# Zabezpečení datových sítí - Filtrace provozu, Firewall, NAT

Honeypot (česky hrnec medu)

je návnada síťově dostupného zdroje pro útočníky, která zaznamená jejich činnost, může být nasazen v síti jako nástroj pro sledování a včasné varování. Techniky používané útočníky, kteří se pokoušejí narušit tyto síťové návnady jsou analyzovány během a po útoku, kvůli dohledu nad novými exploit technikami. Taková analýza může být použita pro další zpřísnění zabezpečení sítě a dalšímu testování pomocí návnady. Návnada může být využita k přesměrování útočníkovy pozornosti od legitimních serverů a přinutit ho plýtvat časem a energií k vedení útoku na falešný cíl. Podobně jako honeypot je honeynet síť s úmyslně nastavenými chybami v zabezpečení. Jejím cílem je také zmást útočníka, aby útočil na návnadu a jeho útoky se mohly zaznamenat a dále využít k zlepšení zabezpečení sítě. Honeynet obvykle obsahuje jeden nebo více honeypotů.

## Firewall

Jedná se o síťové zařízení, které slouží k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti a zabezpečení. Zjednodušeně se dá říci, 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 (Intrusion Detection System). 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](https://cs.wikipedia.org/wiki/Intrusion_Detection_System)

Podle umístění lze firewally dělit na:

* Síťový firewall – samostatné hardwarové řešení pro ochranu počítačové sítě
* Personální firewall – realizován na koncových stanicích (počítačích)

### Paketové filtry

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](https://cs.wikipedia.org/wiki/Paket), tj. kontrola se provádí na třetí a čtvrté vrstvě modelu síťové komunikace [OSI](https://cs.wikipedia.org/wiki/Referen%C4%8Dn%C3%AD_model_ISO/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](https://cs.wikipedia.org/wiki/File_Transfer_Protocol), video/[audio streaming](https://cs.wikipedia.org/w/index.php?title=Audio_streaming&action=edit&redlink=1), [RPC](https://cs.wikipedia.org/wiki/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

Jen o málo později, než jednoduché paketové filtry, byly postaveny firewally, které na rozdíl od paketových filtrů zcela oddělily sítě, mezi které byly 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](https://cs.wikipedia.org/wiki/Proxy" \o "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](https://cs.wikipedia.org/wiki/Referen%C4%8Dn%C3%AD_model_ISO/OSI) (proto se těmto firewallům říká aplikační brány).

Jedním vedlejším efektem použití aplikační brány je, že server nevidí zdrojovou adresu klienta, který je původcem požadavku, ale jako zdroj požadavku je uvedena vnější adresa aplikační brány. Aplikační brány díky tomu automaticky působí jako nástroje pro překlad adres ([NAT](https://cs.wikipedia.org/wiki/Network_address_translation)), nicméně tuto funkcionalitu má i většina paketových filtrů.

Výhodou tohoto řešení je poměrně vysoké zabezpečení známých protokolů.

Nevýhodou je zejména vysoká náročnost na použitý HW – aplikační brány jsou schopny zpracovat mnohonásobně nižší množství spojení a rychlosti, než paketové filtry a mají mnohem vyšší latenci. Každý protokol vyžaduje napsání specializované proxy, nebo využití tzv. generické proxy, která ale není o nic bezpečnější, než využití paketového filtru. Většina aplikačních bran proto uměla kontrolovat jen několik málo protokolů (obyčejně kolem deseti). Původní aplikační brány navíc vyžadovaly, aby klient uměl s aplikační branou komunikovat a neuměly dost dobře chránit svůj vlastní operační systém; tyto nedostatky se postupně odstraňovaly, ale po nástupu stavových paketových filtrů se vývoj většiny aplikačních bran postupně zastavil a ty přeživší se dnes používají už jen ve velmi specializovaných nasazeních.

### 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 pakety odpovědí a u známých protokolů i další spojení, která daný protokol používá. Například pro FTP tedy stačí nastavit pravidlo, ve kterém povolíte klientu připojení na server pomocí FTP a protože se jedná o známý protokol, firewall sám povolí navázání řídícího spojení z klienta na port 21 serveru, odpovědi z portu 21 serveru na klientem použitý zdrojový port a po příkazu, který vyžaduje přenos dat, povolí navázání datového spojení z portu 20 serveru na klienta na port, který si klient se serverem dohodli v rámci řídícího spojení a pochopitelně i pakety odpovědí z klienta zpět na port 20 serveru. Zásadním vylepšením je i možnost vytváření tzv. virtuálního stavu spojení pro bezstavové protokoly, jako např. [UDP](https://cs.wikipedia.org/wiki/User_Datagram_Protocol) a [ICMP](https://cs.wikipedia.org/wiki/ICMP).

K největším výhodám stavových paketových filtrů patří jejich vysoká rychlost, poměrně slušná úroveň zabezpečení a ve srovnání s výše zmíněnými aplikačními branami a jednoduchými paketovými filtry řádově mnohonásobně snazší konfigurace – a díky zjednodušení konfigurace i nižší pravděpodobnost chybného nastavení pravidel obsluhou.

Nevýhodou je obecně nižší bezpečnost, než poskytují aplikační brány.

### 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* nebo *Application Intelligence*. 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](https://cs.wikipedia.org/wiki/Http) spojení, v němž objeví indikátory, že se nejedná o požadavek na [WWW](https://cs.wikipedia.org/wiki/WWW) server, ale tunelování jiného protokolu, což často využívají klienti [P2P](https://cs.wikipedia.org/wiki/Peer-to-peer) sítí ([ICQ](https://cs.wikipedia.org/wiki/ICQ), [gnutella](https://cs.wikipedia.org/wiki/Gnutella" \o "Gnutella), [napster](https://cs.wikipedia.org/wiki/Napster" \o "Napster), apod.), nebo když data v hlavičce [e-mailu](https://cs.wikipedia.org/wiki/E-mail) nesplňují požadavky [RFC](https://cs.wikipedia.org/wiki/RFC) apod.

Nejnověji se do firewallů integrují tzv. *in-line IDS* (*Intrusion Detection Systems* – systémy pro detekci útoků). Tyto systémy pracují podobně jako [antiviry](https://cs.wikipedia.org/wiki/Antivir) a pomocí databáze signatur a [heuristické analýzy](https://cs.wikipedia.org/wiki/Heuristick%C3%A1_anal%C3%BDza) jsou schopny odhalit vzorce útoků i ve zdánlivě nesouvisejících pokusech o spojení, např. skenování adresního rozsahu, rozsahu portů, známé 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.

### Bezpečnostní politika

Nastavení pravidel pro komunikaci přes firewall se běžně označuje termínem „bezpečnostní politika firewallu“, zkráceně „bezpečnostní politika“. Bezpečnostní politika zahrnuje nejen samotná pravidla komunikace mezi sítěmi, ale u většiny dnešních produktů také různá globální nastavení, překlady adres (NAT), instrukce pro vytváření šifrovaných spojení mezi šifrovacími branami (VPN – Virtual Private Networks), vyhledávání možných útoků a protokolových anomálií (IDS – Intrusion Detection Systems), autentizaci a někdy i autorizaci uživatelů a správu šířky přenosového pásma (bandwidth management).

## NAT (Network Address Translation)

je v počítačových sítích způsob úpravy síťového provozu procházejícího přes router přepisem zdrojové nebo cílové IP adresy, případně i hlaviček protokolů vyšší vrstvy (např. číslo portu u TCP, UDP, ICMP Query ID u ICMP atd.). NAT může být implementován softwarově na běžném počítači (např. v jádře Linuxu pomocí iptables/netfilter) nebo může být realizován přímo ve firmware či hardware routeru. NAT se většinou používá pro přístup více počítačů z lokální sítě do Internetu prostřednictvím jediné veřejné IP adresy (viz gateway). NAT však znemožňuje přímou komunikaci mezi klienty (end–to–end spojení) a může snížit rychlost přenosu.

### Source NAT

V IP datagramu se dle pravidla změní zdrojová IP, případně i zdrojový port u protokolů TCP/UDP. Výraz S-Nat je využíván některými firmami i k označení "secure NATu".

### Destination NAT

V IP datagramu se dle pravidla změní cílová IP/cílový port TCP/UDP. D-NAT se často používá v tzv. [captive portálech](https://cs.wikipedia.org/w/index.php?title=Captive_port%C3%A1l&action=edit&redlink=1" \o "Captive portál (stránka neexistuje)) veřejných Wi-Fi hotspotů, nebo při zveřejnění služby, umístěné v privátní síti, pro veřejně přístupnou IP adresu.

### Maškaráda (NAT 1:N)

Maškaráda (dynamický NAT) je zřejmě nejčastěji používanou variantou NATu. Dorazí-li na router z vnitřní sítě IP datagram, je mu změněna zdrojová IP a tcp/udp port, paket je následně odeslán dále dle směrovací tabulky. Změněný i původní port včetně původní IP je uchován v dynamické NAT tabulce. Příchozím paketům na vnějším rozhraní routeru je porovnáván cílový port oproti NAT tabulce, podle které je následně přepsána cílová adresa. Příkladem může být NAT v domácích routerech překládající adresy v rozsahu domácí privátní sítě na veřejnou adresu přidělenou poskytovatelem, nebo jiný privátní rozsah poskytovatele (v případě použití tzv. [carrier-grade NATu](https://cs.wikipedia.org/wiki/Carrier-grade_NAT" \o "Carrier-grade NAT)).

### NAT 1:1

Adresy jsou překládány tak, že adresa, nebo dokonce jen služba v síti lokální má vyhrazenu adresu na vnějším rozhraní routeru.[[8]](https://cs.wikipedia.org/wiki/Network_address_translation#cite_note-8) Tato konfigurace bývá řešením menších ISP pro poskytnutí veřejných IPv4 koncových uživatelů za [carrier-grade NATem](https://cs.wikipedia.org/wiki/Carrier-grade_NAT" \o "Carrier-grade NAT).

### Výhody

* umožňuje připojit více počítačů na jednu veřejnou [IP adresu](https://cs.wikipedia.org/wiki/IP_adresa), čímž se obchází nedostatek [IPv4](https://cs.wikipedia.org/wiki/IPv4) adres
* ztěžuje rozkrytí struktury sítě připojené za NATem, nicméně NAT sám o sobě nelze považovat za bezpečnostní opatření[[11]](https://cs.wikipedia.org/wiki/Network_address_translation#cite_note-#1-11)

### Nevýhody

* zařízení za NATem nemají skutečné připojení k Internetu a není např. možné se snadno připojit k jinému zařízení za NATem
* nemožnost blokování (greylisting) IP, za kterou jsou kromě zlomyslného uživatele nebo napadeného zařízení rovněž legitimní uživatelé
* NAT znemožňuje správnou funkcionalitu některých software
* snižuje výdrž baterie u mobilních zařízení

## VPN (virtual private network)

je v informatice prostředek k propojení několika počítačů prostřednictvím nedůvěryhodné počítačové sítě (např. veřejný Internet). Lze tak snadno dosáhnout stavu, kdy spojené počítače budou mezi sebou moci komunikovat, jako kdyby byly propojeny v rámci jediné uzavřené privátní (a tedy většinou důvěryhodné) sítě. Při navazování spojení je totožnost obou stran ověřována pomocí digitálních certifikátů, dojde k autentizaci, veškerá komunikace je šifrována, a proto lze takové propojení považovat za bezpečné.

VPN vytváří virtuální point-to-point propojení skrze tunneling protokolu přez již existijící síť. Aby mohla VPN fungovat tak jak jí všichni známe musíme celý náš provoz routovat skrze tento tunel až na zařízení které to VPN hostuje.

### Typy VPN

**Site-to-site** … propojuje dvě sítě, nebo skupiny kanceláří například k datacentru. Propojení může běžet přes ne tak standartní síť jako je například dvě IPv6 sítě připojené přes Ipv4 síť.

**Remote access** (host-to-network) … propojuje jednotlivce se sítí, poskytuje přístup do enterprise sítě, jako je intranet. Můžou používat zaměstnanci, když pracují z domova, nebo přístup z mobile k důležitým funkcionalitám bez potřeby je zveřejňovat na internet.

## Antivirový program (zkráceně antivir)

je počítačový software, který slouží k identifikaci, odstraňování a eliminaci počítačových virů a jiného škodlivého softwaru (malware). K zajištění této úlohy se používají dvě odlišné techniky:

* prohlížení souborů na lokálním disku a ram, které má za cíl nalézt sekvenci odpovídající definici některého počítačového viru v databázi
* detekcí podezřelé aktivity nějakého počítačového programu, který může značit infekci. Tato technika zahrnuje analýzu zachytávaných dat, sledování aktivit na jednotlivých portech či jiné techniky.

### Virové slovníky/databáze

Při kontrole souboru antivirový program zjišťuje, zda se nějaká jeho část neshoduje s některým ze známých virů, které má zapsány v databázi. Pokud je nalezena shoda, má program tyto možnosti:

* pokusit se opravit/vyléčit soubor odstraněním viru ze souboru (pokud je to technicky možné)
* umístit soubor do karantény (virus se dále nemůže šířit, protože ho nelze dále používat)
* smazat infikovaný soubor (i s virem)

### Nebezpečné chování

Metoda zjištění nebezpečného chování se oproti virovým databázím nesnaží najít známé viry, namísto toho sleduje chování všech programů. Pokud se takový program pokusí zapsat data do spustitelného programu, antivirus například označí toto nebezpečné chování a upozorní uživatele, který je antivirovým programem vyzván k výběru dalšího postupu.

### Sandbox

neboli pískoviště, napodobuje systém a spouští .exe soubory v jakési simulaci. Po ukončení programu software analyzuje sandbox, aby zjistil nějaké změny, ty mohou ukázat právě přítomnost virů. Tato metoda může taky selhat a to pokud jsou viry nedeterministické a výsledek nastane za různých akcí nebo akce nenastanou při běhu - to způsobí, že je nemožné detekovat virus pouze z jednoho spuštění.