**NAT, FIREWALL, PROTOKOLY, PORTY**

**Network Address Translation**

**NAT** - P**řeklad síťových adres** je způsob úpravy síťového provozu přes router **přepisem zdrojové nebo cílové IP adresy**, **často i změnu čísla TCP/UDP portu** u průchozích IP paketů. K tomu patří i **změna kontrolního součtu** (u IP i TCP/UDP), aby změny byly brány v úvahu. NAT se většinou používá pro **přístup více počítačů z lokální sítě na Internet** **pod jedinou veřejnou adresou** (viz gateway). NAT ovšem může způsobit problémy v komunikaci mezi klienty a snížit rychlost přenosu. **Vznikl jako důsledek omezeného počtu veřejných IP adres** (IPv4 má 32 bitů a část z nich je navíc rezervována pro speciální účely). Každý uživatel dnešního internetu nemůže mít adresu z vnějšího rozsahu, **NAT umožňuje celou vnitřní síť ukrýt za adresu jedinou.**

V typické konfiguraci používá lokální síť některý z rozsahů **Neveřejných IP adres**. **Router má přidělenu neveřejnou adresu**, ale také je **spojen s Internetem veřejnou adresou** přidělenou od ISP. Jakmile jde paket z lokální sítě do Internetu, je jeho **zdrojová adresa** (neveřejná) **přeložena na veřejnou**. **Router si uchovává základní data o každém aktivním spojení** (adresu cíle, port). Když se vrátí odpověď na router, využi**je data získaná při odchozí fázi a určí kam na vnitřní síť je třeba odpověď zaslat**. Pro systém na Internetu **se jeví** **router jako zdroj i cíl komunikace.**

**Překlad síťových adres je funkce, která umožňuje překládání adres**. Což znamená, že **adresy z lokální sítě přeloží na jedinečnou adresu**, která slouží pro vstup do jiné sítě (např. Internetu), **adresu překládanou si uloží do tabulky pod náhodným portem**, při odpovědi si v **tabulce vyhledá port a pošle pakety na IP adresu přiřazenou k danému portu**. **NAT je vlastně jednoduchým proxy serverem**.

NAT může být softwarového typu (Nat32, Kerio Winroute firewall), nebo hardwarového typu (router s implementací nat).

**Vlastní komunikace – Jak to vlastně funguje**

Uživatel Honza má svůj PC v lokální síti. V síti je i Router, který je propojen s internetem.   
Uživatel Honza se chce dotázat na webový server v Internetu a zjistit počasí v Praze.

Co víme

* Honzův PC má Neveřejnou IP 192.168.1.3
* Router v naší LAN má Neveřejnou IP 192.168.1.1
* Router v naší LAN má Veřejnou (Přidělenou od ISP) IP 12.13.14.15
* V Internetu je Webový server s Veřejnou IP 40.30.20.10

Jak proběhne komunikace:

* Honzův PC vytvoří paket, kde je: **Zdrojová IP a port (192.168.1.3:42301)** a **Cílová IP a port (40.30.20.10:80)**
* Paket opustí Honzův PC a dojde na Router
* Router Paket rozbalí, změní Zdrojovou IP z 192.168.1.3 na svou Veřejnou IP (12.13.14.15) a změní zdrojový port (42301) na nějaký náhodný (26035).
* Router si uloží do NAT tabulky, že je aktivní spojení mezi 192.168.1.3 a 40.30.20.10 a zapíše si prohozené IP i Porty.
* Paket dojde na Webový server a Sever odešle odpověď. Cílová adresa bude 12.13.14.15:26035
* Paket dojde na router, Router se podívá, jaké spojení je na portu 26035 a jaká IP adresa se dotazovala na IP 40.30.20.10.
* Ze své NAT tabulky zjistí, že je to zařízení s adresou 192.168.1.3:42301
* Router odesílá paket na tuto adresu

**Výhody**

* umožňuje připojit více počítačů na jednu veřejnou IP adresu - řeší se tak nedostatek přidělených veřejných IP adres.
* zvyšuje bezpečnost počítačů připojených za NATem (potenciální útočník nezná opravdovou IP adresu)

**Druhy NAT**

**SNAT**

Příkladem použití SNATu je zamaskování IP adresy routovaných paketů adresou routeru.

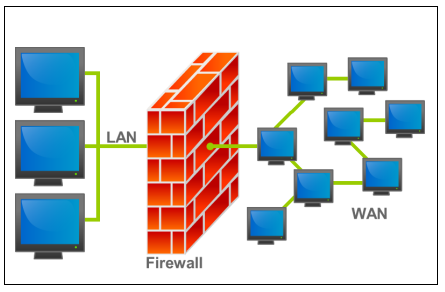
Pravidlo způsobí, že pokud z vnitřní sítě (řekněme z adresy 192.168.0.2) přijde paket, který má v úmyslu opustit router přes vnější rozhraní, tak dojde k nahrazení jeho původní IP adresy za adresu 1.2.3.4. To znamená, že příjemce datagramu bude mít "pocit" jako kdyby s ním nekomunikovala stanice s nesměrovatelnou adresou 192.168.0.2, ale přímo router 1.2.3.4, který pak zajistí, aby se pakety v opačném směru dostaly zpátky k 192.168.0.2

**DNAT**

Naopak DNAT použijeme tehdy, když potřebujeme měnit IP adresu adresáta datagramu. Používá se to při přesměrovávání datagramů na jiného adresáta. Například někde v síti máme umístěný WWW server a rádi bychom na něj přesměrovali všechny požadavky směřující přes náš router.

Tímto zápisem jsme zařídili, že pokud se na našem routeru objeví TCP segment určený pro port 80 (WWW server), tak se jeho cílová adresa přepíše na 192.168.0.2. (Což je Web Server v naší síti).

**FIREWALL**

**Firewall je síťové zařízení hlídající provoz mezi lokální sítí (LAN) a „zbytkem světa“ (WAN),** které slouží k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti a zabezpečení. Zjednodušeně se dá říct, že slouží jako **kontrolní bod**, který **definuje pravidla pro komunikaci mezi sítěmi**, které od sebe odděluje. Tato pravidla historicky vždy zahrnovala **identifikaci zdroje a cíle dat** (**zdrojovou a cílovou IP adresu**) a **zdrojový a cílový port**, což je však pro dnešní firewally už poměrně nedostatečné – modernější firewally se opírají přinejmenším o **informace o stavu spojení, znalost kontrolovaných protokolů a případně prvky IDS.**

Firewally se během svého vývoje řadily zhruba do následujících kategorií:

* Paketové filtry
* Aplikační brány
* Stavové paketové filtry
* Stavové paketové filtry s kontrolou známých protokolů a popř. kombinované s IDS

**Paketové filty**

**Nejjednodušší a nejstarší forma firewallování**, která spočívá v tom, že **pravidla přesně uvádějí, z jaké adresy a portu na jakou adresu a port může být doručen procházející paket**, tj. **kontrola se provádí na třetí a čtvrté vrstvě** modelu síťové komunikace **OSI.**

**Výhodou tohoto řešení je vysoká rychlost zpracování**, proto se ještě i dnes používají na místech, kde není potřebná přesnost nebo důkladnější analýza procházejících dat, ale spíš jde o vysokorychlostní přenosy velkých množství dat.

**Nevýhodou je nízká úroveň kontroly procházejících spojení**, která zejména u složitějších protokolů (např. FTP, video/audio streaming, RPC apod.) nejen nedostačuje ke kontrole vlastního spojení, ale pro umožnění takového spojení vyžaduje otevřít i porty a směry spojení, které mohou být využity jinými protokoly, než bezpečnostní správce zamýšlel povolit.

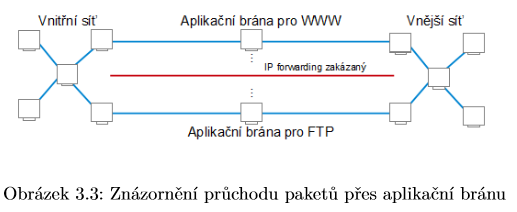
**Aplikační brány**

Aplikační brány na rozdíl od paketových filtrů **zcela oddělují sítě, mezi které jsou postaveny**. Říká se jim většinou **Aplikační brány**, někdy také **Proxy firewally**. Veškerá komunikace přes aplikační bránu probíhá formou dvou spojení – **klient (iniciátor spojení) se připojí na aplikační bránu (proxy)**, ta **příchozí spojení zpracuje** a na základě požadavku klienta **otevře nové spojení k serveru**, kde klientem je aplikační brána. Data, která aplikační brána dostane od serveru, pak zase v původním spojení předá klientovi. **Kontrola se provádí na sedmé (aplikační) vrstvě síťového modelu OSI**.

Tento typ firewallu se liší od předešlých především tím, že **není možné komunikovat mezi dvěma sítěmi napřímo**, tak jako tomu bylo u stavového či paketového filtru.

**Filtr pracuje na aplikační vrstvě ISO/OSI**, proto lépe rozumí **obsahu paketů** a jejich návaznosti. **Veškerá komunikace mezi klientem a serverem probíhá přes třetí stranu, kterou zastává proxy brána (proxy server).** Nikdo z vnější sítě proto nevidí do vnitřní sítě. Pakety určené do vnější sítě nejdříve putují na proxy bránu se zdrojovou adresou původce, na aplikační bráně se zdrojová adresa změní na adresu brány, která dále vystupuje jako zprostředkovatel. Zpětná komunikace funguje na podobném principu, tedy paket, který je určen pro naši síť, jde nejdříve na bránu, ze které je následně poslán na cílovou adresu.

Proxy bránu si lze představit jako kombinaci klienta a serveru v jednom, navíc se specializací pro jednu službu. Existují tedy brány zvlášť pro služby WWW, FTP, SSH, elektronickou poštu a mnoho dalších.



**Výhodou aplikační brány** je kvalitní **zabezpečení známých protokolů**, díky tomu, že **pracuje na aplikační vrstvě**. **Nevýhodou** však je poměrně **velká náročnost na hardware počítače**, brány nejsou schopné pracovat na stejné rychlosti jako například paketové filtry a **mají vyšší latenci**.

**Stavové paketové filtry**

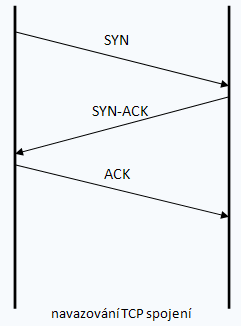
**Stavové paketové filtry** provádějí kontrolu **stejně jako jednoduché paketové filtry**, navíc si však **ukládají informace o povolených spojeních**, které pak mohou využít při rozhodování, zda procházející **pakety patří do již povoleného spojení** a **mohou být propuštěny**, nebo zda **musí znovu projít rozhodovacím procesem**.

**To má dvě výhody** – jednak se tak **urychluje zpracování paketů** již povolených spojení, jednak lze v pravidlech pro firewall **uvádět jen směr navázání spojení** a firewall bude samostatně schopen povolit i odpovědní pakety a u známých protokolů i další spojení, která daný protokol používá.

**K největším výhodám stavových paketových filtrů** patří jejich **vysoká rychlost**, poměrně **slušná úroveň zabezpečení** a **mnohonásobně snazší konfigurace** – a díky zjednodušení konfigurace i **nižší pravděpodobnost chybného nastavení** pravidel obsluhou.

**Nevýhodou je obecně nižší bezpečnost**, než poskytují aplikační brány.

**Stav spojené** lze sledovat různými způsoby, přičemž nelze zvolit jeden konstantní. Například u protokolu **TCP** je využit **trojcestný** **handshake**. Jako **zahajovací paket** je považován paket, který má nastavený příznak **SYN**, čímž firewall pozná, že se **jedná o nové spojení**. **Jeli služba** na serveru **k dispozici**, odpoví odesláním paketu s nastaveným příznakem **SYN a ACK**. Celou **úvodní komunikaci završí klient**, který pošle serveru paket s nastaveným příznakem **ACK**, čímž je **spojení ustaveno jako ESTABLISHED**. Filtr pak propustí pouze pakety s již navázaným a povoleným spojením.



**Stavové paketové filtry s kontrolou protokolů a IDS**

**Moderní stavové paketové filtry** kromě **informací o stavu spojení** a schopnosti **dynamicky otevírat porty** pro různá řídící a datová spojení složitějších známých protokolů implementují něco, co se v marketingové terminologii různých společností nazývá nejčastěji **Deep Inspection**. Znamená to, že **firewally jsou schopny kontrolovat procházející spojení až na úroveň korektnosti procházejících dat** známých protokolů i aplikací. Mohou tak například **zakázat průchod http spojení**, v němž **objeví indikátory**, že se ne**jedná o požadavek na WWW server**, ale **tunelování jiného protokolu**, což často využívají klienti P2P sítí (ICQ, gnutella, napster, apod.), nebo když **data v hlavičce e-mailu nesplňují požadavky RFC**.

Nejnověji se do firewallů integrují tzv. in-line **IDS (Intrusion Detection Systems** – **systémy pro detekci útoků)**. Tyto systémy **pracují podobně jako antiviry** a pomocí **databáze signatur a heuristické analýzy** jsou schopny **odhalit vzorce útoků** i ve zdánlivě nesouvisejících pokusech o spojení, např. **skenování adresního rozsahu**, **rozsahu portů**, zná**mé signatury útoků** uvnitř povolených spojení apod.

**Výhodou těchto systémů je vysoká úroveň bezpečnosti kontroly** procházejících protokolů při zachování relativně snadné konfigurace, poměrně **vysoká rychlost kontroly** **ve srovnání s aplikačními branami**, nicméně **je znát významné zpomalení** (zhruba o třetinu až polovinu) **proti stavovým paketovým filtrům**.

**Nevýhodou** je zejména to, že z hlediska bezpečnosti designu je základním pravidlem bezpečnosti udržovat bezpečnostní systémy co nejjednodušší a nejmenší. **Tyto typy firewallů integrují obrovské množství funkcionality** a **zvyšují tak pravděpodobnost**, že v některé části jejich kódu **bude zneužitelná chyba**, která povede ke kompromitování celého systému.

**Protokoly**

**Protokol udává způsob komunikace mezi koncovými stanicemi v síti.**

**Vzájemná komunikace stejnolehlých vrstev různých uzlů** se musí řídit předem dohodnutými **pravidly**. **Pravidla této vzájemné komunikace** **definuje tzv. protokol.**

**Protokol definuje:** Co si komunikující strany posílají, jaký to má formát a význam, kódování..  
Komunikující strany si předávají tzv. **PDU (Protokol Data Unit).** Každý PDU má dvě části: **Hlavička** (Header) a **Tělo**. Protokol definuje dál, **jak komunikace probíhá**, jak mají strany reagovat na různé situace atd. **Protokol ošetřuje nestandartní situace**.

V každé vrstvě může existovat a fungovat několik relativně samostatných entit (entita může být například proces, démon, úloha atd.)

**Definuje pravidla komunikace mezi entitami stejnolehlých vrstev** (každý protokol vždy patří do určité konkrétní vrstvy). **Protokol určuje způsob, jakým je realizována určitá služba**. Pro každou vrstvu může existovat několik alternativních protokolů.

Představa, že si stejnolehlé vrstvy skutečně předávají přímo mezi sebou jednotky PDU(Rámce, pakety, segmenty) je pouze iluze. (Ve skutečnosti je předávají nižší vrstvě s požadavkem na doručení druhé straně)

**Protokoly fyzické vrstvy** - stanovují elektrické vlastnosti rozhraní (napěťové úrovně, průběhy, kmitočty, modulace, rychlosti, elektrické vlastnosti kabelů) a mechanické vlastnosti (tvary, velikosti a zapojení konektorů).

**Protokoly linkové vrstvy** – Ethernet

**Protokoly Síťové vrstvy** – IP, ICMP, IGMP, IGRP, IPSec, ARP,

**Protokoly Transportní vrstvy** – TCP, UDP

**Protokoly Relační vrstvy** – SSL, NFS,

**Protokoly Prezentační vrstvy** – Telnet,

**Protokoly Aplikační vrstvy** – DHCP, FTP, http, IMAP, LDAP, POP3, SSH, SMB, SMTP, SNMP, TFTP, Telnet

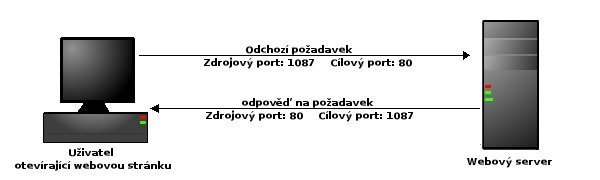
**Porty**

**Síťový port je speciální číslo (0 až 65535)**, které slouží v [počítačových sítích](https://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%A1_s%C3%AD%C5%A5) **při komunikaci pomocí protokolů**[**TCP**](https://cs.wikipedia.org/wiki/Transmission_Control_Protocol)**a**[**UDP**](https://cs.wikipedia.org/wiki/User_Datagram_Protocol) k **rozlišení aplikace v rámci počítače**.

**TCP/UDP Pomocí portů ví, které aplikaci má poslat data, která mu přišly.**

**Příklad:** [Server](https://cs.wikipedia.org/wiki/Server), který je používán k odesílání a přijímání [elektronické pošty](https://cs.wikipedia.org/wiki/Elektronick%C3%A1_po%C5%A1ta) bude pravděpodobně poskytovat služby [SMTP](https://cs.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) a [POP3](https://cs.wikipedia.org/wiki/Post_Office_Protocol). Ty jsou na serveru obsluhovány rozdílnými procesy a čísla portů se použijí k rozlišení, která data patří jakému procesu. Obvykle je tomu tak, že SMTP server naslouchá na portu 25, zatímco POP3 na portu 110

**Ve zkratce k čemu Porty potřebujeme:**



* Uživatel (Browser) si nastaví Zdrojový port, na kterém bude očekávat odpověď od Webového serveru
* Na Webový server pošle dotaz s portem 80, protože Webový server používá http a naslouchá na portu 80
* Webový server zpracuje dotaz a pošle odpověď na cílový port 1087
* Odpověď dojde na Uživatele
* Transportní vrstva zjistí, že Port je 1087. Uvědomí si, že Aplikace, která Vysílala na portu 1087 dotaz, je Browser a předá mu Data.

**Well-Known-Porty**

|  |  |  |
| --- | --- | --- |
| **Port** | **Protokol** | **Popis** |
| 21, 20 | [FTP](https://cs.wikipedia.org/wiki/File_Transfer_Protocol), FTP-data | Přenos souborů (řídící a datové spojení) |
| 22 | [SSH](https://cs.wikipedia.org/wiki/Secure_Shell) | Secure shell - šifrovaná obdoba protokolu telnet, přenosy souborů |
| 23 | [Telnet](https://cs.wikipedia.org/wiki/Telnet) | Vzdálený textový terminál – nešifrovaná komunikace |
| 25 | [SMTP](https://cs.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) | Simple Mail Transfer Protocol – přenos [elektronické pošty](https://cs.wikipedia.org/wiki/E-mail) |
| 53 | [DNS](https://cs.wikipedia.org/wiki/Domain_Name_System) | Domain Name System – překlad doménových jmen |
| 67, 68 | DHCP | Dynamic Host Configuration Protocol – Přiděluje IP |
| 69 | TFTP | Triviální FTP |
| 80 | [HTTP](https://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol) | HyperText Transfer Protocol – přenos WWW stránek i jiných dat |
| 110 | [POP3](https://cs.wikipedia.org/wiki/Post_Office_Protocol) | Post Office Protocol version 3 – stahování [elektronické pošty](https://cs.wikipedia.org/wiki/E-mail) |
| 143 | [IMAP](https://cs.wikipedia.org/wiki/Internet_Message_Access_Protocol) | Internet Message Access Protocol 4 – vzdálená správa poštovní schránky s [elektronickou poštou](https://cs.wikipedia.org/wiki/E-mail) |
| 161 | [SNMP](https://cs.wikipedia.org/wiki/Simple_Network_Management_Protocol) | Simple Network Management Protocol |
| 443 | [HTTPS](https://cs.wikipedia.org/wiki/HTTPS) | Šifrovaný přenos HTTP protokolu přes [TLS](https://cs.wikipedia.org/wiki/Transport_Layer_Security) |