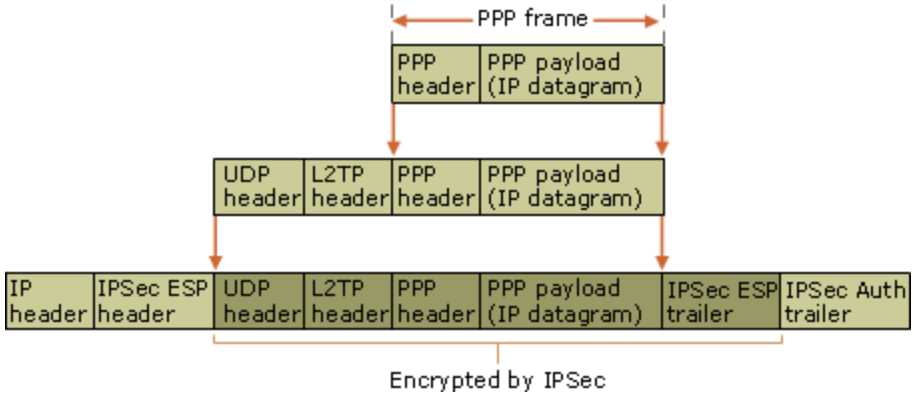
Ako L2TP pracuje.

L2TP využíva dva druhy „správ“ Control Messages (CM, Ovládacie správy) slúžiace na vytvorenie,

spravovanie a ukončenie tunela a relácii v ňom. A Data Messages (DM, Dátové správy) slúžiace na obalenie PPP rámcov, ktoré majú byť tunelované.
Zapuzdrenie dátových správ vyzerá tak, ako na prvom obrázku. Tieto rámce, ak neboli doručené, nie sú posielané znova. Ak je to potrebné, musí to zabezpečovať protokol bežiaci cez L2TP.

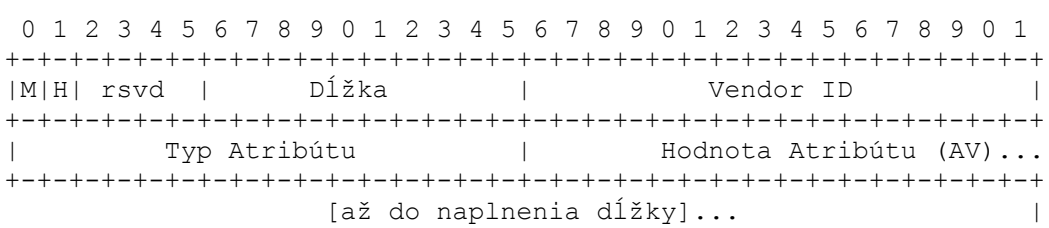


Spoľahlivosť v tomto prípade znamená, či je správa poslaná znova, ak náhodou nedorazí do cieľa. Všimnite si, že L2TP využíva UDP/IP spojenie.



Druhý obrázok ukazuje využitie IPSec, ktoré môže práve túto spoľahlivosť zabezpečiť (a okrem nej šifrovanie, autentifikáciu a neporušenosť dát na vyššej vrstve).

Spoločná pre oba druhy správ (kontrolné a dátové) je L2TP hlavička, ktorá obsahuje pole pre flags, ktoré určujú, či sú prítomné aj voliteľné polia. Flags obsahujú ešte značky pre prioritu (používa sa hlavne pre správy udržujúce spojenie nažive) a verziu L2TP. Identifikátor tunela a relácie. Potom sa v nej nachádzajú voliteľné polia, presnejšie voliteľné pre DM, CM ich má presne určené. Najprv je dĺžka správy. Tá sa nachádza medzi flagmi a tunel ID, CM ju musí obsahovať. Po Session ID nasledujú dve polia pre dve počítadlá, ktoré cez Mod 16 vlastne počítajú počet správ, ktoré sa vymenili, hlavne na zabezpečenie spoľahlivosti. Označenie Ns = číslo správy, ktorá práve prichádza a Nr = číslo nasledujúceho CM (používa sa iba pri nich, DM toto pole ignorujú) a v CM musia byť prítomné. Offset = toto pole určuje na koľkých oktetách za hlavičkou sa nachádzajú prenášané dáta, CM ho nesmú mať.
CM-AVP
Na maximalizáciu rozšíriteľnosti, no zároveň umožniť kompatibilitu, L2TP používa jednotné kódovanie pre typ správy a jej telo, a to AVP = Atribute-Value pair(vysvetlený vyššie).



M – mandatory bit. Určuje ako sa treba zachovať pri prijatí neznámeho AVP. Ak je M nastavené a je prijaté neznáme AVP tak bude ukončená buď relácia, ak AVP súvisí s niektorou, alebo ak správa súvisí s tunelom ako celkom, tak bude ukončený celý tunel spolu so všetkými reláciami, ktoré

v ňom bežia. Ak M nie je nastavený, tak neznáme AVP je ignorované, akoby ani neprišlo.
H – Hidden. Určuje či je hodnota atribútu „skrytá“ alebo nie. Používa sa na to, aby sa citlivé informácie, ako napríklad heslo, neprenášali ako obyčajný text.
Dĺžka – má to isté ako pri hlavičke, určuje počet oktet, ktoré zaberá správa.
Vendor ID, identifikuje producenta AVP. Štandardné AVP tu majú 0 = IETF. Pokiaľ chce mať nejaký výrobca vlastné AVP, dáva sa tu iná hodnota.
Typ Atribútu (AT) – určuje aký druh atribútu to je. Je to číslo zo špecifického zoznamu AVP, ktoré používa konkrétny Vendor.

Hidding = Skrývanie
Ako sme už spomínali, H bit určuje či je hodnota v AVP v „obyčajnom texte“ alebo je to senzitívna hodnota, napríklad heslo alebo prihlasovacie meno. A to len ak existuje nejaké spoločné „tajomstvo“ medzi LAC a LNS (prihlasovanie a podobne). Ak sa má použiť skrývanie, tak v správe musí byť prítomný AVP náhodného vektora pred prvým AVP s H = 1.
Skrývanie prebieha v niekoľkých krokoch. Najprv sa zoberie zo skrývaného AVP dĺžka a hodnota a pridá sa k nej náhodný počet oktet, ktoré zmenia dĺžku výsledného AVP. Následne na vzniknutom polotovare prebehne MD5 hash, presnejšie na reťazci AT a spoločnom tajomstve (polotovar) a náhodnom vektore. Tento vektor je následne pridaný ako povedané pred AVP do správy, a hash je vlúčený do prvých 16 oktet AV pôvodného AVP.

Tunelovanie
Po napojení sa vzdialeného zariadenia na LAC, to pošle cieľovému LNS CM so žiadosťou o vytvorenie tunela a ID tunela, cez ktorý bude posielať správy a zároveň touto správou vytvorí ovládacie spojenie (CC). LNS následne pošle späť CM s ID tunelu, na ktorom bude posielať odpovede. Následne sa cez tunel rovnakým spôsobom dohodne na vytvorení relácie pre protokol, ktorý je tunelovaný, 1 alebo viacero, podľa potreby. Tunelované správy, sú potom posielané cez tieto dva tunely s rovnakou ID relácie.
CC = slúži v prvom rade na autentifikáciu účastníkov, zistenie ich verzie L2TP a ich kapacít čo sa týka spojenia a podobne. Autentifikácia prebieha ak je vyžadovaná cez CHAP (vysvetlený vyššie) a využíva sa tu spoločné tajomstvo, ktoré pozná LAC aj LNS tak ako pri skrývaní.

Pri L2TP tunel môže pokrývať celú PPP reláciu, alebo iba 1 z 2 segmentovej relácie. Takto dostávame 4 druhy Tunelovacích modelov:

1. *voluntary tunnel*
2. *compulsory tunnel — incoming call*
3. *compulsory tunnel — remote dial*
4. *L2TP multi-hop connection*

- V prvom modeli je tunel vytvorený medzi LAC klientom, ktoré posiela L2TP pakety svojmu Internet Service Providerovi (ISP), ktorý ich posúva ďalej LNS. ISP nemusí podporovať L2TP keďže pakety iba posúva ďalej. Tunel pokrýva celú PPP reláciu.
- Pri druhom modely je tunel vytvorený medzi LAC patriaceho ISP a LNS. Podnik môže vzdialenému užívateľovi poskytnúť konto na virtuálnu privátnu sieť, cez ktoré môže pristupovať k jej serveru. Užívateľ teda posiela PPP pakety svojmu ISP (LAC), ktoré ich potom tuneluje patričnému LNS a tunel teda pokrýva iba spojenie medzi ISP(LAC) a LNS.
- Tretí model sa používa napríklad ak podnik potrebuje zo svojej siete napojenie na nejakú vzdialenú kanceláriu. Čiže v tomto prípade tunel inicializuje LNS, pričom tunel končí u ISP (LAC) a inštruuje ISP, aby vytvorilo PPP spojenie k vzdialenému zariadeniu. Vzdialené zariadenie musí mať pevné a známe „číslo“ u ISP.
- Posledný model má tunel rozdelený na dva. Presnejšie v tomto prípade. Vzdialené zariadenie cez LAC vytvorí tunel k L2TP Multi-hop bráne a tá potom vytvorí ďalší tunel k LNS. Správy od jedného k druhému sú potom presmerované k cieľom cez bránu. Toto umožňuje v prvom rade

zmeniť cieľové LNS bez toho, aby LAC muselo vytvárať nový tunel, pretože o to sa postará brána, ktorá je často krát omnoho bližšie k LNS.

5. Tunelovacie protokoly 3. vrstvy

Teraz si povedzme o protokoloch oficiálne tunelujúcich na 3. vrstve. Transportná vrstva je tá, ktorá nám stačí na zachovanie informácie, kam má náš paket smerovať, a preto ju aj najčastejšie tunelujeme. Z toho dôvodu môžeme tieto tunelovacie protokoly nazvať aj všeobecnejšími – využívajú sa aj pri vytvorení VPN, aj pri premostení iných problémov. Za všetko hovorí, že zástupcovia sú tak často využívané protokoly ako GRE a IPSec.

5.1 GRE (Generic Routing Encapsulation)

GRE je univerzálny tunelovací protokol vyvinutý spoločnosťou Cisco Systems. Pôvodne bol zámer používať ho ako prenosný prostriedok cez siete typu IP. Pred pár rokmi, keď sme ešte mohli siete okolo nás považovať za multi-protokolové, TCP/IP bolo teprv na svojom vzostupe a popri ňom sa vyskytovali aj iné riešenia ako AppleTalk suite, či IPX suite, bol takýto prenosný nástroj nesmierne praktický. Dnes už multi-protokolové siete temer vymreli. Novellovské IPX suite nemalo stavbu na to, aby mohlo konkurovať TCP/IP (popritom Novell servery neustále vytláčané z trhu Microsoftovými a Unixovými), spoločnosť Apple dala tiež prednosť TCP/IP a nastalo obdobie jedného internetu.
Preto aj potreba GRE protokolu bola na ústupe. Avšak nástup nového obľúbeného zabezpečujúceho protokolu IPSec znamenal znovuzrodenie GRE, ktorý sa doslova „spolu s ním“ vyniesol opäť do povedomia.

Univerzálnosť

Charakteristické pre GRE je, že sa doň môže zaobaliť prakticky hocičo. Rovnako aj to, že GRE sa zvyčajne zaobalí do prenosného protokolu IP. Toto využíva mnoho ostatných tunelovacích riešení, ktorých je GRE súčasťou (PPTP, L2TP, ...). Práve pre neskrývanú všeobecnosť protokolu sa aj v dokumentácií píše, že často nie je vhodný v situácií špecifického prípadu „X over Y“, kam sa viac hodia konkrétnejšie tunelovacie protokoly. Jeho úlohou je skôr ponúknuť mechanizmus všeobecného významu, ktorý zmierňuje napr. časovú, ale aj routovaciu problematiku tunelovania. Veľmi časté je použitie GRE protokolu (a ešte častejšie v kombinácií s IPSec) napr. pri prenášaní IP cez IP.

Zapuzdrenie

Zapuzdrenie pri GRE protokole funguje názorne, ako sme si už ukázali: payload protokol je zaobalený do GRE protokolu (pridaná GRE hlavička) a následne do delivery protokolu. Výsledný datagram môže mať tvar ako na obrázku (IP cez IP).

Komunikácia

Na vytvorenie tunela treba vytvoriť rozhranie tunela, konfigurovať GRE (lokálnu a cieľovú adresu, a ďalšie dôležité nastavenia ako keepalives, kľúče, checksum) a konfigurovať aj routovacie informácie.
Dôležité pri GRE je, že umožňuje vo svojom tuneli prenášať routovacie protokoly (preto Generic Routing Encapsulation). Zaobalí ich rovnako ako iné payloady. Takže routovanie môže byť okrem statického aj dynamické a cesta sa môže flexibilne meniť (použitím napr. routovacích protokolov

OSPF, RIP a iných). Iné tunelovacie protokoly (napr. IPSec) to nedovoľujú a práve v tomto tkvie veľká využiteľnosť GRE protokolu. Najmä pri virtuálnych privátnych sieťach je veľmi žiaduce, aby si informácia vedela sama nájsť cestu aj v prípade výpadku, cez iný tunel. Preto časté využívanie variácií GRE protokolu aj v iných tunelovacích riešeniach.
Na druhú stranu, celková jednoduchosť protokolu sa prejaví v kontrole a riadení tunela. GRE nemá žiaden stav, ktorý by sa mohol meniť (je stateless). Jeho koncové stanice neuchovávajú žiadne informácie o stave svojich náprotivkov. Začiatku tunela stačí na vysielanie vedieť, že je koniec routovateľný, že je v tuneli a že má fungujúce rozhranie na 1. vrstve (v opačnom prípade vypne svoje GRE rozhranie). To znamená, že lokálne rozhranie nezistí, či je druhá strana dobre nakonfigurovaná, či sa dáta cestou nestrácajú, alebo či funguje GRE rozhranie na druhej strane (v týchto prípadoch posiela správy nikam).
Posledné dva problémy sa ale dajú vyriešiť nakonfigurovaním a použitím tzv. GRE keepalives (technológiou podobnou ako zvyčajne využívajú nižšie vrstvy). GRE rozhrania si medzi sebou po tuneli posielajú keepalive správy spôsobom – keď dostanú odpoveď, je všetko v poriadku. Counter sa zvyšuje každým poslaním keepalive a vynuluje sa len ak dostane odpoveď. Pokiaľ ale odpoveď neprichádza a counter prekročí istú hranicu, GRE rozhranie sa vypne. Zvláštnosťou je, že stačí aby len jedna strana mala keepalives nakonfigurované. Tajomstvo je v tom, že keepalive správa je 2-krát zaobalený payload, z ktorého druhá strana len odstráni vrchný obal a zvyšok odošle naspäť.
V prípade, že sa GRE rozhranie vypne, okolité routre z tunela, ktoré naň poukazovali, si vymažú daný uzol z routovacej tabuľky a ak sa dá, komunikácia pokračuje iným tunelom.

Bezpečnosť

GRE ponúka len veľmi slabé bezpečnostné opatrenia. Pole „key“ v GRE hlavičke môže byť využité na autentifikáciu, ale je nešifrované a ľahko prečítateľné pre každého. Preto sa skôr používa na rozlíšenie prípadných viacerých tunelov medzi koncovými bodmi, ako na nejakú bezpečnosť.
Práve preto sa GRE spája s tunelovacím protokolom IPSec (tunel GRE cez tunel IPSec). Univerzálnosť GRE sa vynikajúco dopĺňa s bezpečnosťou IPSec. O tejto kombinácií napíšeme viac pri IPSec protokole.

Takže suma sumárom, využitie GRE: Ako nástroj pre premosťovanie ľubovoľných protokolov. Vo VPN sieťach. A často práve tam, ale nielen tam, sa využíva v spolupráci s IPSec na vytvorenie bezpečného spojenia.

5.2. IPSec (Internet Protocol Security)

IPSec definovaný IETF (Internet Engineering Task Force) vznikol ako uspokojivé riešenie neustále väčšej potreby mať bezpečné súkromné siete. Je to vlastne množina protokolov zabezpečujúca bezpečnosť, neporušenosť, autentifikáciu aj správne poradie dát. Zároveň poskytuje možnosť vytvoriť tunel (zapuzdriť payload a vytvoriť novú IP hlavičku). Táto možnosť pridania „vrstvy bezpečnosti“ vo VPN sieti sa stala veľmi obľúbenou a IPSec je dnes štandard používaný v kombinácií s ostatnými tunelovacími protokolmi.
Keďže pracuje podobne ako GRE na 3. vrstve, obaľuje všetko, čo je nad ním. V tom má plus pred ostatnými šifrovaniami, ktoré väčšinou pracujú na vyšších vrstvách (TLS, SSL, SSH a iné). Viac toho šifruje a zároveň nemusí byť zahrnutý v dizajne aplikácie. 2. vrstvu niekedy treba tiež šifrovať, ale to je možné len v súkromnej sieti ako je LAN. IPSec chcelo byť použiteľné viac všeobecne a hlavne na internete, kde sú linky na 2. vrstve verejné, preto je práca na 3. vrstve logická. Keďže zapuzdruje prevažne IP, je IPSec považovaný právom za oficiálneho „IP bodyguarda“.

Funkcie

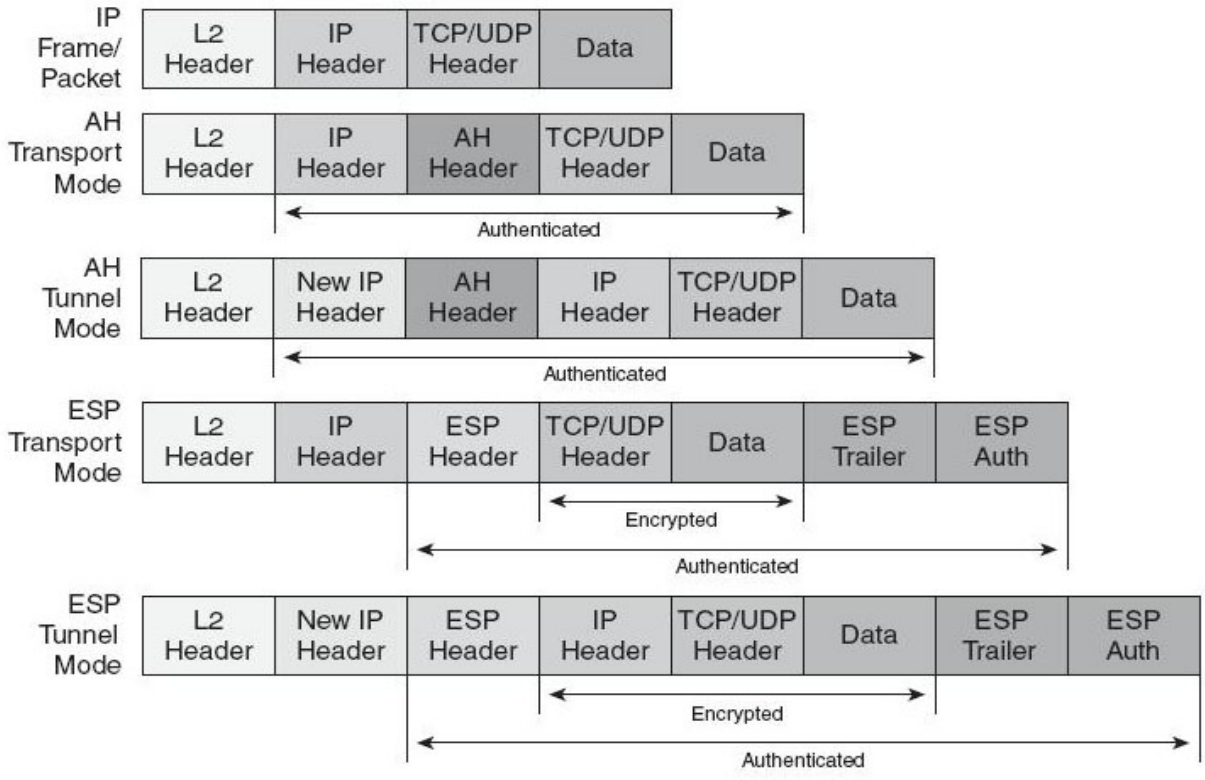
Bezpečnosť dát (data confidentality). IPSec ju môže zariadiť šifrovaním so spoľahlivým generovaním a bezpečnou výmenou kľúčov.
Neporušenosť dát (data integrity). Typicky je zabezpečená hashovacími algoritmami.
Autentifikácia zdroja (data origin authentication). Spoľahlivé overenie identity druhej strany.
Poradie dát (anti-reply). Detekovanie oneskorených paketov a zamedzenie duplikátov.
Z týchto funkcií sú data confidentality a anti-reply uvádzané ako voliteľné.

IPSec protokoly

Spomínané funkcie sú zabezpečené IPSec protokolmi. Dôležité je poznamenať, že tieto protokoly sú založené na medzinárodne uznávaných otvorených algoritmoch. IPSec nešpecifikuje nijaké nové riešenie.
IKE (Internet Key Exchange): Využívaný na výmenu kľúčov pri symetrickom šifrovaní, ale aj na synchronizáciu nastavení koncových bodov.
ESP (Encapsulating Security Payload): Hlavný nástroj na šifrovanie. Dajú sa ním zabezpečiť všetky spomínané funkcie IPSec.
AH (Authentication Header): Využíva hashovacie algoritmy a spravuje neporušenosť dát a autentifikáciu. Pretože tieto autentifikačné dáta nechráni, málokedy sa využíva osamote.

Zapuzdrenie

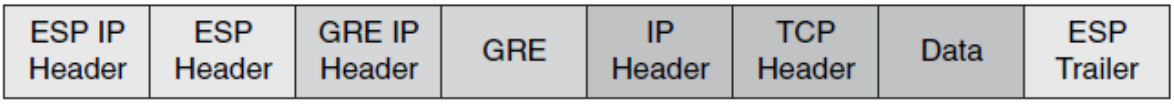
Pretože IPSec sa skladá z viacerých (voliteľných) súčastí, môže aj v praxi fungovať vo viacerých módoch. Treba mať na pamäti, že IPSec môže fungovať aj v nie tunelovacom (transport) móde, kedy len zabezpečuje IP paket bez druhej IP hlavičky. Dva uvedené varianty – použitie ESP a použitie AH môžu byť aj kombinované (hoci AH nepridáva ESP význačnú funkcionalitu).



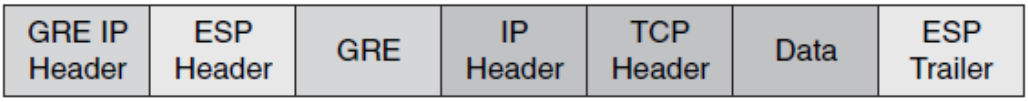
GRE cez IPSec

Neschopnosť IPSec zapuzdrovať routovacie protokoly sa rieši elegantne – zapuzdrením GRE do IPSec. Bezpečný tunel IPSec je teda navrchu a univerzálny a dobre routovaný GRE tunel je vnútri. Opačný prístup nedáva zmysel. Avšak, dá sa využiť fakt, že IPSec funguje aj v netunelovom móde. V tom prípade je navrchu GRE IP hlavička, ide vlastne o GRE tunel, ktorý je akoby kúzlom chránený IPSec-om, a ušetrí sa tým navyše jedno zapuzdrenie (keďže IP hlavičky vytvorené GRE a IPSec väčšinou bývajú rovnaké).

Tunnel Mode

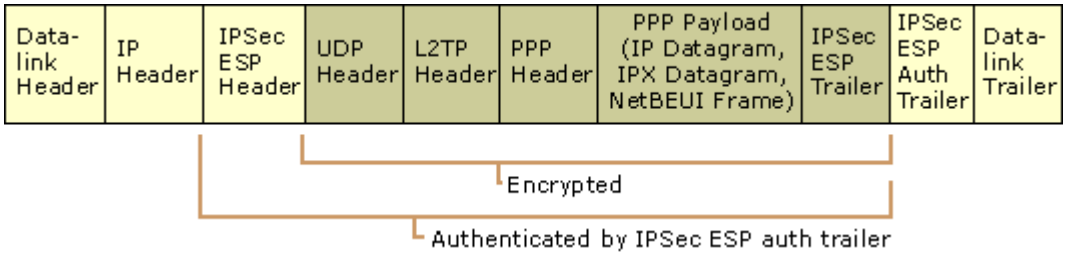


Transport Mode



L2TP cez IPSec

Veľmi využívané je aj spojenie L2TP a IPSec. Dva najvyužívanejšie štandardy v oblasti VPN a bezpečnosti dávajú dokopy riešenie, kde je dobre zabezpečené prakticky všetko. V praxi, keď sa povie L2TP tak sa často myslí L2TP zabezpečené IPSec. IPSec sa tu využíva tiež v transportnom móde.



6. Tunelovanie a IPv6

Táto časť nášho projektu zameraného na tunneling sa sústredí najmä na jednu oblasť využitia tunnelingu a to konkrétne IP tunneling a prečo je relevantný pre prechod z technológie IPv4 na IPv6.

Uvedieme dôvody, prečo sa to využíva práve v tejto oblasti, teda zasiahneme do problematiky IPv6 vs. IPv4. Avšak len mierne, pretože objasnenie architektúry, funkčnosti a rozdielom týchto dvoch protokol nie je cieľom nášho referátu, avšak istá vedomosť sa vyžaduje pre pochopenie, keďže časť nášho referátu s nimi bude úzko súvisieť.

Nosná časť nasledujúcej pasáže nášho referátu sa bude venovať IP tunnelingu, najprv niečo o princípe vo všeobecnosti na zopakovanie a potom bližšie o protokoloch 6to4, 6in4 – čo sú zač, ako fungujú a na čo sa využívajú. Keď už budeme teoreticky nabití vedomosťami, tak na konci si niečo z toho vyskúšame.

IPv6 a IPv4

Prvá verejne používaná verzia Internet Protokolu (IPv4) ponúka adresovaciu kapacitu veľkosti približne 4 miliardy adries (2^{32}). Toto sa považovalo za dostatočné v ranných etapách vývoja internetu, kedy asi nikto ani vo svojich tajných snoch neočakával súčasný enormný rast.

Začiatkom 90-tych rokov bola vyvinutá nová kategorizácia IP adries za účelom lepšej alokácie, tzv. „classless network“ avšak aj toto dizajnové vylepšenie sa veľmi rýchlo ukázalo ako nedostačujúce a bolo jasné, že budú potrebné ďalšie zmeny v infraštruktúre. Koncom roku 1992 sa začalo pracovať na novej generácií IP protokolu, priebežne nazvanom IPng. Postupne sa z IPng vykryštalizoval IPv6 a v roku 1996 boli položené jeho základy.

IPv4 používa 32-bitové adresovanie, IPv6 na rozdiel od toho ponúka 128-bitové adresovanie a teda 2^{128} možných adries, čo je pri terajšej populácií Zeme skoro jedna adresa na jedného obyvateľa. Sú medzi nimi ešte ďalšie rozdiely, ale toto je pre bežného používateľa ten najhlavnejší.

Základy IPv6 boli síce položené už v roku ’96, ale protokol je stále len v zárodočnej fáze, čo sa nasadenia týka. Podľa nedávneho výskumu Google zaberá IPv6 menej než 1% z internetovej prepravy v akejkoľvek krajine na svete. Naopak IPv4 je momentálne silne dominantná. Preto sa predpokladá, že IPv4 bude ešte dlho podporovaná spolu s IPv6. Teda kľúčom k úspešnému prechodu na IPv6 je kompatibilita s už existujúcou základňou IPv4 hostiteľov a routerov. Dokým (ak?) sa postupne implementuje IPv6, existujúca IPv4 infraštruktúra musí zostať funkčná, teda musia byť tieto protokoly schopné navzájom spolu komunikovať. Avšak uzly IPv4 nie sú schopné priamo komunikovať s IPv6 uzlami, je nutná asistencia prevodových mechanizmov – jedným z nich je **tunneling.**

Tunneling

Momentálne, na spojenie s IPv6 internetom, izolovaný hostiteľ alebo sieť musí použiť IPv4 infraštruktúru na prenos IPv6 packetov. Toto sa robí pomocou techniky známej ako tunneling, ktorá pozostáva z obalenia (/enkapsulácie) IPv6 packetov do IPv4, vlastne sa tak používa IPv4 ako link layer pre IPv6. Zaobalené pakety cestujú po IPv4 internete dokým nedosiahnu svoju destináciu. Router alebo hostiteľ si je vedomý, že ide o zaobalený packet a tak si ho správne spracuje.
Priame obalenie IPv6 diagramov do IPv4 packetov je zahrnuté v protokole IP číslo 41. IPv6 môže byť tiež zaobalená do UDP packetov, čo sa používa v prípadoch keď router alebo NAT zariadenie blokuje protokol 41. Existujú ešte iné spôsoby obalenia, jedno z nich je použité napríklad

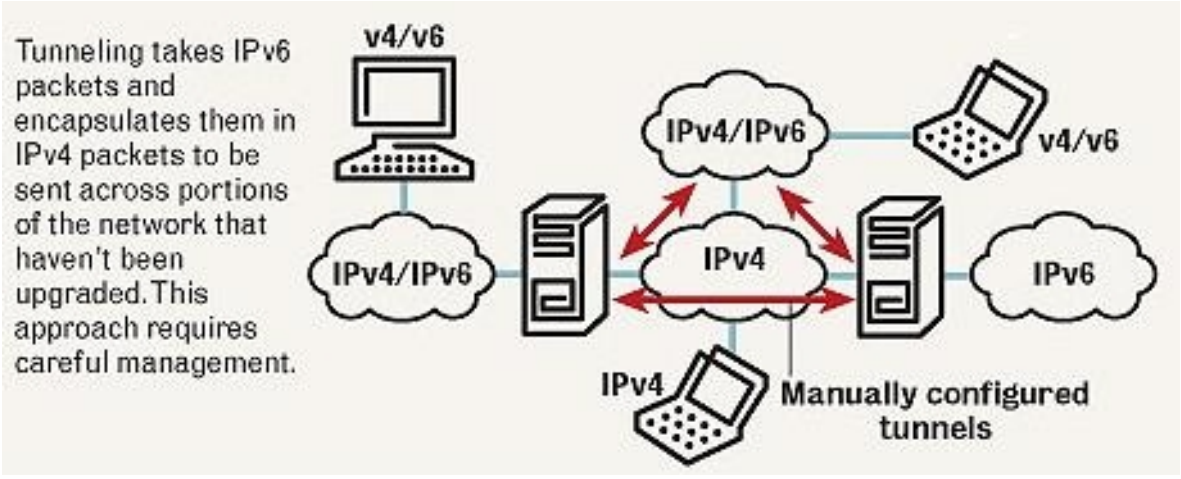
v populárne používanom protokole GRE.

Tunelovacie techniky sú väčšinou klasifikované podľa mechanizmu, ktorým vstupný uzol tunelu (encapsulating node) rozhodne o adrese uzla na konci tunelu. Vstupný uzol vytvorí obaľovaciu IPv4 hlavičku a vysiela obalený paket. Koncový uzol (decapsulating node) získava zabalený paket, odstráni IPv4 hlavičku, aktualizuje IPv6 hlavičku a spracuje obdržaný IPv6 paket.

Existujú dve tunelovacie techniky pre IPv6:

a) Automatický tunneling - routovacia infraštruktúra automaticky rozhoduje o koncových uzloch. RFC 3056 odporúča 6to4 tunneling pre túto techniku, ten používal obalenie protokolu 41. Koncové body sú determinované pomocou známej IPv4 anycast adresy na vzdialenej strane a uložením informácie o IPv4 adrese do IPv6 adresy na lokálnej strane. Používanie 6to4 je dnes veľmi rozšírené.
Teredo je automatická tunelovacia technika, používajúca UDP enkapsuláciu. Jej používanie nie je dnes veľmi rozšírené, ale experimentálna verzia Tereda je súčasťou Windows XP SP2. 6to4 a Teredo sú štandardne povolené vo Windows Vista. Väčšina Unix systémov natívne podporuje iba 6to4.

b) Manuálne nastavený tunneling (6in4) – pri tejto technike sú natvrdo nastavené koncové body tunela, buď administrátorom, configuračným mechanizmom operačného systému alebo automatickým servisom známym pod názvom tunnel broker. Nastavený tunneling je obvykle viac deterministický a jednoduchšie zbavený chýb ako automatický tunneling, a preto je odporúčaný pre veľké siete. Nastavený tunneling používa IP protokol číslo 41.



Ilustračný príklad manuálneho tunela

Problémy tunnelingu

Existujú obavy ohľadom bezpečnosti tunelovacích techník. Napríklad pri automatickom tunnelingu nie je jednoduché neustále sledovať kto vlastne komunikuje cez tunely a nepozná sa koncový bod tunelu. Môže sa stať, že router komunikuje s neautorizovanými routermi. Tiež je možné posielať podvrhnutú prepravu smerom ku koncovému bodu a router potom obržiava prepravu falošne vloženú do tunela.

Tunely musia byť neustále monitorované a upravované. Akonáhle bude prechod na IPv6 kompletný budú musieť byť odstránené. Opravovanie chýb v prostredí plnom tunelov je náročné, preto sú tunely len prechodnou technikou.

IP tunneling

Pripomeňme si základy IP tunnelingu, ktorý sme si spomínali aj pri zriadení VPN sietí. Je to metóda prepojenia dvoch IP sietí, ktoré nemajú natívnu routovaciu cestu medzi sebou. Spojenie sa uskutočnuje cez komunikačný kanál (IP tunel), ktorý používa enkapsulačné technológie.

Použitie

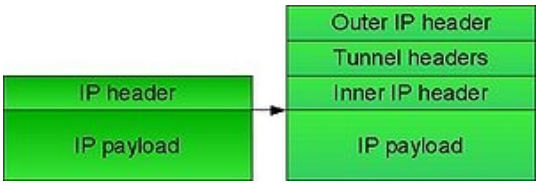
IP tunely sú často používané spolu s IPsec protokolom na vytvorenie virtuálnej privátnej siete medzi dvomi alebo viacerými privátnymi sieťami cez verejnú sieť ako napríklad internet. Ďalšie populárne využitie je spojenie ostrovov IPv6 implementácií cez IPv4 internet.

Princíp

V IP tunnelingu, každý IP paket, s informáciou o adrese prvotnej a koncovej IP siete, je obalený do ďalšieho paketu vo formáte, ktorý je natívny sieti, ktorá paket prenáša.

Na hraniciach medzi prvotnou sieťou a prenosnou sieťou, ako aj medzi prenosnou sieťou a cieľovou sieťou, sú použité brány (gateway) na vytvorenie koncových bodov IP tunela cez prenosnú sieť. Takže, koncové body IP tunnelu sa stanú natívnymi IP routermi, ktoré vytvárajú štandardnú IP cestu medzi prvotnou a cieľovou sieťou. Pakety prechádzajúce cez tieto koncové body, ktoré prichádzajú z prenosnej siete sú zbavené svojho prenosného obalu (hlavičky a pätičky používané v tunelovacom protokole) a teda sú prekonvertované do natívneho IP formátu a vložené do IP zásobníka koncových bodov. Ak bola počas prenosu použité nejaká iná protokolová enkapsulácia (ako napr. IPsec, TLS/SSL), tak bude taktiež odstránená.

IP tunneling často s prehľadom preskakuje jednoduché pravidlá, pretože špecifický princíp a adresovanie originálnych datagramov sú schované. Preto ak chceme blokovať IP tunely je na to nutný špecializovaný software.



IP tunneling enkapsulácia

Protokol 6to4

Ide o systém, ktorý umožňuje IPv6 paketom, aby boli prenášané cez IPv4 sieť (všeobecne cez IPv4 internet) bez nutnosti nastavovať explicitné tunely. Existujú tiež aktívne routovacie konvencie, ktoré dovoľujú 6to4 hostiteľom komunikovať s hostiteľmi na IPv6 internete. Typicky je tento systém používaný keď sa chce koncová stránka alebo koncový používateľ pripojiť na IPv6 internet, použitím svojho existujúceho IPv4 pripojenia.

Keďže IPv6 nevyžaduje nastavenie alebo podporu na hocijakom priľahlom sieťovom zariadení

relatívnom k hostiteľovi, 6to4 je obzvlášť relevantné počas prvých fáz spustenia plnej, natívnej IPv6 prepojiteľnosti. Avšak, ide len o prechodný mechanizmus a nie je jeho cieľom byť používaný permanentne.

6to4 môže byť použitý individuálnym hostiteľom alebo lokálnou IPv6 sieťou. Ak je používaná individuálnym hostiteľom, tak ten musí mať IPv4 pripojenie a globálnu IPv4 adresu. Hostiteľ je zodpovedný za enkapsuláciu odchádzajúcich IPv6 paketov a dekapsuláciu prichádzajúcich 6to4 paketov. Mnohé hostiteľské operačné systémy implementujú túto enkapsuláciu/dekapsuláciu cez 6to4 pseudo-interface.

Ak 6to4 používa lokálna sieť, celá sieť potrebuje iba jednu IPv4 adresu. Vnútri siete hostitelia spoznajú svoju IPv6 adresu a routovanie pomocou normálnych protokolov, presne ako v natívnej IPv6 sieti.

Ako to funguje?

6to4 vykonáva tri funkcie:

1. Pridelí blok adresného priestoru IPv6 akémukoľvek hostiteľovi alebo sieti, ktorá má globálnu IPv4 adresu.
2. Enkapsuluje IPv6 pakety do IPv4 paketov na prenos cez IPv4 sieť pomocou protokolu 6in4.
3. Routuje prepravu medzi 6to4 a natívnymi IPv6 sieťami.

Pridelenie adresného bloku

Pre akúkoľvek 32-bitovú globálnu IPv4 adresu, ktorá je pridelená hostiteľovi, môže byť skonštruovaný 48-bitový 6to4 IPv6 prefix určený na použitie pre daného hostiteľa (a sieť, ktorá sa za ním skrýva) pridaním hexadecimálneho reťazca „2002“ na začiatok IPv4 adresy. IPv4 adresy používajú notáciu dot-decimal (desatinné čísla oddelené bodkou), IPv6 adresy však používajú hexadecimálnu notáciu. Takže napríklad pre globálnu IPv4 adresu 192.0.2.42 je korešpondujúci prefix 2002:c000:022a::/48. Prefix dĺžky 48 bitov necháva miesto na 16-bitové pole pre podsieť a 64 bitov pre hostiteľskú adresu vnútri podsiete.

Akákoľvek IPv6 adresa, ktorá sa začína na 2002::/16 je teda automaticky považovaná za 6to4 adresu, na rozdiel od natívnej IPv6 adresy, ktorá daný prefix nepoužíva.

Enkapsulácia a prenos

6to4 vsadí IPv6 paket do „užitočnej“ (payload) časti IPv4 paketu s typom protokolu nastaveným na 41. Na poslanie IPv6 paketu cez IPv4 sieť na 6to4 konečnú adresu, je IPv4 hlavička s protokolovým

typom 41 pripojená na začiatok IPv6 paketu. Konečná IPv4 adresa pre túto hlavičku je odvodená z konečnej IPv6 adresy vnútorného paketu a to tak, že sa extrahuje 32 bitov, ktoré okamžite nasledujú IPv6 adresný prefix 2002::. IPv4 adresa zdroja v pripojenej paket hlavičke je IPv4 adresa hostiteľa alebo routera, ktorý posiela daný paket cez IPv4. Výsledný IPv4 paket je potom routovaný na IPv4 konečnú adresu presne tak isto ako každý iný IPv4 paket.

Routing medzi 6to4 a natívnou IPv6

Aby si mohli hostitelia a siete používajúce 6to4 adresy vymieňať prepravu s hostiteľmi používajúcimi natívne IPv6 adresy, boli zriadené takzvané „relay routers“. Relay router je spojený s IPv4 sieťou a aj s IPv6 sieťou. Payload časť 6to4 packetu prichádzajúceho na IPv4 interface bude routovaná do IPv6 siete, zatiaľ čo packety prichádzajúce na IPv6 interface s prefixom 2002::/16 cieľovej adresy budú enkapsulované a presmerované cez IPv4 sieť.

Aby mohol 6to4 router komunikovať s natívnym IPv6 internetom, musí mať nastavenú štandardnú bránu na 6to4 adresu, ktorá obsahuje IPv4 adresu nejakého 6to4 relay routera. Aby toto nemuseli používatelia nastavovať manuálne, 6to4 relay anycast adresa 192.88.99.1 (čo obalené do 6to4 je 2002:c058:6301), bola alokovaná za cieľom posielania packetov pre relay router. Kvôli routovacím dôvodom bolo celé 192.88.99.0/24 alokované pre cesty smerované na 6to4 relay routre, ktoré používajú anycast IP. Provajderi ochotný zabezpečiť 6to4 pre svojich klientov alebo partnerov by mali propagovať anycast prefix ako každý iný IP prefix, a routovať daný prefix do svojho 6to4 relay routera.

Packety z IPv6 internetu smerujúce do 6to4 systému musia byť poslané do 6to4 relay routra pomocou bežných IPv6 routovacích metód. Špecifikácia stanovuje, že relay routre môžu propagovať iba 2002::/16 a žiadne pododdiely, aby sa predišlo tomu, že IPv4 cesty naplnia routovacie tabuľky IPv6 routrov. Odtiaľto môžu byť potom cez IPv4 do svojej destinácie.

Bezpečnosť

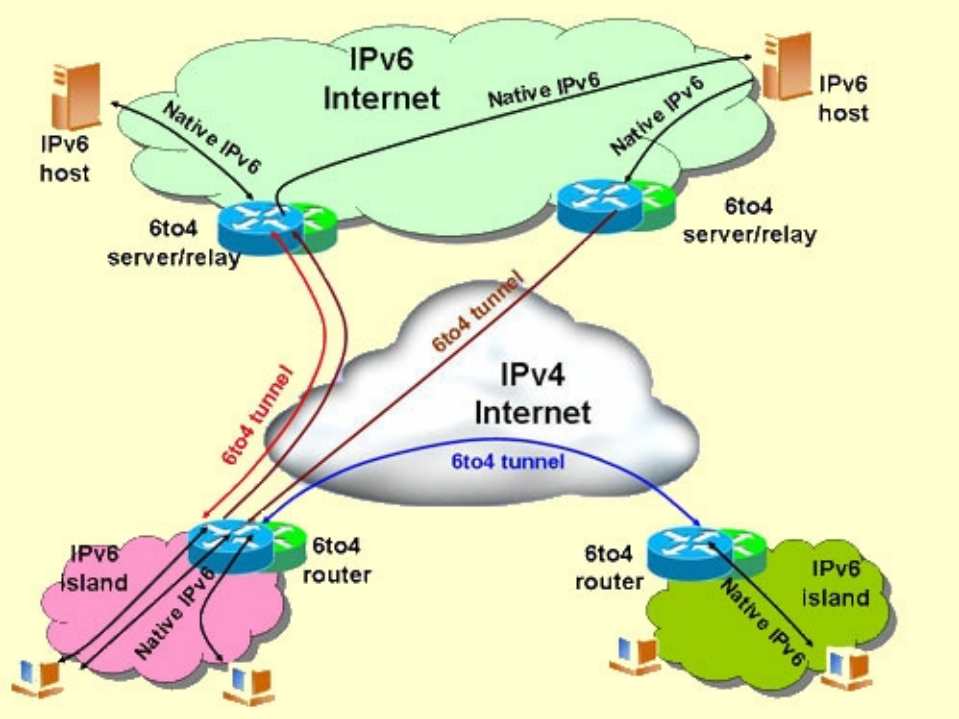
Ako už bolo spomenuté tunneling má aj svoje temné stránky. Preto by pri 6to4 tunelovaní treba zabezpečiť.

a) Jeden alebo obaja, z prvotnej a cieľovej adresy obaleného paketu, je rozmedzí 6to4 IPv6 prefixu 2002::/16.
b) Ak je prvotná adresa 6to4 IPv6 adresa, tak korešpondujúca IPv4 adresa 6to4 routra sa zhoduje s adresou zdroja v IPv4 hlavičke obaleného paketu.
c) Podobne, ak je cieľová adresa IPv6 adresa, tak korešpondujúca IPv4 adresa 6to4 routra sa zhoduje v IPv4 hlavičke.
d) Akákoľvek uložená 6to4 router adresa je globálny unicast.

Protokol 6in4

6in4 používa tunneling techniky enkapsulácie IPv6 prepravy cez manuálne nastavené IPv4 tunely (tak ako sú definované v RFC 4213 http://tools.ietf.org/html/rfc4213). IPv4 tunelová preprava používa IP protokol číslo 41 v hlavičke. Toto číslo protokolu je špeciálne rezervované pre IPv6-in-IPv4. 6in4 tunneling je často nazývaný ako „proto-41 static“, pretože konečné body sú nastavené staticky.

Pri tejto technike je IPv4 hlavička paketu okamžite nasledovaná celým IPv6 paketom. Toto samozrejme minimalizuje veľkosť paketu o 20 bytov, ak má napr. Ethernet nastavené MTU na 1500, tak maximálna veľkosť nefragmentovaného IPv6 paketu je 1480 bytov.



Ako sa dostať na IPv6 internet s naším IPv4 pripojením?

Keď už teda vieme ako to zhruba funguje a ako by sme sa mohli vedieť dostať na náš, tak nám nič nebráni v tom si to nevyskúšať. Spomenuli sme viacero, vyskúšajme tri základné – využitím servisu „tunnel broker“, využitím pseudo-interface operačného systému.

a) Tunnel broker – je servis, ktorý poskytuje sieťový tunel. Niekoľko ich je vypísaných na wikipedii - http://en.wikipedia.org/wiki/List_of_IPv6_tunnel_brokers . Ja som si vybral **Hexago/Go6** providera.

Ako na to?

1. Najprv ideme na stránku go6.net, keď už sme tam tak klikneme na tlačidlo, ktoré sa nachádza vpravo hore - "Free IPv6 Connectivity with Freenet6".

2.Teraz sa treba zaregistrovať, po vyplnení údajov prejdeme do download sekcie a vyberieme si software podľa OS, na ktorom momentálne fungujeme. (my si vyberáme installer pre Windows XP).

3. Stiahneme a nainštalujeme si náš nový software. (Windows si môže sťažovať na inštaláciu niektorého nepodpísaného drivera, ak Vás nepustí ďalej treba editovať Local Computer Policy pomocou gpedit.msc).

4. Už stačí len nastaviť Gateway6 klienta. Gateway6 adresu nastavíme napríklad na broker.freenet6.net a do prihlasovacích údajov dáme svoje registračné údaje.

5. Stlačíme „Connect“ , chvíľku počkáme a sme pripojení!

6. Testovanie. Vyskúšajme ping-núť nejakú ipv6 adresu. Úplne podobne ako obyčajný ping funguje aj ping6, takže idem do command line a zadávam „ping6 ipv6.google.com“

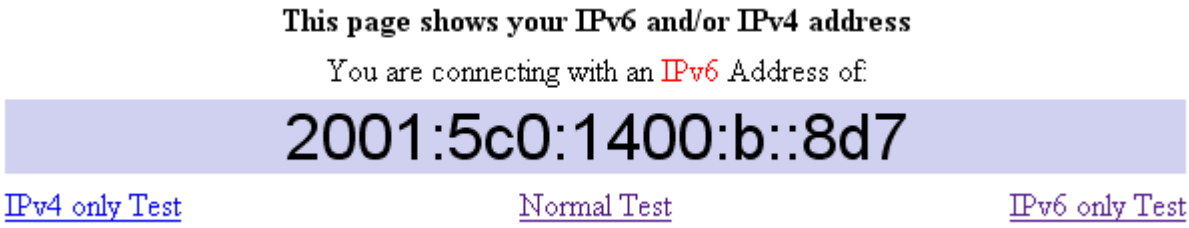
```
C:\>ping6 ipv6.google.com

Testuje sa dostupnosť ipv6.l.google.com [2001:4860:a003::68]
od 2001:5c0:1400:b::8d7 s 32 bajtmi údajov:

Odpoveď od 2001:4860:a003::68: počet bajtov = 32 čas = 59 ms
Odpoveď od 2001:4860:a003::68: počet bajtov = 32 čas = 57 ms
Odpoveď od 2001:4860:a003::68: počet bajtov = 32 čas = 57 ms
Odpoveď od 2001:4860:a003::68: počet bajtov = 32 čas = 57 ms

Štatistika testovania dostupnosti 2001:4860:a003::68:
    Pakety: odoslané = 4, prijaté = 4, stratené = 0 (straty: 0 ),
Približné časy výmeny údajov v milisekundách:
    minimum = 57 ms, maximum = 59 ms, priemer = 57 ms
```

Odozva je v poriadku, takže sme úspešne pripojení na IPv6 internet. Môžem si dokonca zistiť svoju IPv6 adresu na stránke http://whatismyv6.com/



Ako vidíme servis go6 funguje dobre.

b) V predchádzajúcom prípade za nás software urobil, takmer všetko, skúsme si to trošku urobiť náročnejším a 6to4 hostiteľa si nastavíme sami, pomocou pseudo-interfejsu, ktorý nám ponúka Windows XP. Návod ako postupovať nájdeme na http://www.tunnelbroker.net/forums/index.php?topic=20.0 .

Ako na to?

1) V prvom rade si treba nainštalovať IPv6. Spustíme si príkazový riadok a napíšeme doň „ipv6 install“. Tu by sme mohli aj skončiť, pretože ak máte Service Pack >= 2 tak by malo byť všetko automaticky prednastavené. Ak je všetko v poriadku s vaším providerom, teda neexistuje problém s routovaním hlavičiek s protokolom 41, tak by to už malo fungovať. Ale skúsme si ich nastaviť korektne manuálne. (Poznámka nám práve toto nefungovalo, takže toto bolo uskutočnené inde.)

2) Pokračujeme príkazom „**ipv6 rtu 2002::/16 2**“.

Preklad: rtu - routing table update
2002::/16 - prefix špecifický pre 6to4
2 – číslo interfejsu, ktorý má zaslaný paket spracovať, 2 je v mojom prípade automatický pseudo-interface, ktorý sa stará o nastavené tunely, čiže aj o 6to4.

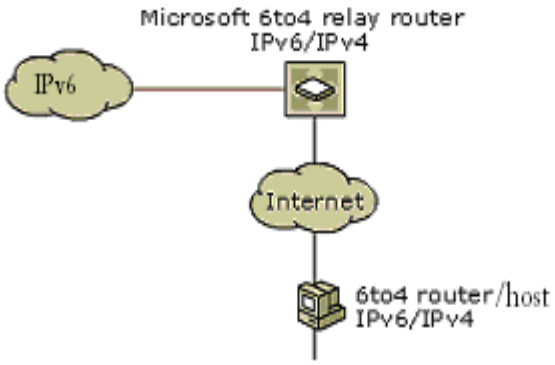
3) Ďalej potrebujeme nastaviť našu 6to4 adresu. To spravíme príkazom „**ipv6 adu 2/2002:3e00:93cd::62.0.147.205**“.

Čo znamená:
adu – address update
2 – číslo interfejsu
2002 – prefix
3e00:93cd – hexidecimálny preklad IPv4 adresy hostiteľa 62.0.147.205, a za :: samotná IPv4 adresa .

4) Ďalej treba zabezpečiť aby všetky naše pakety určené pre IPv6 internet prišli do relay routra. Anycastová adresa, ktorá sa na toto používa je 192.88.99.1. Príkaz „**ipv6 rtu ::/0 2/::192.88.99.1 pub life 1800**“.

rtu – routing table update.
::/0 – štandartná cesta.
2/::192.88.99.1 – dalšia zastávka pre 2. interfejs
pub – publikovaná cesta (nedôležité)
life 1800 – time to live (TTL)

1. Teraz by už malo byť všetko správne nastavené, architektonicky to vyzerá zhruba takto.



Test tunela:

```
Pinging 6bone.net [3ffe:b00:c18:1::10]
from 2002:3e00:93cd::3e00:93cd with 32 bytes of data:

Reply from 3ffe:b00:c18:1::10: bytes=32 time=524ms
Reply from 3ffe:b00:c18:1::10: bytes=32 time=541ms
Reply from 3ffe:b00:c18:1::10: bytes=32 time=561ms
Reply from 3ffe:b00:c18:1::10: bytes=32 time=581ms

Ping statistics for 3ffe:b00:c18:1::10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 524ms, Maximum = 581ms, Average = 551ms
```

Funguje taktiež správne.

IPv6 a tunelovanie - záver

Predstavili sme si mechanizmy, ktoré slúžia na prepojenie dvoch mechanizmov – jedného momentálne rozšíreného – IPv4 – a jedného plánovaného pre budúcnosť – IPv6. Keďže má jeden časom celkom nahradiť druhý ide len o dočasné techniky, ale dokým nebola migrácia úplne dokončená sú veľmi podstatné a užitočné.

7. SSH Tunneling

Pozrime sa teraz na tunelovanie iným pohľadom – ako na užitočný nástroj využívaný jednotlivcom na prácu so svojimi dátami (ktoré si často potrebuje sťahovať a uploadovať z nejakého servera). Jedna z najpopulárnejších tunelovacích metód je tunelovanie pomocou známeho terminálového protokolu SSH.

SSH (Secure Shell)

Secure Shell je sieťový protokol slúžiaci na vytvorenie bezpečnej komunikačnej linky, medzi nami a vzdialeným zariadením, najvyužívanejší pritom je na bezpečné prihlasovanie sa a autentifikáciu.
Bol vyvinutý ako náhrada Telnetu a iných nezabezpečených spojovacích protokolov. Zabezpečuje dôvernosť a integritu posielaných dát, aj keď budú prenášane nezabezpečeným prostredím ako Internet.
SSH vznikol na Helsinskej Technickej Univerzite v r. 1995 ako freeware, neskôr jeho tvorca založil SSH Comunication Security a SSH sa postupne čoraz viac stával plateným. V 96-tom bol vytvorený SSH-2, ktorý vylepšil bezpečnostné prvky pôvodnej verzie a v spojení so začatím štandardizácie, použitím prvkov ako Diffie-Hellmanova výmena kľúčov, silná kontrola integrity a ďalšie, prispeli k výraznému zvýšeniu bezpečnosti oproti pôvodnej verzii a dnes je odporúčané používať len tú.
V roku 1999 bol vytvorený OpenSSH proti vtedy už platenému komerčnému SSH, pričom ako základ bola posledná voľná verzia SSH. OpenSSH je veľmi výrazne podobný SSH v tom nastavovaní a vo väčšine používaní a je výrazne rozšírený, hlavne ako súčasť voľných OS, napríklad rôzne Linuxy a podobne.
SSH sa používa na prihlásenie sa do vzdialeného systému, vykonávanie príkazov na tomto zariadení, bezpečné kopírovanie súborov zo vzdialeného zariadenia, alebo servera, SFTP ako bezpečnejšia alternatíva k FTP, tunelovanie, a mnohé ďalšie.

Ako funguje

Základom je dvojica SSH servera a klienta. Server sedí na vzdialenom zariadení a zvyčajne na porte 22 počúva či niekto nechce spojenie. Klient sa nachádza u užívateľa a väčšinu akcií inicializuje on.
SSH sa skladá z 3 vrstiev:
Transportnej zabezpečuje overenie identity servera, prvotnú výmenu kľúčov, nastavenie kompresie, šifrovania a kontroly integrity. Tak isto sa stará o znovu vymenenie kľúčov po transporte 1GB dát alebo 1 hodine. Bežne beží cez TCP/IP spojenie.
Autentifikačnéj slúži na Overenie identity klienta, pre server, všetko vychádza zo strany klienta, ten si od užívateľa žiada heslo, a následne sa autentifikuje u servera.
Na identifikáciu sa používajú viaceré spôsoby:
Heslo , Verejný kľúč(aspoň podpora DSA, RSA kľúčových párov, alebo aj certifikáty), Klávesnicová interakcia (Server pošle nejakú zakódovanú na ktorú užívateľ musí správne reagovať napísaním do konzoly), a iné.
Spojovacej beží na autentifikačnej vrstve, a zabezpečuje rozdelenie šifrovaného tunela do kanálov, umožňujúcich viacero transportov na raz, nezávisle na sebe, pričom kanále sú obojsmerné.

Každý SSH server má vlastný kľuč, ktorým si klient vie overiť, že hovorí s tým serverom, s ktorým chce, a nie s niekým kto sa za neho vydáva. Na to ale musí mať klient niekde uloženú informáciu o kľúči a ku ktorému serveru patrí. To sa dá zabezpečiť dvoma spôsobmi, buď má klient lokálnu databázu s týmito údajmi, alebo každý server má kľuč pridelený Certifikačnou Autoritou(CA). Prvý spôsob nepotrebuje žiadnu centrálnu autoritu ani infraštruktúru, ale na druhej strane aktualizácia a údržba databázy môže časom byť dosť problematická. Pri druhom spôsobe klient pozná iba CA root kľúč, ktorým dokáže verifikovať všetky hostove kľúče certifikované CA, čím odpadá potreba údržby, klient má len jeden kľuč od CA, na druhej strane Server musí mať platnú certifikáciu od

CA, a kladie to veľké nároky na dôveryhodnosť CA. Súčasťou protokolu je možnosť aby pri prvom kontakte neautentifikoval kľuč servera, to umožňuje komunikáciu bez predchádzajúcej výmeny kľúča alebo certifikácie. Problémom je ale, že takto zmizne ochrana proti aktívnym Man-in-the-middle útokom, čiže to, že sa niekto postaví medzi teba a server a tebe sa vydáva za server, a server si myslí že si to ty. Táto možnosť by mala byť defaultne vypnutá, ale keďže neexistuje jednotná infraštruktúra kľúčov, robí protokol flexibilnejším a použiteľnejším do doby kým sa takáto infraštruktúra objaví.

SSH pakety obsahujú číslovanie od 1 do 255, ktoré vyjadruje na čo je správa určená a zároveň umožňuje kontrolovať integritu dát. Ak paket vypadne je možnosť si ho vyžiadať znova. Čísla sú rozdelené takto:
1-19 Všeobecné pre transportnú vrstvu
20-29 Dohodnutie použitých algoritmov.
30-49 Špecifické metódy na výmenu kľúčov*
50-59 Všeobecná autentifikácia používateľa
60-79 Pre špecifické spôsoby autentifikácie užívateľa*
80-89 Všeobecné spojovacieho protokolu
90-127 Správy súvisiace s kanálmi
128-191 Rezervované
192-255 lokálne rozšírenia

Tunneling cez SSH

Na použitie tunelovania pomocou SSH je jedna podmienka - aby obe strany mali umožnený a nastavený Port forwarding cez svoje firewally, či routre. Princíp je jednoduchý, vytvorí sa klasické SSH spojenie, pričom sa určia dva pevné porty, medzi účastníkmi sa vytvorí tunel a následne je komunikácia cez tunel prevádzaná pevne medzi týmito portami.

Táto technika je síce šikovná ale aj ľahko zneužiteľná, pretože umožňuje obísť programom v tuneli firewall, alebo iné obmedzenia na routeri, alebo na serveri pretože tieto prostriedky vidia len tunel a nie to, čo je v ňom. Napríklad ak vaša organizácia zamedzuje prehliadanie web stránok, ale povoľuje port forwarding, môžete sa napojiť na vzdialený SSH server, vytvoriť si k nemu tunel cez tieto obmedzenia a následne využiť server na prehliadanie webu, pričom požiadavky mu posielate vy, a výsledky sú cez tunel presmerované k vám.
Pozitívnejším využitím je sprístupnenie dát za NAT, zároveň je zabezpečená časť medzi SSH serverom a klientom, pričom to umožňuje bezpečné prepojenie dvoch privátnych sietí, cez takýto tunel.
Cez jeden tunel sa dá forwardovať viacero portov z rôznych destinácii.

8. NAT (Network Address Translation)

Tunelovanie vie preklenúť, ako sme si písali, viacero problémov v sieti, berie si to priam za svoju úlohu. Jeden z týchto problémov je však dosť osobitný, preto mu venujeme aj osobitnú kapitolu. Je to problém s NAT adresovaním.

Úvod do NAT

NAT – po slovensky preklad sieťových adries, je v dnešnej dobe veľmi rozšírená technika, ktorá má za účel ušetriť na verejných IP adresách, v prípade napojenia siete na internet, keďže schová celú sieť do jednej alebo menšieho počtu verejných IP adries, a zároveň môže fungovať aj ako akýsi primitívny Firewall.
NAT sa nachádza na routeri, cez ktorý sa daná sieť pripája k Internetu. Táto sieť má privátne IP adresy, ktoré identifikujú účastníkov siete a často nie sú použiteľné ako verejné IP adresy, alebo sú už obsadené, alebo by ľahko mohlo prísť ku kolízii, keď viacero zariadení by malo rovnakú adresu. Práve tu do hry vstupuje NAT, vo chvíli keď niekto zo siete chce nadviazať spojenie, vytvorí datagram kde ako zdrojovú adresu použije svoju privátnu adresu, tento balíček príde routeru ,na ktorom beží NAT protokol, ten zaregistruje, že tam nie je dobrá adresa, zapíše si do tabuľky od koho to išlo, potom vyberie nejakú voľnú verejnú IP adresu, alebo si podľa zdrojovej adresy z tabuľky vyberie konkrétnu IP a tou potom prepíše zdrojovú adresu v datagrame, zmení checksum hlavičky tak ,aby pasovala k novej adrese a zapamätá si vzťah medzi verejnou adresou, ktorú priradil a privátnou, z ktorej požiadavka prišla. Tento proces prebieha bez vedomia účastníkov spojenia. Čiže pre všetkých ostatných je zdrojom balíku router s NATom a vôbec netušia, že za ním sa nachádza ešte ďalšia sieť, kým pre zariadenie, ktoré k tej sieti patrí, je to obyčajný internetový server. Toto ale skoro vždy znemožňuje zariadeniam z vonka kontaktovať zariadenia, ktoré sa nachádzajú v sieti za NAT serverom, čo má ale aj svoju výhodu, keďže na váš počítač sa nemôže napojiť z vonka nikto pokiaľ spojenie nezačnete vy sami.
Rozšírením tohto je PAT (Port Address Translation), kedy sa k IP adresám pridávajú a aj prekladajú aj čísla Portov.
Pozrime sa teda na NAT z bližšia.

Prečo NAT

Prvá otázka, ktorá človeka napadá je istotne: „Prečo?“. Na čo vôbec treba prekladať IP adresy? Nestačilo by, keby každý počítač, ktorý má pripojenie na internet mal svoju vlastnú verejnú IP adresu? Takto sa zmýšľalo na začiatku Internetu, keď sa začalo používať IPv4 na adresovanie zariadení napojených na nejakú sieť (tento problém sme si už spomínali v IPv6 časti). Vtedy sa zdalo, že 32 bitové číslo, ktoré poskytuje 2^{32} (čo je vyše 4 mld) adries bude dostačujúce (v skutočnosti je ich prístupných niečo vyše 3mld, lebo veľké bloky adries, boli zarezervované). No pri boome internetu, ktorý nastal v posledných rokoch sa ukazuje, že to nie je pravda, pretože svoju vlastnú verejnú IP adresu potrebuje mať každý počítač, notebook, PDA, mobile s pripojením na internet ako aj každý server či už fyzický alebo virtuálny. V dnešnej dobe máme miliardy používateľov internetu, z ktorých veľká väčšina má počítač doma, ako aj v práci. Firmy majú stovky až tisíce serverov, ktoré spravujú ich dáta, nové mobily už nemôžu byť bez pripojenia na Internet atď. Takto veľmi rýchlo zistíme, že tie 3 mld sú málo. Na spomalenie tohto trendu a získanie času na vývoj a prechod na nové IPv6 adresovanie, sa prišlo s viacerými technológiami, pričom pravdepodobne najrozšírenejšou je práve NAT.
Ďalším dôvodom bola potreba zvýšenia bezpečnosti. S rozšírením internetu v 90tych rokoch sa network stal nástrojom aj pre zlých chlapcov, a tak sa firma s väčším počtom počítačov priamo napojených na internet stala zraniteľnejšou.
A tak sa prišlo s myšlienkou nepriameho pripojenia k internetu, ktorá je základom aj pre NAT. Tento spôsob podporili aj ďalšie fakty: - Väčšina účastníkov internetu sú klienti a tí vo všeobecnosti

nemusia byť nevyhnutne verejne prístupní. Tak isto drvivá väčšina komunikácie na Internete je inicializovaná zo strany klienta, ktorý o ňu žiada. – V prípade väčších sieti je v jednom okamihu zvyčajne na internet pripojených len o dosť menšia množina užívateľov a je vysoko nepravdepodobné, že by nastala situácie, kedy by spojenie potrebovali naraz všetci a aj pri prehliadaní webu sa k nemu v skutočnosti pripájame len v okamihu vyžiadania si informácie. – A komunikácia medzi sieťou a internetom prebieha cez router.
Najlepšie sa to prirovnáva k telefónnej sieti vo väčšom podniku. Tie slúžia hlavne na to, aby mohli zamestnanci volať von, ak treba niečo vybaviť, a nie je potrebné ,aby sa dalo jednotlivým zamestnancom dovolať z vonka. O to sa väčšinou stará ústredňa alebo nejaký systém a v jednom okamihu telefonuje z podniku len malá časť zamestnancov, a teda pre podnik by bolo neefektívne zaviesť priamu a samostatnú telefónnu linku pre každého zamestnanca, ale efektívnejšie je keď sa hovorí primajú v nejakej ústredni a cez systém klapiek alebo niečo podobne prevádzané na zamestnanca, ktorému sú určene.
Takéto niečo robí NAT.

Adresovania a rôzne druhy NAT

Router, na ktorom prebieha NAT, rozpoznáva 4 druhy adries. Prvé delenie je na vonkajšie a vnútorné adresy. Vnútorné sú adresy zariadení patriacich do privátnej siete, ktoré sa pripájajú von cez router. Logicky vonkajšie sú adresy všetkých zariadení mimo tejto siete nachádzajúcich sa na verejnej alebo v inej sieti. Oba druhy adries sa ešte delia na lokálne a globálne. Lokálna adresa je adresa, ktorú nesie datagram keď sa nachádza v privátnej sieti patriacej k routeru, kým globálna je tá, ktorú nesie datagram mimo tejto siete.
Z týchto dvoch delení dostaneme štyri adresy, ktoré rozpoznáva a spravuje NAT.
Vnútorná Lokálna – adresa zariadenia patriaceho do privátnej siete, v rámci tejto siete.
Vnútorná Globálna – verejná adresa ktorú dostane pre pripojenie sa na internej po prejdení cez NAT
Tieto dve Adresy a ich vzájomné výmeny sú hlavnou prácou NAT. Tieto dve sa vymieňajú vždy, keď prejde balíček cez NAT a smeruje od alebo z prístroja v jeho sieti.
Vonkajšia Globálna – Verejná adresa zariadenia mimo privátnej siete.
Vonkajšia Lokálna - Adresa ktorú ma prístroj z vonka, keď jeho balíček vo vnútri privátnej siete, táto adresa je väčšinou rovná Globálnej, keďže nie je dôvod ju meniť, ale nie je tomu tak vždy.
Je viacero typov NAT, a každý je v niečom špecifický, hlavne tým, nakoľko umožňuje adresovanie zariadení v privátnej sieti. No v prvom rade sa všetky dajú rozdeliť do dvoch kategórii:
Statické NAT
Statické NAT je vlastne naozaj len čisté prekladanie adries. Keďže sa vytvorí pevný zoznam, ktorý routeru povie presne, ktorú jednu verejnú IP adresu má priradiť konkávnej lokálnej IP adrese.
Čiže napríklad adresa 10.1.1.21 sa vždy preloží na povedzme 195.54.38.11. Toto je vhodné pre zariadenia, ktoré sú na privátnej sieti ale zároveň musia byť jednoznačne určiteľné aj na verejnej sieti. Tento druh NAT dovoľuje v určitej miere adresovať tieto zariadenia aj zvonka. Lenže ho treba manuálne nastaviť a potom aj sa oň starať.
Dynamické NAT
Tento NAT nielenže prekladá adresy zariadení, ale spĺňa aj druhú dôležitú úlohu, s ktorou bol NAT vyvinutý a to umožňuje šetriť verejné IP adresy. Vychádza z faktu, že v drvivej väčšine prípadov je počet zariadení, ktoré sa chcú v istom okamihu napojiť na verejnú sieť, výraznejšie menší ako počet všetkých zariadení na sieti, a teda nepotrebujeme taký veľký počet verejných IP adries.
Dynamický NAT má teda zoznam verejných adries, ktoré má k dispozícii, a v prípade, že sa zariadenie z privátnej siete chce napojiť na verejnú, vyberie mu zo zoznamu voľných adries jednu náhodne. To, ktorej lokálnej adrese priradil akú verejnú si potom zapamätá v translačnej tabuľke, a to kým trvá spojenie. Keď sa spojenie ukončí, tak sa tento záznam zmaže a pri ďalšom spojení dostane to isté zariadenie znova náhodne vybranú adresu.
Napríklad zariadenie 10.1.1.21 z predchádzajúceho príkladu sa tento krát napája cez dynamický NAT router, ktorý má k dispozícii 20 verejných adries 195.54.38.1-20 . Keď toto zaradenie

datagram, ktorý smerujúci von zo siete, router mu vyberie náhodne jednu z týchto 20 adries, teda pokiaľ už ju nepriradil inému zariadeniu, a zapamätá si daný vzťah. Keď spojenie skončí, záznam o priradení sa zmaže a adresa sa vráti do zoznamu voľných adries.

Ďalej sa NAT dá rozdeliť podľa funkčnosti na viaceré typy

Tradičný/ jednosmerný NAT

Tradičný NAT vychádza z toho, že drvivá väčšina spojení je zahájená klientom, ktorý sa zväčša nachádza na nejakej privátnej sieti za routerom, pričom keďže privátne adresy ani to ako sa im priraďujú verejné adresy nikto z vonka nevidí, tak nie je možné, aby pri tomto druhu NAT zahájil spojenie niekto z vonka siete.

Ako to funguje najlepšie ukáže príklad. Naše staré známe 10.1.1.21 sa chce napojiť na net, konkrétne na server povedzme 204.51.16.12 a tak pošle datagram na vytvorenie spojenia. Tento datagram má v hlavičke v políčku zdrojová adresa uvedené 10.1.1.21 a v políčku cieľovej adresy 204.51.16.12.

Tento datagram príde na router, ktorý má NAT, ten v hlavičke zbadá 10.1.1.21 a uvedomí si že to nie je správna verejná adresa a tak zmení zdrojovú adresu za 195.54.38.11 a pošle datagram cieľovému zaradeniu. To vygeneruje odpoveď, kde bude v zdrojovej adrese 204.51.16.12 a v cieľovej 195.54.38.11. Odpoveď príde na router a ten zbadá 195.54.38.11 v cieľovej adrese pozrie do prekladovej tabuľky a zamení ju za 10.1.1.21 a pošle danému zariadeniu. Samozrejme okrem tejto zmeny je treba urobiť často aj mnohé ďalšie, treba zmeniť checksum hlavičky, poprípade celého datagramu a často aj na mnohých ďalších miestach, čo spôsobuje komplikácie s NAT – tie rozoberieme ďalej.

Obojsmerný NAT

Hoci väčšinou vychádza žiadosť o spojenie z vnútra privátnej siete, niekedy jednoducho potrebujeme ísť opačne a žiadať spojenie z vonka. Preto sa vymyslelo obojsmerný NAT. Dôvodom prečo nám na to nestačí tradiční je to, že zariadenia na privátnej sieti poznajú adresy cieľových zariadení, a teda poslať ho prvý datagram von nie je problém. No tí z vonka nepoznajú privátnu adresu stroja na sieti. Oni vidia iba adresy, ktoré má router a v prípade, že ten nemá záznam vzťahu medzi verejnou a lokálnou adresou, nevie komu má nasmerovať daný balík.

Riešilo by to statické priradenie verejných adries k lokálnym. To by ale vyžadovala verejnú adresu pre každé zaradenie a jedným z hlavných účelov NAT je ušetriť tieto adresy, preto je to väčšinou dynamické NAT a tam to jednoducho nejde. Druhým riešením je spolupráca NAT a DNS, ktoré využíva miesto IP adries mená, ktoré pridelí jednotlivým zaradenia.

Povedzme máme na privátnej sieti zariadenie, ktoré má v DNS názov www.ikars.com, naň sa chce napojiť nejaké zariadenie z vonka, tak pošle tento názov DNS serveru, ktorý pracuje na tejto privátnej sieti, server si nájde vnútornú lokálnu adresu zariadenia a pošle ju NATu , NAT si na žiadosť DNS servera vytvorí translačný záznam a priradí tejto lokálnej adrese globálnu, samozrejme si to zapamätá. DSN server potom pošle žiadateľovi túto vnútornú globálnu adresu, ktorá mu už umožní, vďaka vytvorenému záznamu v prekladovej tabuľke NATu komunikovať s konkrétnym zariadením.

PAT / Overloaded NAT

Jedno- aj oboj- smerný NAT prekladajú lokálne za globálne adresy, v pomere 1 k 1, aby umožnil prístup z privátnej do verejnej siete. Lenže aby sa tieto ušetrili, je najviac používané dynamický NAT, a teda pre väčšiu sieť je k dispozícii menšie množstvo verejných adries. Lenže to má jeden vážny problém, a to ak naraz do verejnej siete bude chcieť pristupovať viac zariadený ako je verejných IP, tie ďalšie sa nebudú môcť napojiť.

Našťastie riešenie tohto je zabudované už priamo v TCP/IP a tou je ďalšia zložka, ktorá sa využíva spolu s IP adresou a to je adresa portu, cez ktorý prebieha komunikácia. Ako sme spomínali už vyššie PAT , znamená prekladanie adries Portov. Porty slúžia v TCP/IP na to, aby mohlo naraz nezávisle od seba bežať viacero služieb, ako napríklad viaceré okná prehliadača, alebo k internetu nejaký messenger a podobne. Čo sa teda bude robiť je zrejmé, v prípade PAT je IP adresa v hlavičke sprevádzaná číslom portu preložená, pričom router zmení aj číslo portu, cez ktorý bude spojenie

prebiehať, inak robí to isté čo predtým. Výhodou je fakt, že máme stovky čísiel portov, a teda nezávisle na sebe môžu naraz pod jednou verejnou IP adresou, len s iným číslom portu, pracovať mnohé zariadenia z veľkej siete.

Overlaping

Overlaping – po slovensky prekryv. Predchádzajúce druhy NAT neriešili problém možnej kolízie IP adries. A ani nepotrebovali, keďže všetky tri sú určené v prvom rade na prístup z privátnej do verejnej siete, hlavne Internet, kde sa to, že by dve zariadenia mali rovnakú verejnú IP, jednoducho nemohlo stať, lebo verejné IP boli takým spôsobom prideľované. Lenže čo ak treba poslať požiadavku zariadeniu v inej privátnej sieti? Tam veľmi ľahko môže dôjsť k situácii že IP adresy kolidujú. Napríklad naše 10.1.1.21 chce poslať požiadavku o spojenie zariadeniu v inej privátnej sieti, ktorého adresu pozná a je 10.1.1.120, lenže ako náš router môže vedieť či cieľom je niekto vonku alebo 10.1.1.120 na našej vlastnej sieti?

Presnejšie riešenie tohto problému sa nazýva aj Twice NAT = dvojnásobný NAT. Predchádzajúce riešenia prekladali len zdrojovú adresu, ak datagram išiel z našej siete, alebo cieľovú ak išiel z vonka, pri dvojnásobnom NATe sú prekladané obe adresy v oboch smeroch, a to tak, že si nenamapuje len privátnu sieť, ktorú obsluhuje, ale aj možné prekrývajúce sa adresy. No tak ako pri obojsmernom NAT aj tu potrebuje pomoc DNS.

Najlepšie to ilustruje príklad. Predpokladajme, že pri nastavovaní našej siete sa admin hlúpo nazdával, že ju nebude treba pripojiť na internet a tak použil adresy tvaru 18.0.0.0, pričom tento blok je verejne rezervovaný pre MIT. Samozrejme mýlil sa a sieť sa napojila na internet a z tejto siete sa 18.0.0.13 snaží spojiť s www.twicenat.mit.edu. Náš dvojnásobný NAT router dostane žiadosť, v spolupráci s DNS interpretuje túto adresu a vo svojej tabuľke zistí, že k tomuto názvu má priradiť 172.16.44.55, čo je privátna neroutovateľná adresa. Túto adresu pošle s5 18.0.0.13 a tá ju dá ako cieľovú adresu. Keď potom datagram so zdrojovou adresou 18.0.0.13 a cieľovou 172.16.44.55, ako klasický NAT preloží zdrojovú adresu, pretože pre nás to nie je platná verejná adresa, povedzme na 195.54.38.11, v cieľovej adrese ale nájde adresu 172.16.44.55 to nie je dobre ale vďaka svojej tabuľke vie, že za túto adresu má dosadiť 18.1.2.3, ktorý patrí želanému cieľu. Výsledný datagram potom má v hlavičke ako zdroj 195.54.38.11 a ako cieľ 18.1.2.3.

Zápory NAT

Takto by sa zdalo že NAT je úplné skvelý, alebo aspoň dobrý, no nie je tomu tak. V skutočnosti je toto riešenie dosť nešikovné, napriek svojím výhodám a rozšírenosti a v konečnom dôsledku je len na to, aby získal čas na zavedenie IPv6.

Prečo? Pretože na komunikáciu po nete a v sietiach sa používa okrem IP aj kopa ďalších protokolov, ktoré potrebujú pracovať s IP adresami na viacerých miestach v datagrame nie len v hlavičke, lenže NAT vie sám o sebe len prekladať adresy v hlavičke, maximálne ešte prepočítať checksum v nej, ale prepočítať checksum celého datagramu, či zistiť či netreba preložiť adresu ešte aj niekde inde ako v hlavičke, to už je nad jeho pôvodne sily. To znamená že protokoly ako FTP, ktoré si IP a Porty, posielajú aj v príkazoch nezvláda sám. Znemožňuje použitie bezpečnostných protokolov, keďže tie sledujú či neprišlo k zmenám v dátach a nevedia rozlíšiť medzi zmenou, ktorú by spravil nejaký hacker a zmenou, ktorú spravil NAT router. Síce je pravda, že je vďaka NATu ťažšie nepovolene preniknúť k zariadeniu na privátnej sieti, no zároveň je rovnako ťažšie preniknúť k nemu aj legitímne. Tieto a ďalšie problémy vyžadujú riešenia navyše, väčšinou vo forme nejakej aplikácie, ktorá NATu pomôže, čím samozrejme zvyšuje komplexnosť siete a jej spravovania, zmeny IP adries, znepríjemňujú debagovanie, a všetko dokopy to ide na úkor výkonu.

NAT a Tunneling

No a čo NAT a Tunneling? Pri tunnelingu jedno zariadenie vytvára tunel k druhému zariadeniu, na to ale potrebuje väčšinou vedieť jeho verejnú adresu a port, na ktorom môže tunel vytvoriť. Teraz si predstavme, že medzi dvoma zariadeniami, ktoré chcú vytvoriť tunel sa nachádza NAT router. Pokiaľ je tunel inicializovaný zo súkromnej siete, ktorú zakrýva NAT, je to ešte ten lepší prípad, no

i tu sa rýchlo môžu vyskytnúť problémy. Značná časť tunelovacích protokolov používa rôzne bezpečnostné opatrenia, ktoré majú zabrániť tretej osobe manipulovať s datagramom a tu je zrazu niekto tretí (NAT), kto zrazu datagram prepisuje, a tak bezpečnostný nemusí tunel vôbec vytvoriť, lebo nevie rozlíšiť želaný zásah od neželaného, sám od seba. Alebo môžu posielať informácie o zariadení, ktoré chce vytvoriť alebo niečo robiť s tunelom vo vnútri datagramu (jeho adresu), a pre istotu túto adresu porovnáva s hlavičkou aby zistil, či je to poriadku, lenže tieto dve sú odlišné, a tak tunel znova raz nevznikne. A o opačnom prípade keď niekto chce vytvoriť tunel z vonka na súkromnú sieť, ani nehovoríme. Nielenže tam budú už spomenuté problémy, ale pri väčšine NAT riešení, zaradenie z vonka nevie vnútornú adresu v súkromnej siete, keďže vidí len verejnú adresu NAT. NAT vlastne narúša priame spojenie Point to Point(Point to Point conectivity), ktoré väčšina z týchto protokolov vyžaduje.
Tento problém, ktorý sa týka nielen tunelovania ale aj mnohých iných protokolov ako VoIP a podobne, sa nazýva NAT traversal problém (problém preklenutia NAT). Ako sa už spomínalo pri obojsmernom NAT, sú spôsoby ako sa cez tento problém preniesť, ale všetky vyžadujú pomoc zvonka (hlavne DNS), keďže NAT sám ho riešiť nevie, čím komplikujú prenos dát, aj ho môžu celkom výrazne spomaliť.
Riešenie tohto problému nie je jednoduché, pretože existuje veľké množstvo osobitných adaptácii a implementácií NAT, a preto neexistuje jedno univerzálne riešenie a treba vždy hľadať vhodne pre tú verziu ktorá je na našej sieti využívaná, čo nemusí byť jednoduché.

Ku koncu ponúkame výpis niektorých riešení k NAT traversu:
NAT traversal protokoly a techniky založené na chovaní sa NAT

5. Simple Traversal of UDP over NATs (STUN)
6. Traversal Using Relay NAT (TURN)
7. NAT-T Negotiation of NAT-Traversal in the IKE
8. Teredo tunneling uses NAT traversal to provide IPv6 connectivity.
9. Session Border Controller (SBC)
10. UDP hole punching
11. TCP hole punching

NAT traversal založený na ovládaní NAT

- Realm-Specific IP (RSIP)
- Middlebox Communications (MIDCOM)
- SOCKS
- NAT Port Mapping Protocol (NAT PMP)
- Internet Gateway Device (IGD) Protocol, defined by the Universal Plug and Play (UPnP) Forum.
- Application Layer Gateway (ALG)

NAT traversal kombinujúci niekoľko techník.
Interactive Connectivity Establishment (ICE)

9. Iné tunelovanie

Tunelovanie ako metóda zaobalenia (ukrytia) a odoslania informácie má veľmi široké využitie, a aj veľa rôznych spôsobov prevedenia.
V našej práci sme neobsiahli niektoré zaujímavé špeciálne prípady, ktoré však často závisia priamo od situácie a z toho hľadiska boli pre nás príliš konkrétne.
Napr. sme len spomenuli obchádzanie firewallu, kedy sa protokol bránený firewallom zaobalí do „čistého protokolu“ (SSH alebo HTTP).
Podobne s obchádzaním pravidiel proxy serverov. Využíva sa príkaz HTTP CONNECT. Keďže HTTP je klasicky používaný protokol, proxy ho má povolený a takto môže sprostredkovať pripojenie na konkrétny server. Väčšinou sa však zakazuje prístup ku CONNECT príkazu.
Existuje dokonca aj tzv. DNS tunelovanie, ktoré zas funguje na klasickom posielaní požiadaviek na DNS server. Ak pred doménu servera napíšeme šifrovanú správu, takáto požiadavka môže byť cez DNS server na daný server poslaná. A pokiaľ sme vlastníkom cieľového servera, vieme docieliť, že server aj na takúto naoko nezmyselnú požiadavku odpovie.
Tieto spôsoby sa ale odkláňajú od tém, ktorým sme sa venovali, a skôr odhaľujú konkrétne bezpečnostné diery ako ponúkajú nové štandardy.

Ale práve schopnosť tunelovania, poskytnúť kúzelné riešenie zdanlivo nemožnej situácie, robí túto metódu tak príťažlivou.