

# Bezpečnost v IT: Firewally

Jan Ježek, Kerio Technologies / Samepage Labs

# Úvod

Jan Ježek

- Do r. 2012 vedoucí vývoje produktu Kerio Control (dříve Kerio WinRoute Firewall, ještě dříve WinRoute Pro)
- Kerio Technologies také známé díky:  
Kerio Connect, Kerio Operator  
Samepage



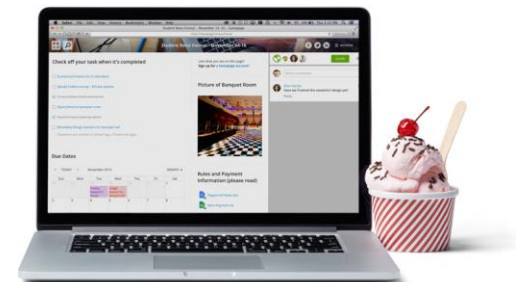
Kerio**Connect**



Kerio**Control**



Kerio**Operator**



# Program přednášky

- Firewall: co to je / původ / dělení / historie
- Princip fungování / komponenty
- Překlad adres (NAT)
- Doplnkové funkce
- Obcházení firewallu / lidský faktor

# Program přednášky

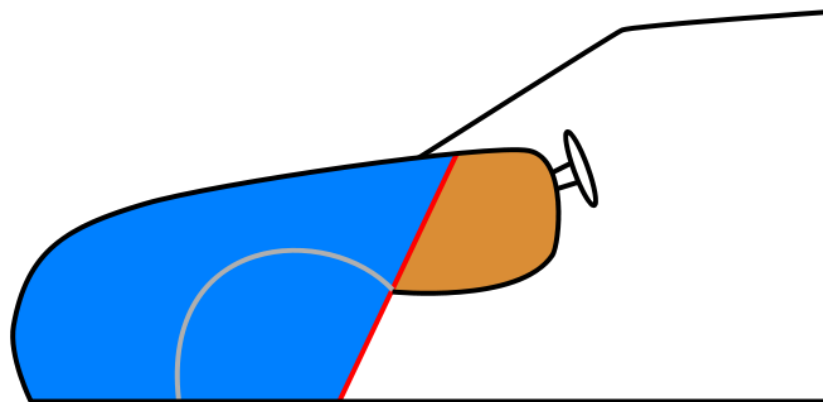
- **Firewall: co to je / původ / dělení / historie**
- Princip fungování / komponenty
- Překlad adres (NAT)
- Doplnkové funkce
- Obcházení firewallu / lidský faktor

# Firewall: co to je?



# Firewall: co to je?

Firewall = „Protipožární zed“



Původ slova není znám. Poprvé snad ve filmu WarGames.

# Firewall: co to je?

Definice:

Jeden bod mezi dvěma sítěmi, kterým musí procházet veškerý provoz mezi nimi.

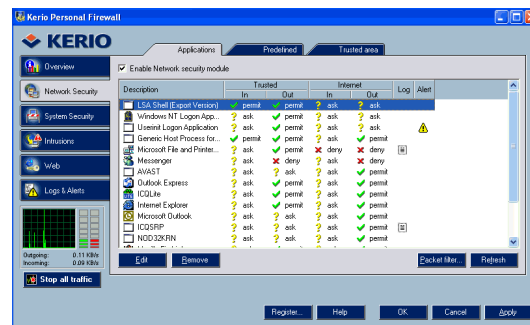
Bariéra mezi námi a nimi, pro libovolnou hodnotu „nimi“.



# Dělení firewallů

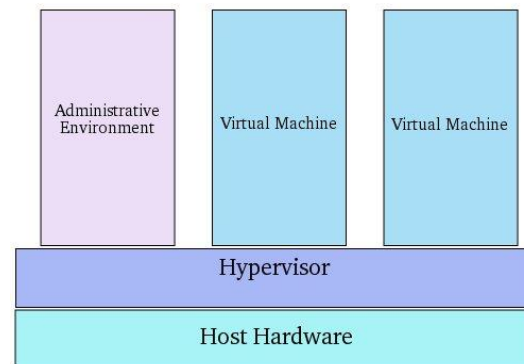
Podle použití:

- Síťové
- Osobní
- Serverové
- Web Application Firewalls



Podle platformy

- Hardwarové
- Softwarové
- Virtuální
- Cloudové – spravované (managed)





# Historie firewallů

Počátky lze vysledovat na přelomu 80. a 90. let:

- Objevuje se Internet ;-)
- resp. přestává být doménou „kolegů z univerzit“

2.11.1988 – první internetový červ „Morris“

- Robert Tappan Morris, student na Cornell University, NY
- Údajně 6000 nakažených počítačů (10% tehdejšího internetu)
- Přinesl vystřízlivění ohledně bezpečnosti
- Červ neměl škodit (dle Morrise)
- 400 hodin prací, \$10,000 pokuta, podmínka



# Historie firewallů

1987 – AT&T Gate

1991 – DEC SEAL

- První komerčně dodávaný firewall
- Méně než 10 000 řádek kódu, napsaný za týden
- \$75000 + náklady na instalaci

1993 – FWTK (Firewall Toolkit)

- Opensource

1994 – CheckPoint Firewall-1

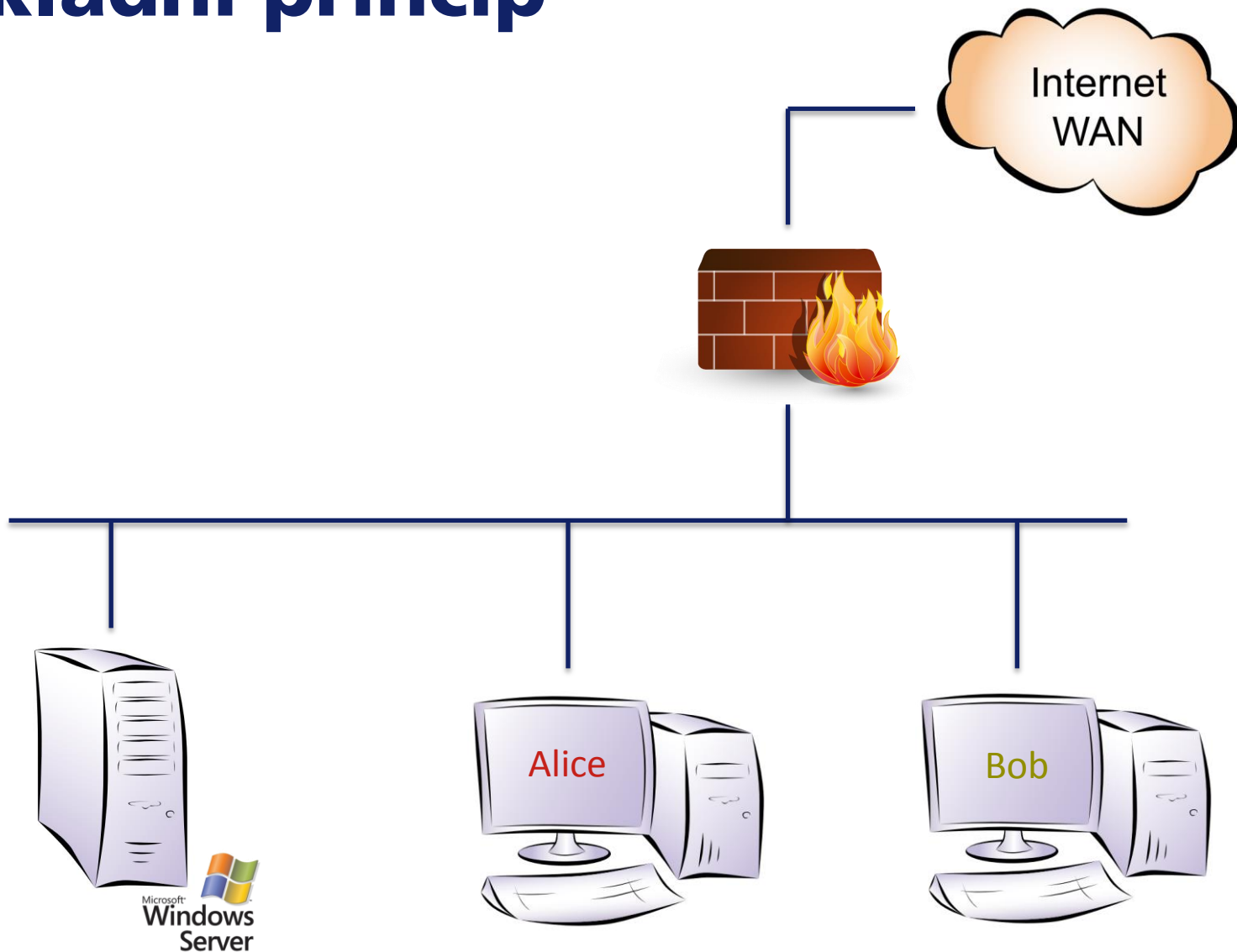
- Započal trend důrazu na výkon oproti bezpečnostním funkcím.

1997 – Tiny Software ;-)

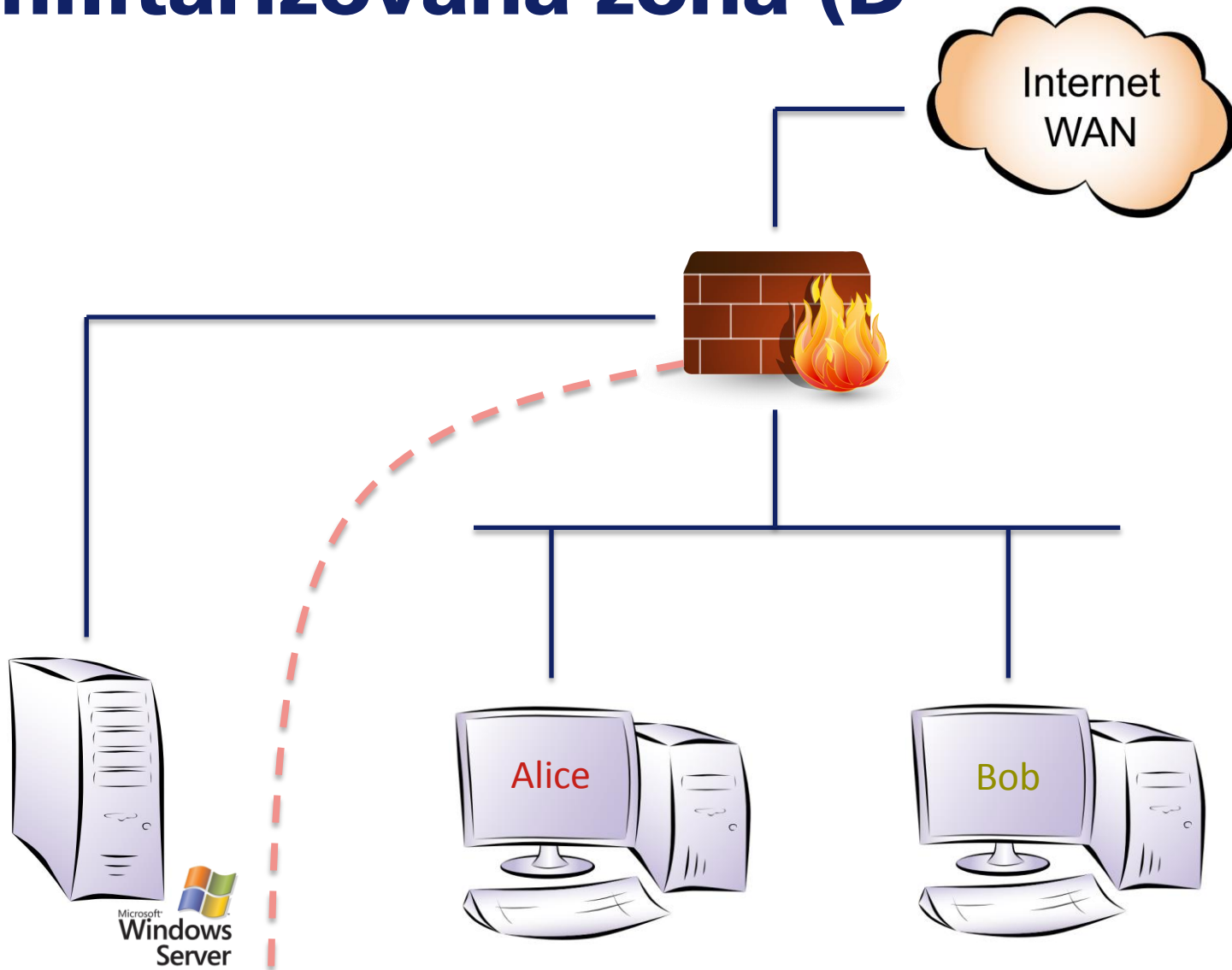
# Program přednášky

- Firewall: co to je / původ / dělení / historie
- **Princip fungování / komponenty**
- Překlad adres (NAT)
- Doplnkové funkce
- Obcházení firewallu / lidský faktor

# Základní princip



# Demilitarizovaná zóna (DMZ)



# Komponenty Firewallu

## Paketový filtr

- Pracuje nad jednotlivými pakety, umí je propustit, zahodit, modifikovat.
- Provádí překlad adres (NAT)
- Řídí využití šířky pásma

## Aplikační filtr

- Pracuje nad vyššími protokoly (HTTP, SIP, ...)

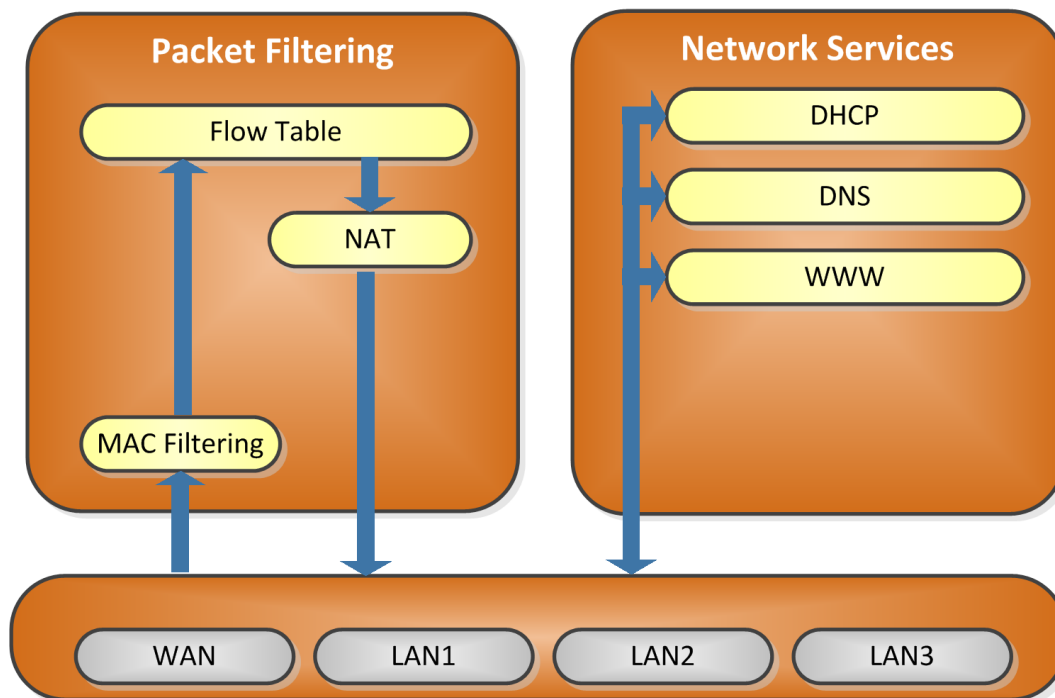
## Filtr obsahu

- Sleduje obsah přenášených dat, např. kontroluje na přítomnost virů, spywaru apod.

## IDS/IPS (Intrusion Detection/Prevention System)

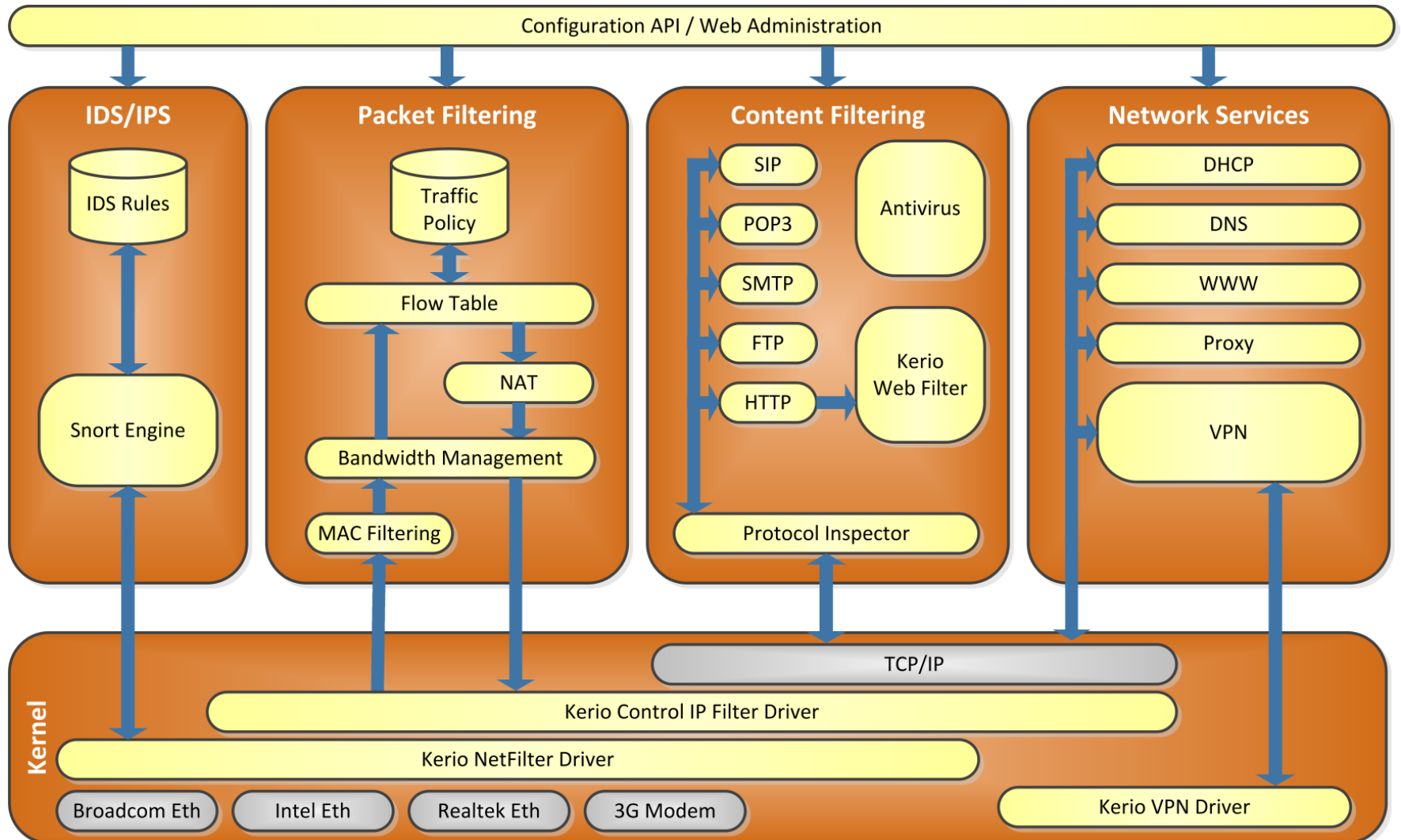
## Síťové služby a provisioning

# Základní architektura firewallu





# Architektura firewallu



# Paketový filtr

- Povoluje, zahazuje, či modifikuje pakety na základě politiky.
- Politika je sada pravidel, které pakety povolit a které zahodit.
- Pravidla v politice se procházejí v daném pořadí, pokud některé vyhoví, další se již nehledají.

Příklad v iptables v Linuxu:

```
iptables -A INPUT -p tcp --dport 80 -j DROP
```

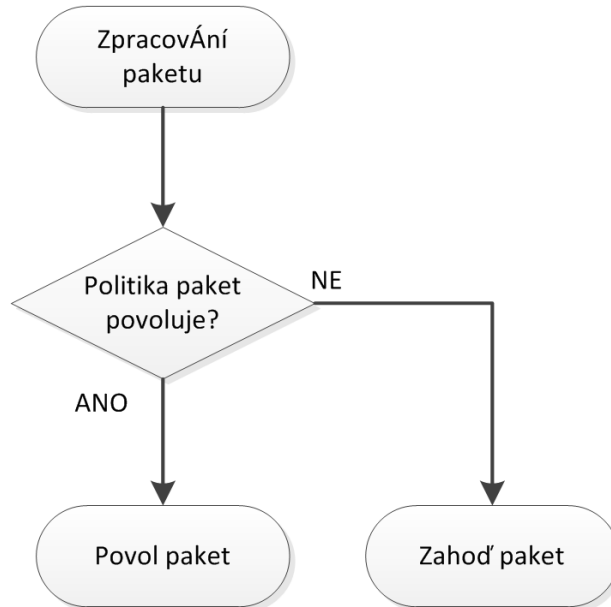
```
iptables -A INPUT --source 10.0.0.1 -j ALLOW
```

Typy paketového filtru:

- Bezstavový
- Stavový
- Smíšený / hybridní

# Bezstavový paketový filtr

- Aplikuje politiku bez ohledu na předchozí komunikaci
- Dnes se v praxi moc nepoužívá



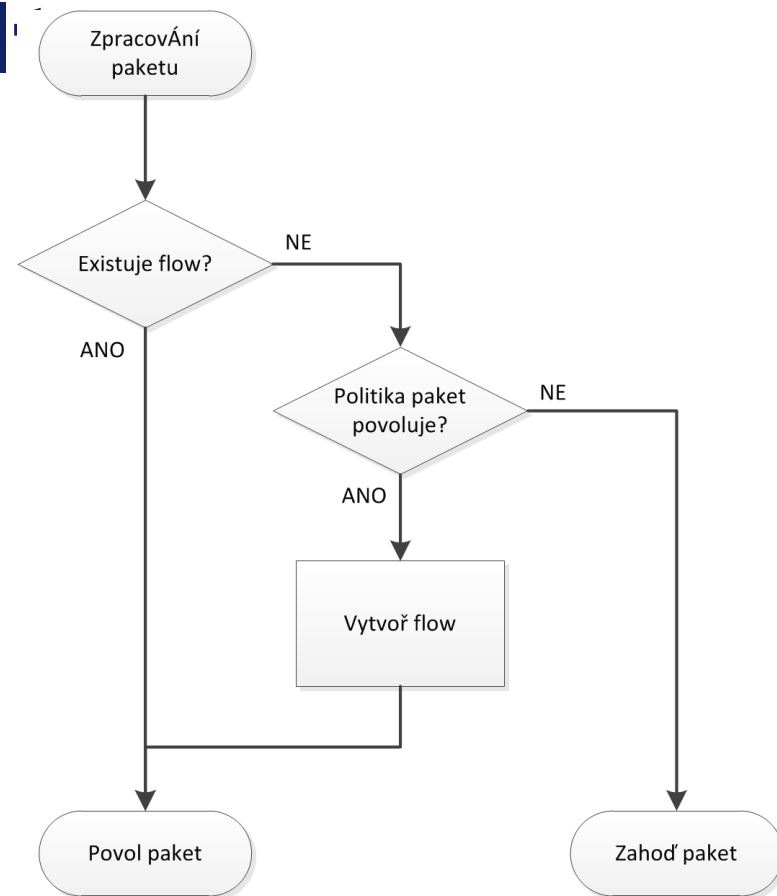
Příklad v iptables:

```
iptables -A INPUT -i eth0 -j DROP
```

Zahodí veškeré pakety přicházející z rozhraní eth0

# Stavový paketový filtr

- Udržuje si stav tzv. flow (relací, spojení, ...)
- Umí k nim jednotlivé pakety přiřadit.
- Politiku aplikuje teprve tehdy, když nelze paket přiřadit k žádnému existujícímu flow
- Pravidla politiky pracují s flows, nikoliv s pakety



## Traffic Rules

Name	Source	Destination	Service	Action
<input checked="" type="checkbox"/> Povol odchozí provoz	Trusted/Local Interfaces	Internet Interfaces	Any	Allow

# Stavový paketový filtr

Příklad tabulky flows:

Zdroj	Cíl	Protokol
147.228.5.5 : 4567	8.8.8.8 : 53	UDP
147.228.67.4 : 1024	147.228.57.10 : 80	TCP

# Smíšený / Hybridní Paketový filtr

- Kombinace obou přístupů
- Umí filtrovat bezstavově a do rozhodovacích podmínek přidat i stav flow
- Příkladem je netfilter v Linuxu

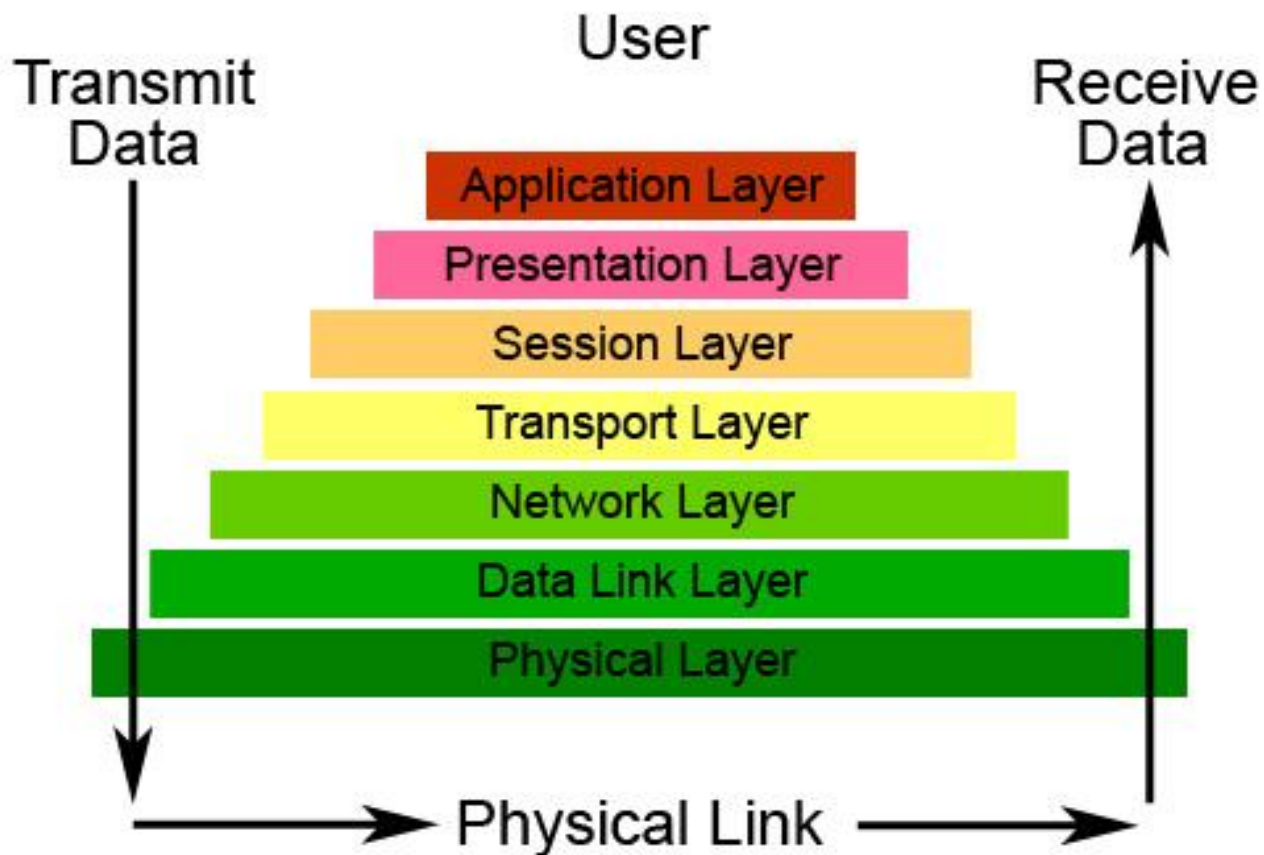
Příklad v iptables v Linuxu:

```
iptables -A OUTPUT -I eth0 -j ALLOW
```

```
iptables -A INPUT -m state --state ESTABLISHED -j ALLOW
```

# Aplikační filtr

## The Seven Layers of OSI



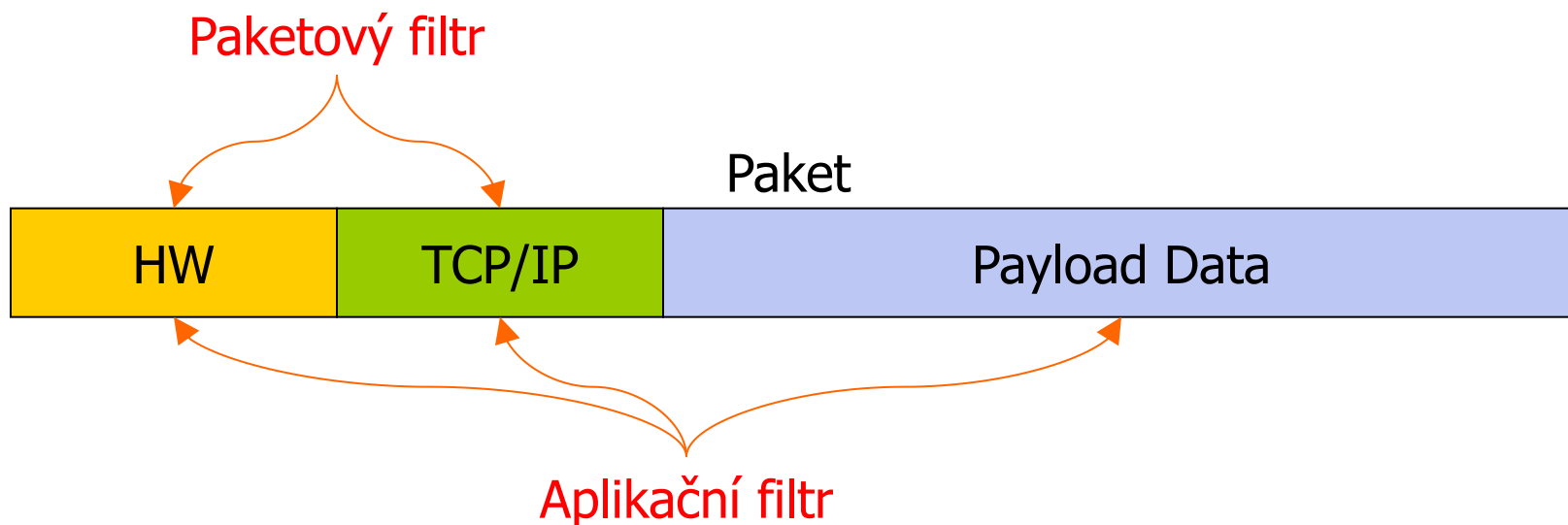


# Aplikační filtr

- Zajímá se o datovou část paketu (payload)
- Pracuje nad aplikačními protokoly (HTTP, FTP, SIP, ...)
- Chrání proti útokům na konkrétní aplikace, např.:
  - Útoky na známé bezpečnostní díry ve web serveru
  - Zakáže používání starších verzí browserů
- Kontroluje integritu protokolu, např.:
  - nemohu na HTTP portu provozovat P2P
  - nepovolí mi poslat nadstandardně dlouhou HTTP hlavičku

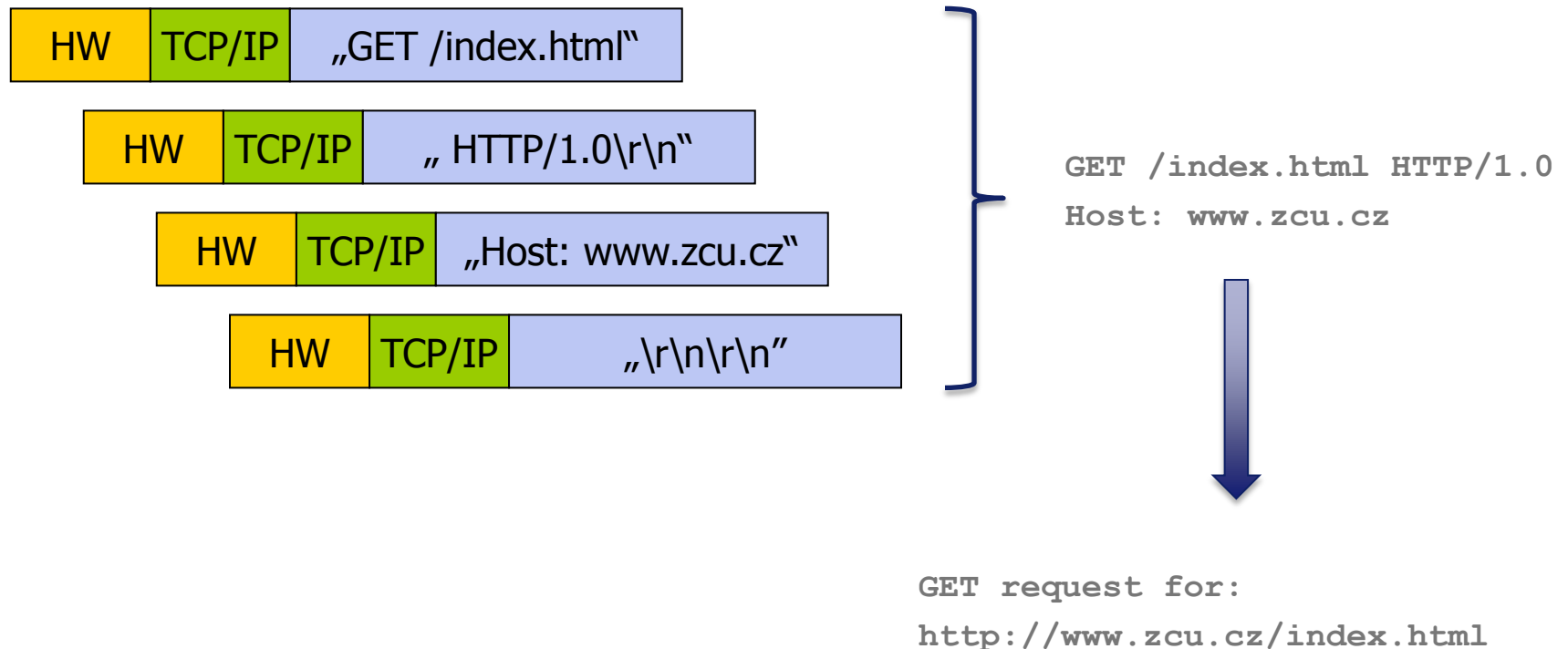
# Jak funguje aplikační filtr

Paketový filtr (většinou) sleduje pouze hlavičky paketů, aplikační filtr se zajímá o data:



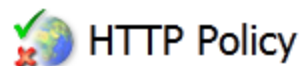
# Jak funguje aplikační filtr



Stavově kombinuje data z více paketů v rámci jednoho flow.  
Na výsledek pak aplikuje politiku:



# Aplikační filtr

Filtrování webových adres



<input checked="" type="checkbox"/> Name	Action	URL
<input checked="" type="checkbox"/> Zakaž celý Seznam	 Deny	*.seznam.cz
<input checked="" type="checkbox"/> Povol ZČU	 Deny	www.zcu.cz

- Akce aplikačního filtru nelze plně nahradit paketovým filtrem
- Je sice možné blokovat IP adresu web serveru
- Ale IP adresa webových serverů se může měnit nebo jich může být více

# Aplikační filtr

Více možností jak zareagovat na filtrovanou událost:

- Zahodit paket
- Zahodit celý flow
- Pozměnit obsah dat v protokolu
- Např.: přesměrovat uživatele na „zakazovací stránku“:

## Access Denied

---

Requested page  
<http://anonymizer.nntime.com/>

You do not have permission to access this site!

Kerio Web Filter categories  
[Anonymizer](#). [Report wrong category](#).

You may get access to the site by unlocking the restriction temporarily. [Unlock](#)

# Filtr obsahu

- Nastavba nad aplikační filtr
- Ještě hlouběji analyzuje data:
  - Anti-virus
  - Anti-spam
  - Analyzování kategorií webů
  - Kontrola, že obrázky jsou „mládeži přístupné“
- Často používán vedle bezpečnosti i pro řízení produktivity pracovníků

# Intrusion Detection/Prevention System

Sleduje síťový provoz a snaží se detekovat, případně zabránit útokům.

IDS pouze detekuje a upozorňuje na podezřelé stavy.

IPS umí těmto útokům zabránit blokováním příslušného provozu.

Typy IDPS:

- Založené na signaturách
- Analyzující chování sítě

Většinou používá data ze všech vrstev filtrování: paketového, aplikačního i obsahového.



# IDS Snort

<http://www.snort.org>

- Open source „etalon“ mezi IDPS systémy.
- Původně a standardně pouze IDS, ale umí i režim IPS.
- Založený na signaturách
- Pracuje samostatně, i bez firewallu

# Snort Signatory

Snort má vlastní „programovací jazyk“ pro definici IDS pravidel (signatur).

Příklad:

```
drop tcp $HOME_NET any -> $EXTERNAL_NET
$HTTP_PORTS (msg:"Fake Anti-Spyware";
flow:established,to_server; content:"POST ";
depth:5; nocase; uricontent:"/chkvs.php?mac=0";
nocase;
pcr:"/mac=0\w\:\w\w\:\w\w\:\w\w\:\w\w\:\w\w\:\w\w/Ui";
classtype:trojan-activity; sid:2007642; rev:5;)
```

# Blacklisty IP adres

- Součást IDPS systémů
- Různé zaměření blacklistů:
  - Známé útočící rozsahy adres
  - Kompromitované počítače
  - Command & Control servery botnetů
  - Spamovací servery

# Botnety



Historical list of botnets  [\[edit\]](#)

Date created <span>↕</span>	Date dismantled <span>↕</span>	Name <span>↕</span>	Estimated no. of bots <span>↕</span>	Spam capacity (bn/day) <span>↕</span>	Aliases <span>↕</span>
2009 (May)	<a href="#">2010-Oct (partial)</a>	<a href="#">BredoLab</a>	30,000,000 <sup>[14]</sup>	3.6	Oficla
2008 (around)	<a href="#">2009-Dec</a>	<a href="#">Mariposa</a>	12,000,000 <sup>[15]</sup>	?	
2008 (November)		<a href="#">Conficker</a>	10,500,000+ <sup>[16]</sup>	10	DownUp, DownAndUp, DownAdUp, Kido
2010 (around)		<a href="#">TDL4</a>	4,500,000 <sup>[17]</sup>	?	TDSS, Alureon
?		<a href="#">Zeus</a>	3,600,000 (US only) <sup>[18]</sup>	?	Zbot, PRG, Wsnpoem, Gorhax, Kneber
2007 (Around)		<a href="#">Cutwail</a>	1,500,000 <sup>[19]</sup>	74	Pandex, Mutant (related to: Wigon, Pushdo)
2008 (Around)		<a href="#">Sality</a>	1,000,000 <sup>[20]</sup>	?	Sector, Kuku
2009 (Around)	2012-07-19	<a href="#">Grum</a>	560,000 <sup>[21]</sup>	39.9	Tedroo
?		<a href="#">Mega-D</a>	509,000 <sup>[22]</sup>	10	Ozdok
?		<a href="#">Kraken</a>	495,000 <sup>[23]</sup>	9	Kracken
2007 (March)	2008 (November)	<a href="#">Srizbi</a>	450,000 <sup>[24]</sup>	60	Cbeplay, Exchanger
?		<a href="#">Lethic</a>	260,000 <sup>[25]</sup>	2	none
2004 (Early)		<a href="#">Bagle</a>	230,000 <sup>[25]</sup>	5.7	Beagle, Mitglieder, Lodeight
?		<a href="#">Marina Botnet</a>	6,215,000 <sup>[25]</sup>	92	Damon Briant, BOB.dc, Cotmonger, Hacktool.Spammer, Kraken
?		<a href="#">Torpig</a>	180,000 <sup>[26]</sup>	?	Sinowal, Anserin
?		<a href="#">Storm</a>	160,000 <sup>[27]</sup>	3	Nuwar, Peacomm, Zhelatin
2006 (Around)	2011 (March)	<a href="#">Rustock</a>	150,000 <sup>[28]</sup>	30	RKRustok, Costrat
?		<a href="#">Donbot</a>	125,000 <sup>[29]</sup>	0.8	Buzus, Bachsoy
2012 (Around)		<a href="#">Chameleon</a>	120,000 <sup>[30]</sup>	?	None
2008 (November)	<a href="#">2010 (March)</a>	<a href="#">Waledac</a>	80,000 <sup>[31]</sup>	1.5	Waled, Waledpak

# Služby firewallu a provisioning

Firewall typicky poskytuje služby nezbytné pro chod sítě:

- DNS (nebo DNS forwarding a cache)
  - Může provádět kontrolu podpisů DNSSEC
- DHCP
  - Přidělování IP adres a dalších parametrů stanicím
- SLAAC a DHCPv6
  - Obdoba DHCP pro svět IPv6
  - Snaha udělat to v IPv6 jednodušší vyústila ve 2 nezávislé protokoly, podstatně složitější než originál

# Služby firewallu a provisioning

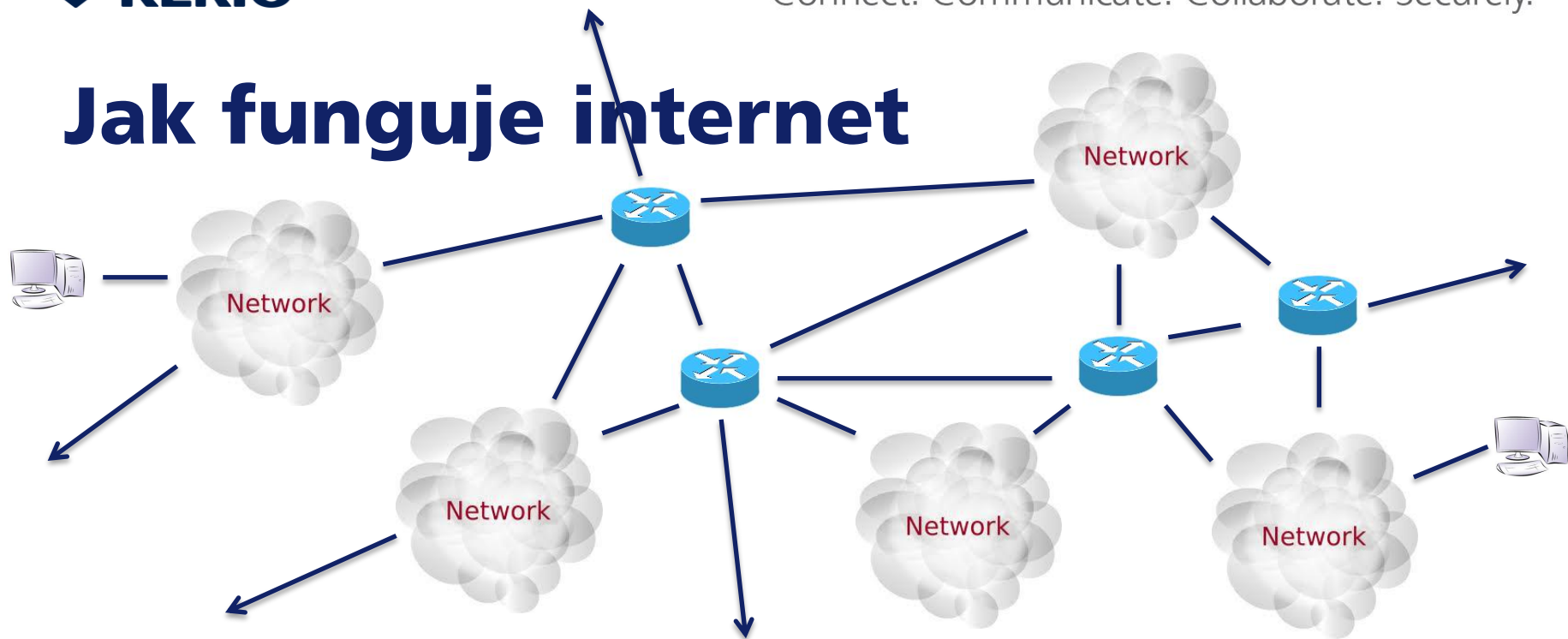
- SNMP
  - Vzdálený monitoring
- Proxy server
  - HTTP proxy server
  - SOCKS proxy server
  - Historická zátěž (a bude tu s námi asi navždy)
- Web server
  - Vzdálená správa
- Autentizace (radius server)
- WiFi

# Program přednášky

- Firewall: co to je / původ / dělení / historie
- Princip fungování / komponenty
- **Překlad adres (NAT)**
- Doplnkové funkce
- Obcházení firewallu / lidský faktor



# Jak funguje internet



- Internet spojuje obrovské množství jednotlivých menších sítí.
- Mezi jednotlivými sítěmi jsou tzv. páteřní routery.
- Každý router umí směřovat pakety mezi k němu připojenými sítěmi.
- Info kam má co směřovat si udržuje v tzv. směrovací tabulce.
- Tyto tabulky dosahují dnes velikosti přes 500 000 záznamů

# IP adresy

+/- 1990

- „Brzo dojdou IP adresy!“

1993

- Classless Inter-Domain Routing (CIDR)
- Do té doby se přidělovaly celé rozsahy tříd A, B a C:  
aaa.x.y.z, bbb.bbb.x.y, ccc.ccc.ccc.x

1996

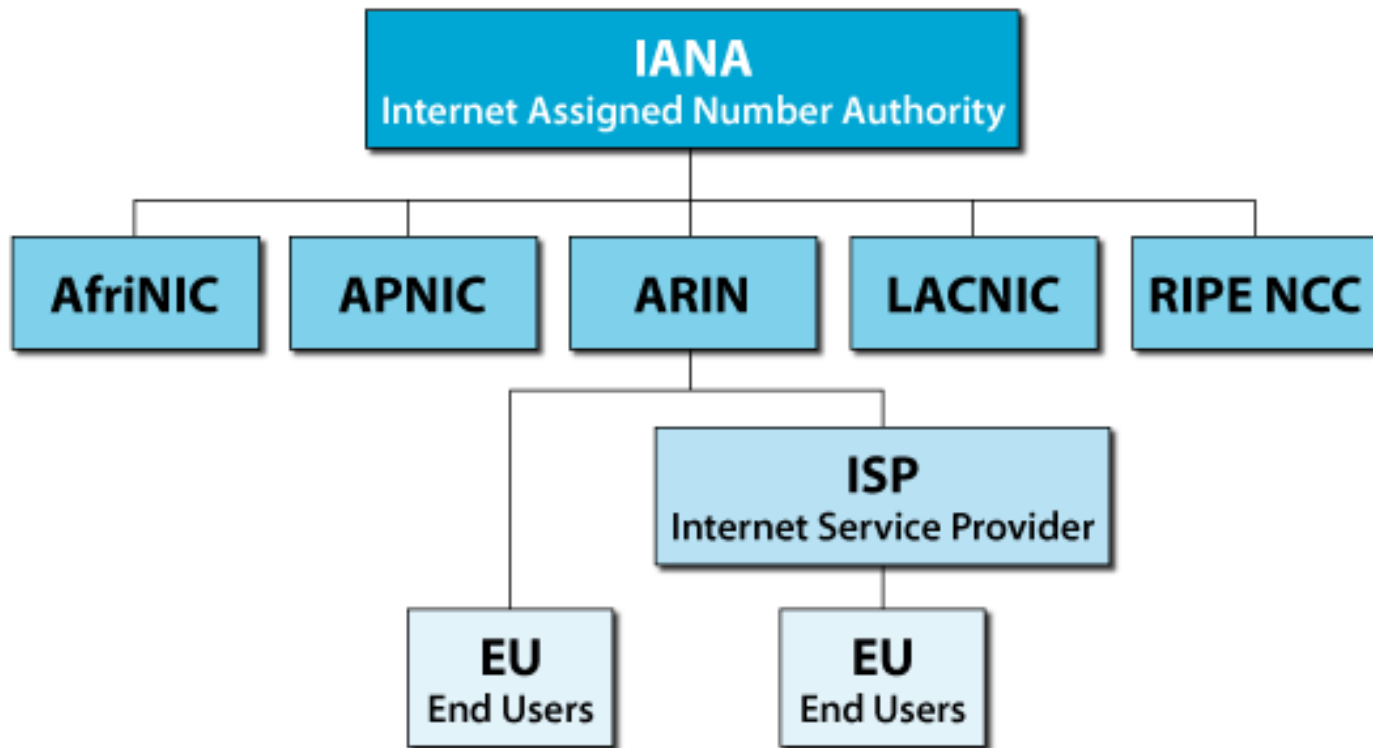
- „Privátní IP adresy“: 10.x.y.z / 172.16-31.x.y / 192.168.x.y

+/- 1998

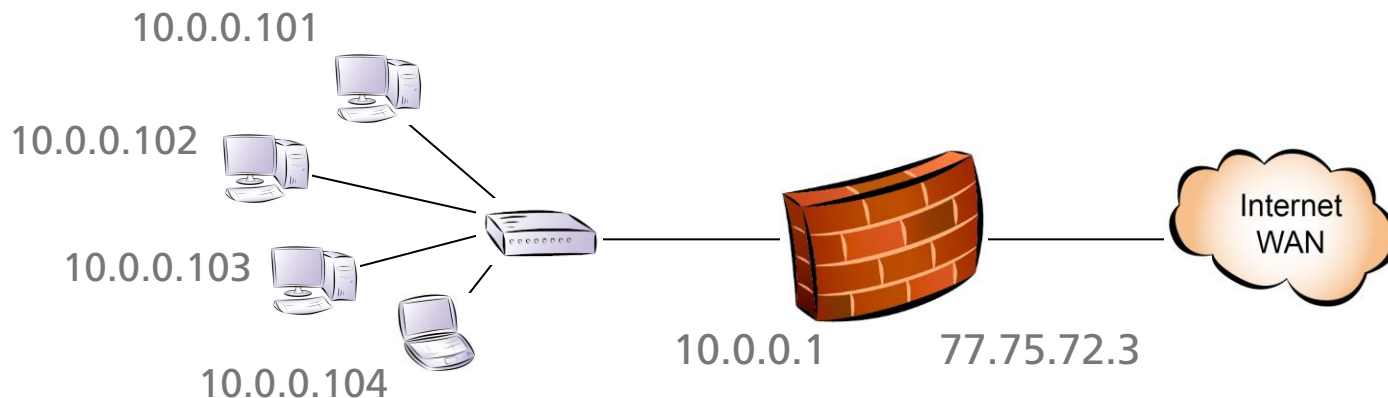
- Technologie NAT
- Počátek vývoje IPv6 protokolu

NAT ve své době výrazně zpomalil tempo vyčerpávání IPv4 adresního prostoru. Těžká rána pro nasazování IPv6.

# IP adresy



# Co je NAT



Umožňuje připojit k internetu celou síť přes 1 IP adresu.

Existuje více druhů:

- One-to-one
- One-to-many (jediná v prakticky používaná varianta)
- Kombinace

# Princip funkce NATu

Základem je tabulka spojení (flow table). Je to vlastně stavový paketový filtr.

Příklad flow table:

Zdroj (vnitřní)	Cíl (vnitřní)	Zdroj (vnější)	Cíl (vnější)	Protokol
10.0.0.101 : 4567	8.8.8.8 : 53	77.75.72.3 : 9876	8.8.8.8 : 53	UDP
10.0.0.102 : 1024	147.228.57.10 : 80	77.75.72.3 : 9877	147.228.57.10 : 80	TCP
10.0.0.103 : 1024	147.228.57.10 : 80	77.75.72.3 : 9878	147.228.57.10 : 80	TCP

Toto vidí klient v LAN

Toto vidí server v internetu

# Průchod paketu přes NAT směrem ven

1. Klient v LAN pošle paket na server, např.:
  - 10.0.0.102 : 1024 → 147.228.57.10 : 80
2. Firewall hledá záznam ve „vnitřní“ části flow tabulky.
3. Pokud jej nenalezne, vytvoří nový záznam:
  - Vnitřní část tabulky vyplní podle paketu
  - Vnější část tvoří vnější IP adresa firewallu a náhodně vygenerované (ale unikátní) číslo portu, cílová adresa bude na vnější straně stejná jako na vnitřní:

10.0.0.102 : 1024	147.228.57.10 : 80	77.75.72.3 : 9878	147.228.57.10 : 80	TCP
-------------------	--------------------	-------------------	--------------------	-----

4. Podle záznamu v tabulce se změní údaje v paketu a paket se odešle do internetu:
  - 77.75.72.3 : 9878 → 147.228.57.10 : 80

# Průchod paketu přes NAT zpět dovnitř

1. Server odpoví a pošle paket zpět na firewall, např.:
  - 147.228.57.10 : 80 → 77.75.72.3 : 9878
2. Firewall hledá záznam ve „vnější“ části flow tabulky.
3. Musí jej nalézt.
  - Pokud jej nenalezne, znamená to, že tento paket je tzv. nevyžádaný (unsolicited) a není možnost jak takový paket propustit, firewall nemůže vědět, kterému počítači v LAN jej přeposlat.

10.0.0.102 : 1024	147.228.57.10 : 80	77.75.72.3 : 9878	147.228.57.10 : 80	TCP
-------------------	--------------------	-------------------	--------------------	-----

4. Podle záznamu v tabulce se změní údaje v paketu a paket se odešle příslušnému počítači v LAN:
  - 147.228.57.10 : 80 → 10.0.0.102 : 1024

# Provozování serveru v LAN za NATem

- NAT nepropustí žádný paket z internetu do LAN, pro který neexistuje záznam ve flow tabulce.
- Co dělat, chceme-li ale provozovat např. web server?
- Řešení: **Port forwarding** (mapování portů, static NAT)

Ve flow tabulce vytvoříme pevný záznam:

Zdroj (vnitřní)	Cíl (vnitřní)	Zdroj (vnější)	Cíl (vnější)	Protokol
10.0.0.101 : 80	* : *	77.75.72.3 : 80	* : *	TCP

Z internetu pak přistupujeme na adresu firewallu, např.:

147.228.5.5 : 1024 → 77.75.72.3 : 80



# Provozování serveru v LAN za NATem

- NAT nepropustí žádný paket z internetu do LAN, pro který neexistuje záznam ve flow tabulce.
- Co dělat, chceme-li ale provozovat např. web server?
- Řešení: **Port forwarding** (mapování portů, static NAT)

Ve flow tabulce vytvoříme pevný záznam:

Zdroj (vnitřní)	Cíl (vnitřní)	Zdroj (vnější)	Cíl (vnější)	Protokol
10.0.0.101 : 80	* : *	77.75.72.3 : 80	* : *	TCP
10.0.0.101 : 80	147.228.5.5 : 1024	77.75.72.3 : 80	147.228.5.5 : 1024	TCP

Z internetu pak přistupujeme na adresu firewallu, např.:

147.228.5.5 : 1024 → 77.75.72.3 : 80

# Provozování serveru v LAN za NATem

- NAT nepropustí žádný paket z internetu do LAN, pro který neexistuje záznam ve flow tabulce.
- Co dělat, chceme-li ale provozovat např. web server?
- Řešení: **Port forwarding** (mapování portů, static NAT)

Ve flow tabulce vytvoříme pevný záznam:

Zdroj (vnitřní)	Cíl (vnitřní)	Zdroj (vnější)	Cíl (vnější)	Protokol
10.0.0.101 : 80	* : *	77.75.72.3 : 80	* : *	TCP
10.0.0.101 : 80	147.228.5.5 : 1024	77.75.72.3 : 80	147.228.5.5 : 1024	TCP
10.0.0.101 : 80	147.228.6.6 : 4567	77.75.72.3 : 80	147.228.6.6 : 4567	TCP

Z internetu pak přistupujeme na adresu firewallu, např.:

147.228.5.5 : 1024 → 77.75.72.3 : 80

# Problémy NATu

- NAT porušuje end-to-end konektivitu v internetu.
- Počítače za NATem jsou „méněcenné“, protože je nelze jednoduše přímo adresovat.
- Některé protokoly mají s NATem problémy (SIP, P2P, ale i obyčejné FTP)

# Přenos souboru protokolem FTP

Klient posílá ze své adresy 10.0.0.102 příkazy:

- `PORT 10,0,0,102,4,0`
- `RETR muj_bezva_soubor.dat`

Server vidí příkazy od klienta připojeného z adresy 77.75.72.3:

- `PORT 10,0,0,102,4,0`
- `RETR muj_bezva_soubor.dat`

Přenos souboru musí zákonitě selhat, protože server se na adresu 10.0.0.102 neumí připojit.

# Application-Level Gateway (ALG)

- Firewall kromě hlaviček mění i obsah paketů.
- Dělá NAT i na aplikační vrstvě.

## ALG pro FTP:

- Hledá příkazy „PORT“ a mění zasílanou IP adresu
- Zároveň předpřipraví záznam ve flow tabulce, aby byl přenos souboru umožněn

## Nevýhody ALG:

- Mohou citelně snížit výkon
- Pro složitější protokoly (VoIP) příliš složité a často nefunkční
- Nelze je použít pro šifrované přenosy

# Přenos souboru protokolem FTP s ALG

Klient posílá ze své adresy 10.0.0.102 příkazy:

- `PORT 10,0,0,102,4,0`
- `RETR muj_bezva_soubor.dat`

Firewall připraví záznam v tabulce a změní adresu v příkazu:

10.0.0.102 : 1024	147.228.57.10 : *	77.75.72.3 : 2048	147.228.57.10 : *	TCP
-------------------	-------------------	-------------------	-------------------	-----

Server vidí příkazy od klienta připojeného z adresy 77.75.72.3:

- `PORT 77,75,72,3,8,0`
- `RETR muj_bezva_soubor.dat`

# NAT Traversal

Lepší než ALG je přímá podpora v aplikačním protokolu

Příklady:

IPsec NAT-T

- Detekuje přítomnost NATu a přizpůsobí se
- Místo posílání přímo IPsec paketů je „balí“ do UDP datagramů

Instant messaging

- Komunikuje přes prostředníka (server)
- Spojení jsou fakticky vždycky směrem od klienta

Většina novějších protokolů podporu pro NAT má.

# STUN

STUN = Simple Traversal for UDP through NAT

Používaný v IP telefonii. Na stejném principu je ale založena podpora ve většině protokolů.

Princip:

Jsem za NATem a chci komunikovat s přítelem v internetu:

1. Pošlu z nějakého svého portu „průzkumný“ paket na STUN server
2. Server mi odpoví a dá vědět, na jakou vnější IP adresu a port byla moje zpráva změněna.
3. Pošlu ze stejného svého portu paket příteli v internetu a spolehnu se na to, že bude firewallem změněn stejně
4. Tím vytvořím záznam ve flow tabulce a přítel z internetu se mnou může na tomto portu komunikovat přímo



# Další aspekty NATu

Typy NATu:

- Symetrický vs. trychtýřový (tzv. Cone NAT)
- Ovlivňuje schopnost NATu podporovat STUN

NAT Hairpin

- Hairpin = vlásenka
- Umožňuje připojení počítačům z vnitřní sítě zpět do vlastní sítě přes veřejnou adresu firewallu
- Bez této funkce nelze přistoupit na server ve vnitřní síti z počítačů ve stejné síti

# NAT a IPv6

„Lidé od IPv6“ nahlížení na NAT s despektem a těší se na okamžik, kdy IPv6 plně nahradí IPv4.

NAT má však i své přínosy:

- Zvyšuje zabezpečení sítě (implicitně funguje jako stavový paketový filtr)
- Výrazně zjednodušuje správu IP adres v síti
- Přejít k jinému poskytovateli internetu neznamena (na rozdíl od IPv6), že musím přechíslovat všechny počítače v síti
- Velmi snadno mohu dosáhnout balancování zátěže mezi více poskytovateli (mohu pro každé flow zvolit jiného)
- IPv6 s těmito funkcemi stále (poněkud neúspěšně) zápasí

# Program přednášky

- Firewall: co to je / původ / dělení / historie
- Princip fungování / komponenty
- Překlad adres (NAT)
- **Doplňkové funkce**
- Obcházení firewallu / lidský faktor

# Bez čeho se firewall neobejde

Firewally dávno neplní prostou funkci filtrování. Mezi další funkce patří:

- Antivirová kontrola přenášených souborů
- Filtrování SPAMu
- VPN
- Kategorizace webů, řízení produktivity
- Statistiky provozu (sledování chování uživatelů)
- Rozložení zátěže provozu mezi více poskytovateli
- Traffic shaping (řízení šířky pásma)

# Antivirová kontrola

## Problémy:

- Šifrování
- Transparentnost kontroly
- Archivy (ZIP, RAR, ...)
- Výkon

## Transparentnost

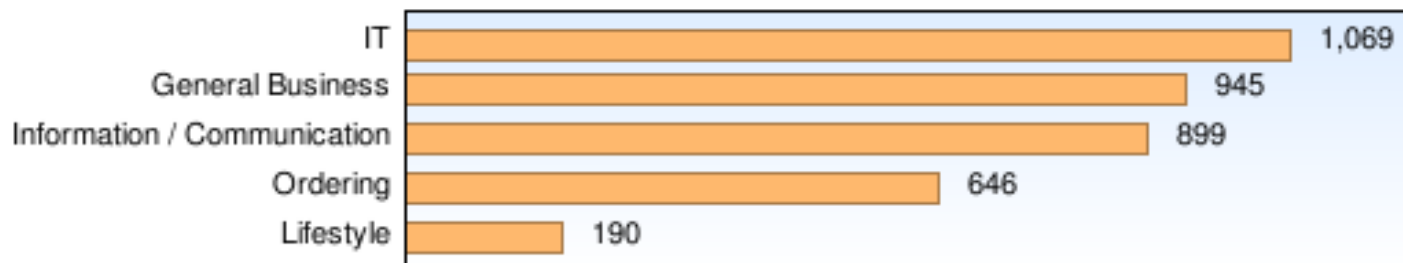
- Soubor lze zkontrolovat až když je celý
- Během stahování firewall propuští soubor na klienta
- Posledních několik KB pošle, až když provede kontrolu
- V případě napadení souboru skončí stahování chybou
- Neexistuje způsob jak klienta na virus upozornit

# Webový filtr

„Cloudová“ kategorizace web stránek.

- Firewall se ptá dedikovaného serveru na kategorii každé stránky kterou uživatel navštíví
- V databázi jsou kategorie stránek (kategorizace probíhá automaticky i ručně)
- Na základě navštívených kategorií lze pak blokovat, logovat, vytvářet statistiky, atp.

## • Top Requested Web Categories



# Statistiky

Kerio StaR Overall  My Account Logout

Today | This Week | This Month | Custom period | Print

The latest data update: 5/1/2012 02:43:10 pm

5/1/2012

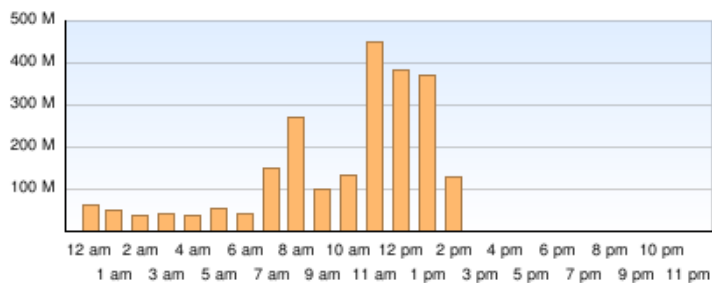
- Overall
- Individual
- Users' Activity
- Users by Traffic
- Visited Sites
- Web Categories

## Hourly Traffic

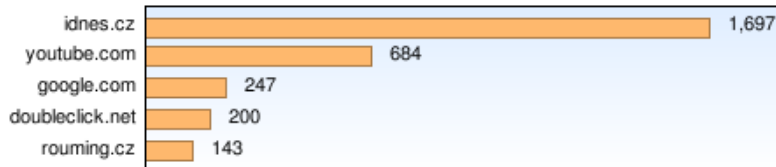
**Total:**  
2,401,214 KB

**Inbound:**  
1,931,351 KB

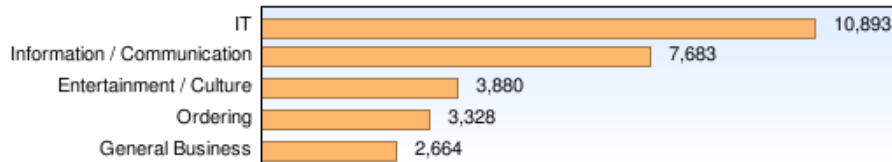
**Outbound:**  
469,862 KB



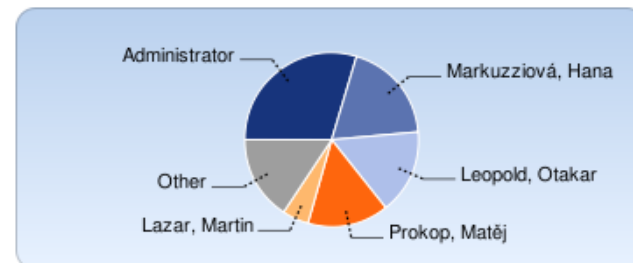
## Top Visited Websites



## Top Requested Web Categories

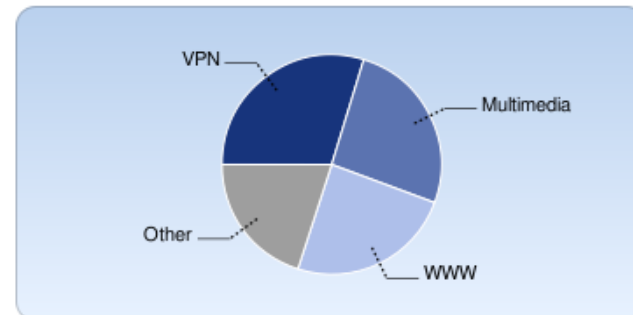


## Top 5 users



User	Total [KB]
Administrator (admin)	709,594
Markuzziová, Hana (hmarkuzziova@kerio.local)	455,617
Leopold, Otakar (oleopold@kerio.local)	377,994
Prokop, Matěj (mprokop@kerio.local)	362,967
Lazar, Martin (mlazar@kerio.local)	118,678

## Used Protocols



# Statistiky


[Click here to find out more!](#)

01:29 pm	iradio.cz   Visits: 1   Categories: Information / Communication <a href="#">Doména iradio.cz je registrovaná na CZECHIA.COM</a>
01:29 pm	7:18 rozhlas.cz   Visits: 9   Categories: Entertainment / Culture, Information / Communication <a href="#">Český rozhlas</a>
01:33 pm	0:35 ihned.cz   Visits: 3   Categories: Information / Communication <a href="#">IHNEDE.cz : Zpravodajský server Hospodářských novin</a>
01:35 pm	idnes.cz   Visits: 1   Categories: Information / Communication <a href="#">Klikni.cz &gt; E-mailová schránka &gt; Login</a>
01:35 pm	google.cz Searched for: <a href="#">práce programátor javascript</a>
01:35 pm	0:16 careerjet.cz   Visits: 2   Categories: Job Search <a href="#">Nabídky práce - javascript - Česká republika   careerjet.cz</a>
01:36 pm	0:01 empleo.cz   Visits: 1   Categories: Job Search <a href="#">Nabídka práce: Javascript Developer - Plzeň - Kerio Technologies s.r.o.   empleo.cz</a>
01:36 pm	jobs.cz   Visits: 1   Categories: Job Search <a href="#">Jobs.cz - Spojení s elitou - nabídka práce, volná pracovní místa i brigády</a>
01:37 pm	0:13 sencha.com   Visits: 2   Categories: IT <a href="#">HTML5 Framework for Desktop and Mobile Devices. Build HTML5 Apps for Any Browser.   Sencha</a>
01:37 pm	vimeo.com   Visits: 1   Categories: Entertainment / Culture <a href="#">Sencha Architect 2 Launch</a>



## Large File Transfers

Files: 4 | Data transferred: 80,073 KB  
P2P activity not detected

[Hide details](#) 

Start	Duration	Details
01:16 pm	0:03	Download from cdn-cache5g.seznam.cz 17,530 KB <a href="#">1789751-VcWaOB.mp4</a>
01:18 pm	0:01	Download from cdn-cache4n.seznam.cz 9,540 KB <a href="#">1789701-S1B85w.mp4</a>
01:23 pm	0:02	Download from cdn-cache4g.seznam.cz 3,770 KB <a href="#">1789605-P1HjS3.mp4</a>
01:38 pm	0:05	Download from cdn.sencha.io 49,231 KB <a href="#">ext-4.1.0-qpl.zip</a>



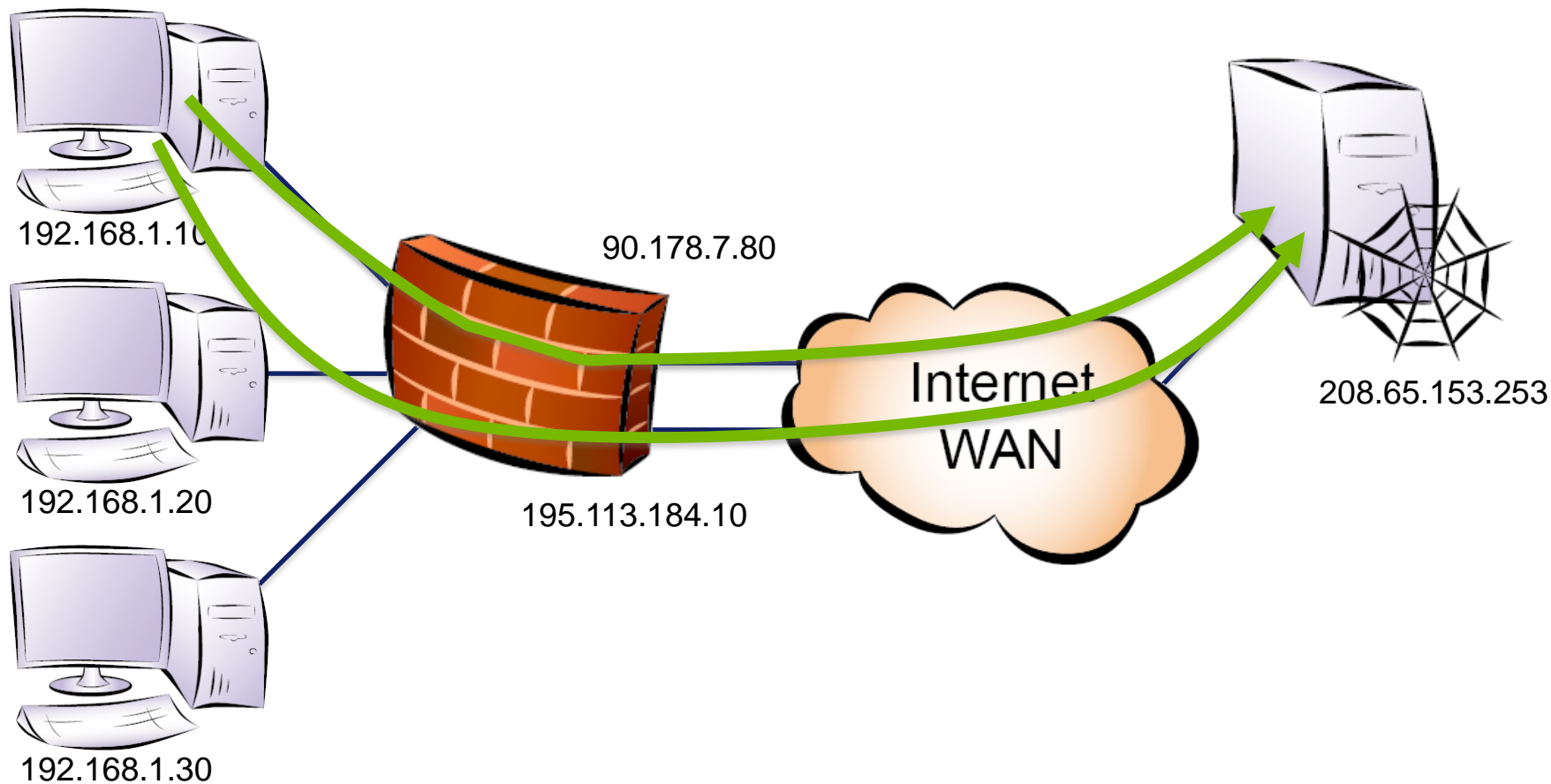
# Rozložení zátěže (Load Balancing)

Firewall dělá NAT, tj. dává do paketů veřejnou IP adresu. Máme-li více internetových linek, můžeme mezi ně rozložit zátěž.

## Load Balancing

- Odchozí pakety můžeme poslat kterou linkou chceme.
- Vhodnou volbou veřejné IP adresy při NATu docílíme, že i příchozí provoz pojedou přes danou linku.
- Budeme sledovat zatížení jednotlivých linek.
- Pro každé nové spojení (flow) se znovu rozhodneme, jakou veřejnou IP adresu použije, podle aktuálního zatížení linek.

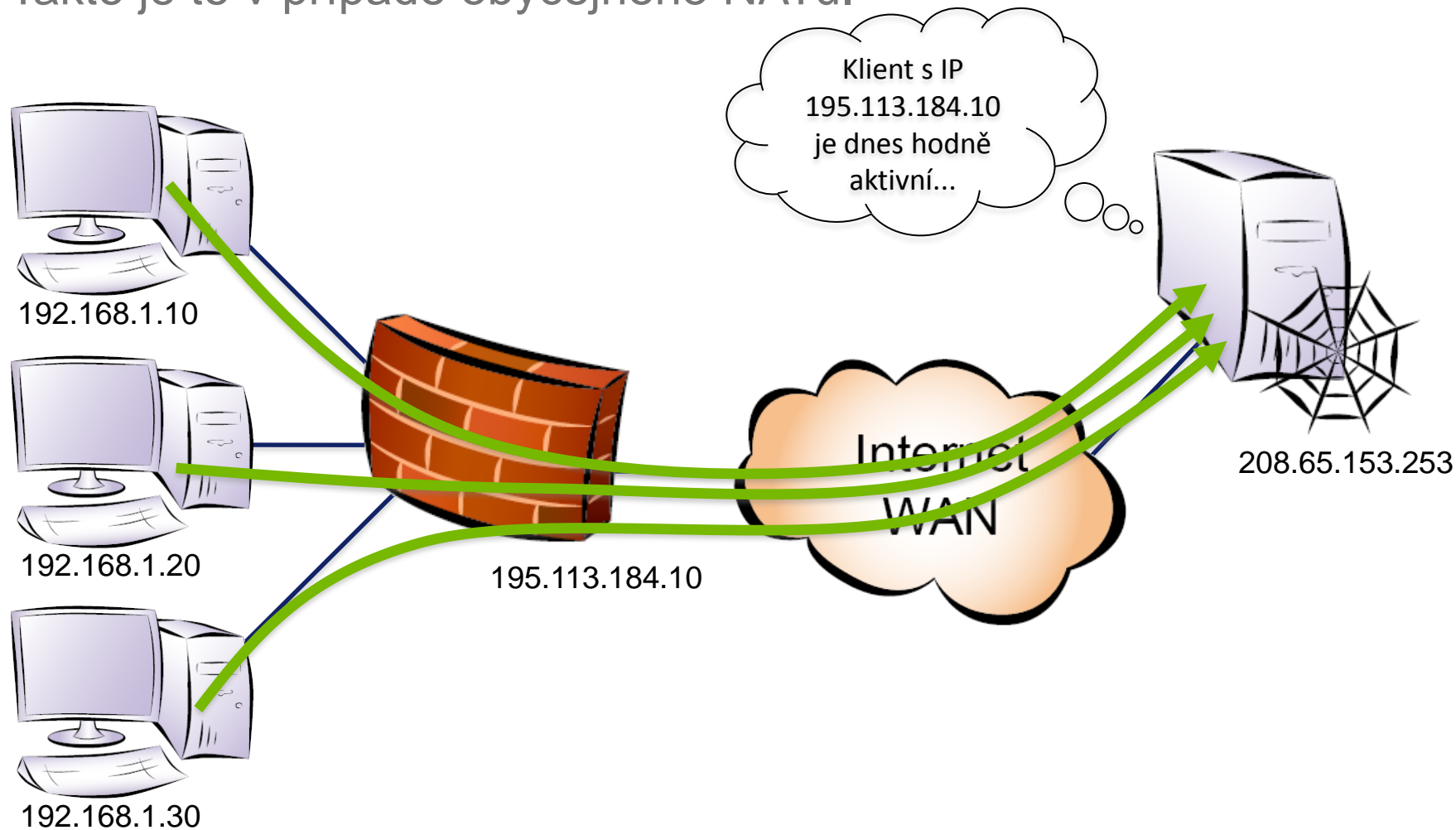
# Rozložení zátěže (Load Balancing)



**Pozor, není to tak jednoduché!**

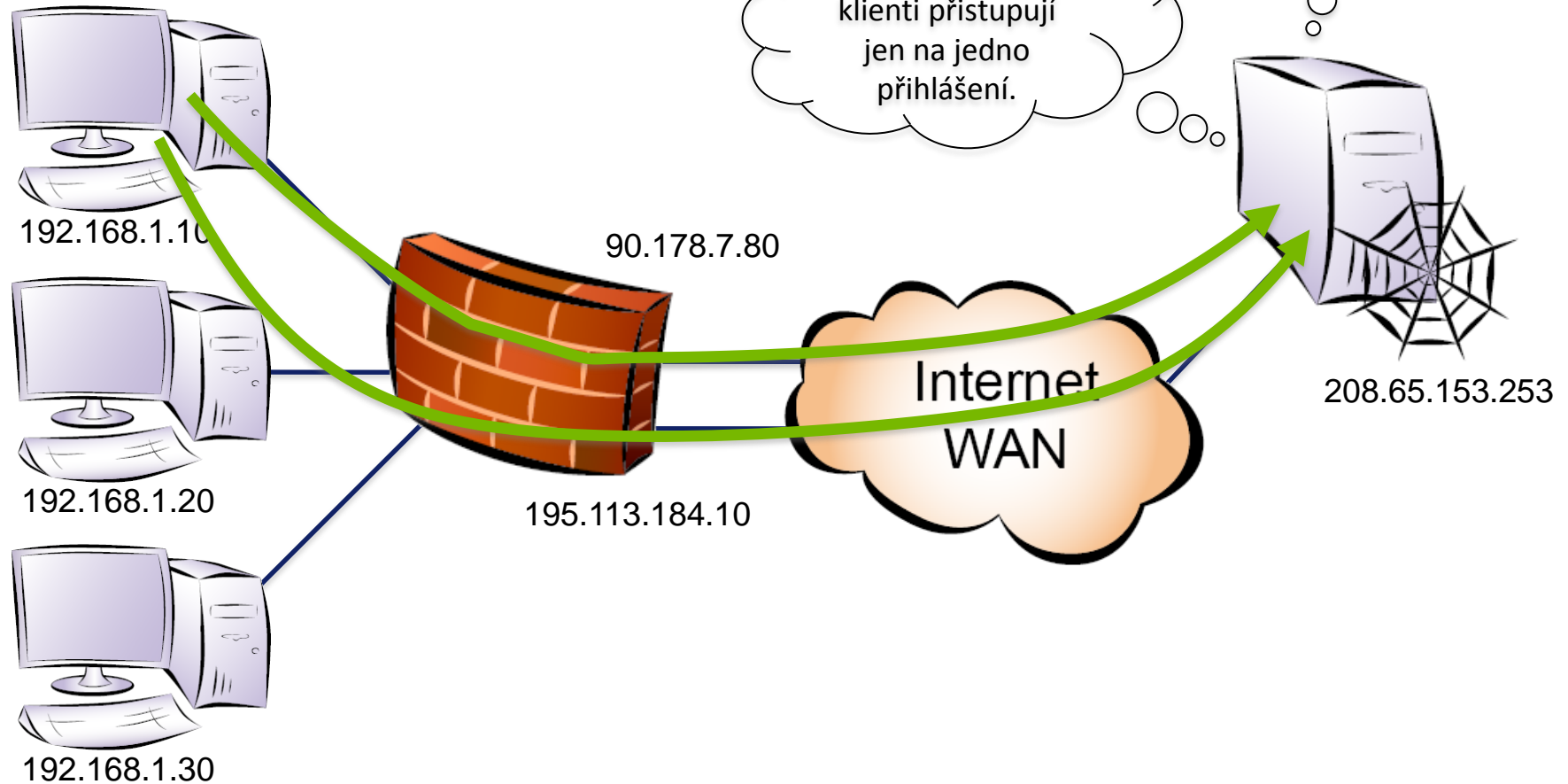
# Load Balancing z pohledu serveru

Takto je to v případě obyčejného NATu.

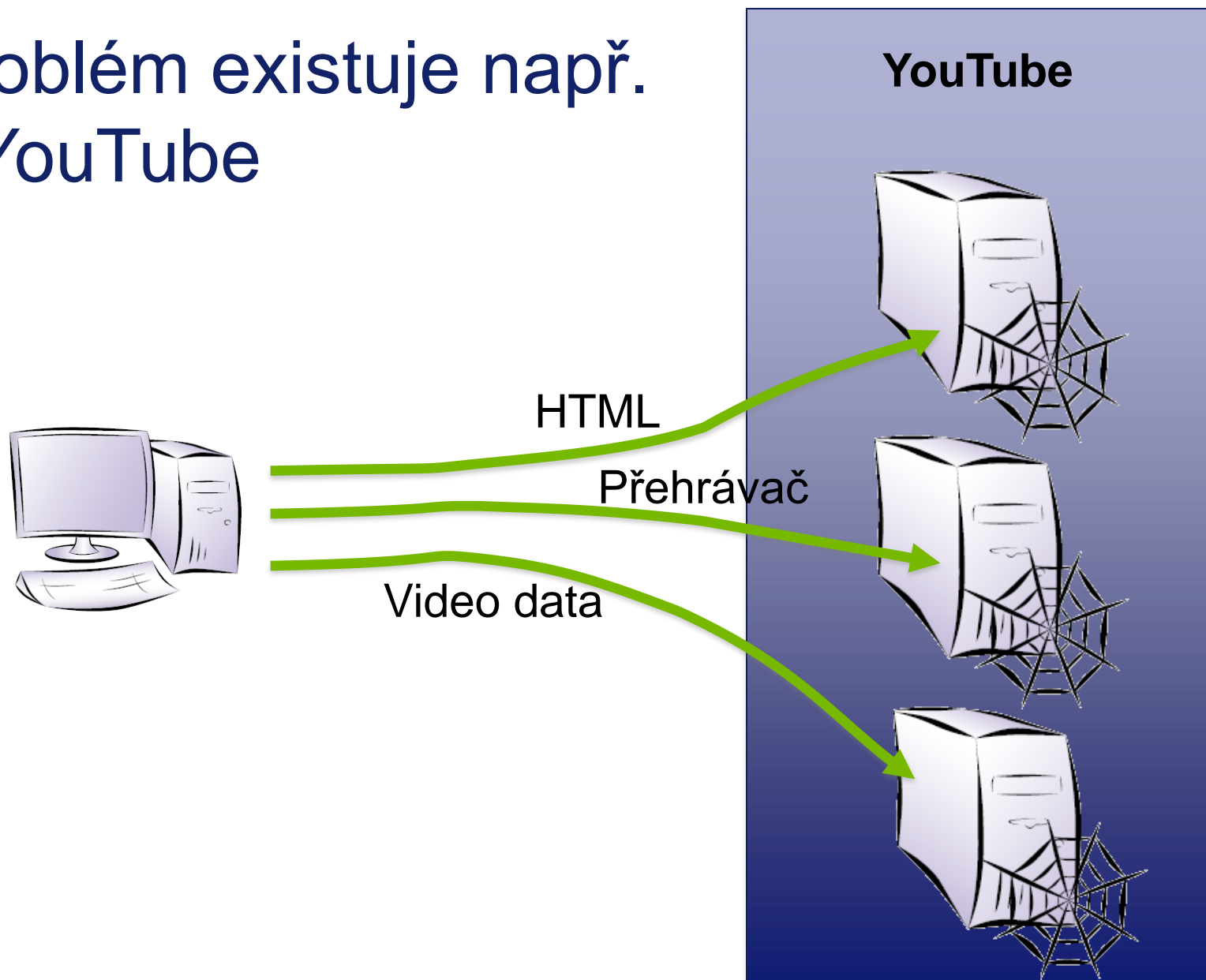


# Load Balancing z pohledu servere

Load balancing ale působí problémy



# Problém existuje např. s YouTube



# Traffic shaping

## Omezování rychlosti

- Paketový filtr každý paket patřící do omezované skupiny na chvíli pozdrží
- Simuluje tak pomalejší přenos
- TCP spojení se přizpůsobí
- UDP většinou nakonec také díky vyššímu protokolu

## Garantování rychlosti

- V podstatě omezení „toho ostatního“
- Garantovat šířku pásma lze pouze jsme-li nejužší místo na lince
- Firewall musí znát skutečnou rychlost linky, ke které je připojen

# Program přednášky

- Firewall: co to je / původ / dělení / historie
- Princip fungování / komponenty
- Překlad adres (NAT)
- Doplnkové funkce
- **Obcházení firewallu / lidský faktor**

# Obcházení firewallu

Dobré obcházení

- STUN
- VoIP

Zlé

- Viry, spam, spyware, obecně malware
- Anonymizery
- P2P

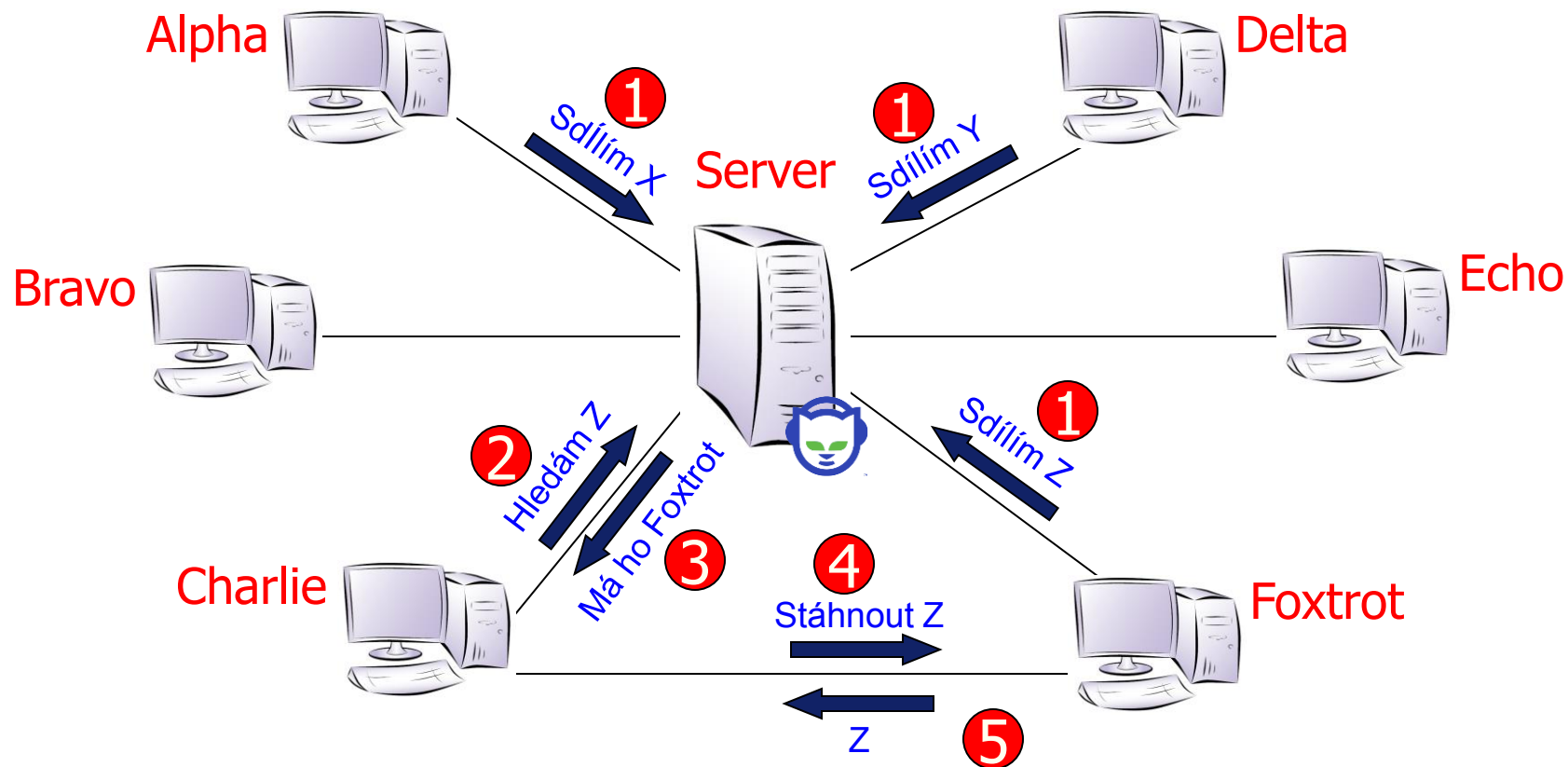
Na pomezí

- Skype



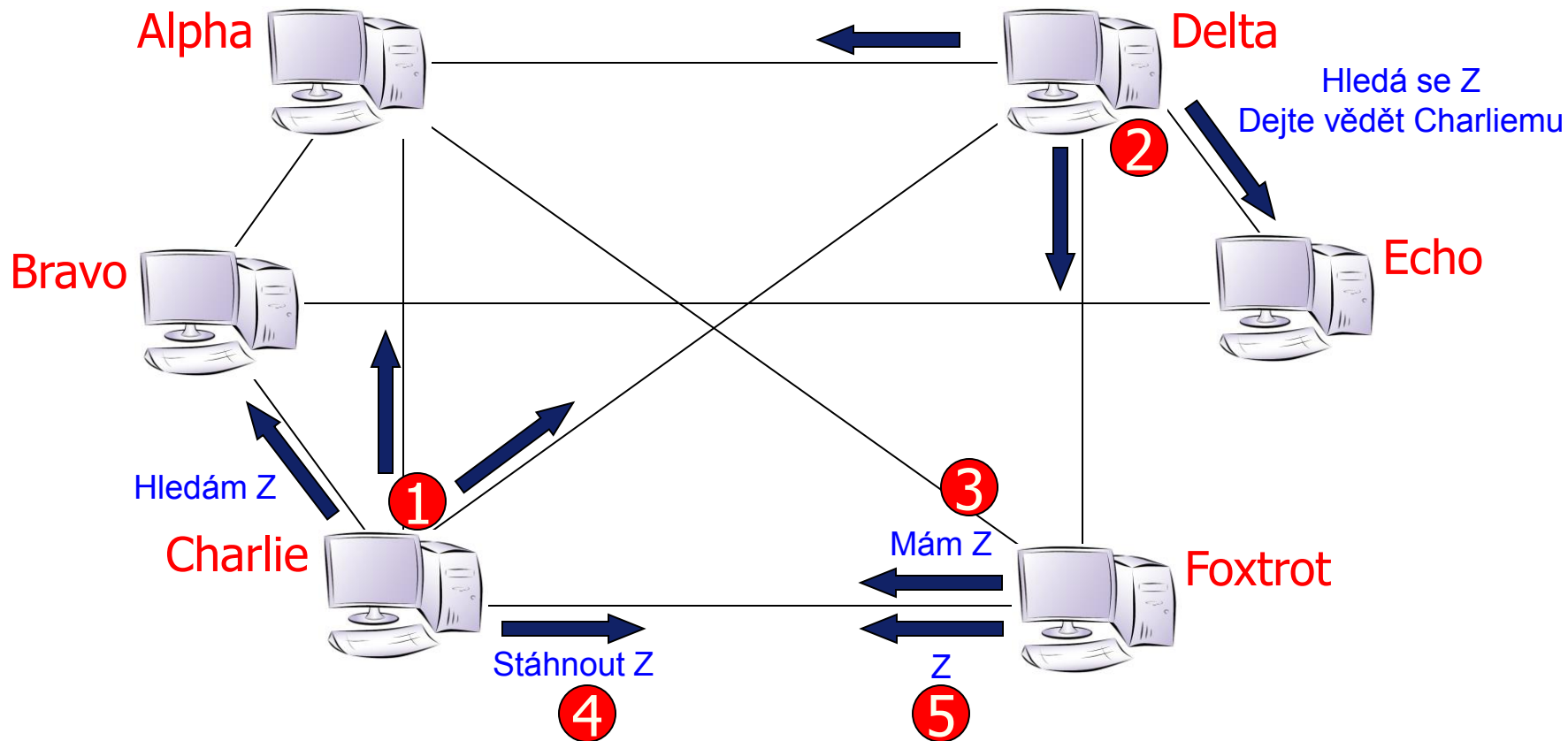
# Evoluce P2P sítí

## 1. Centralizované



# Evoluce P2P sítí

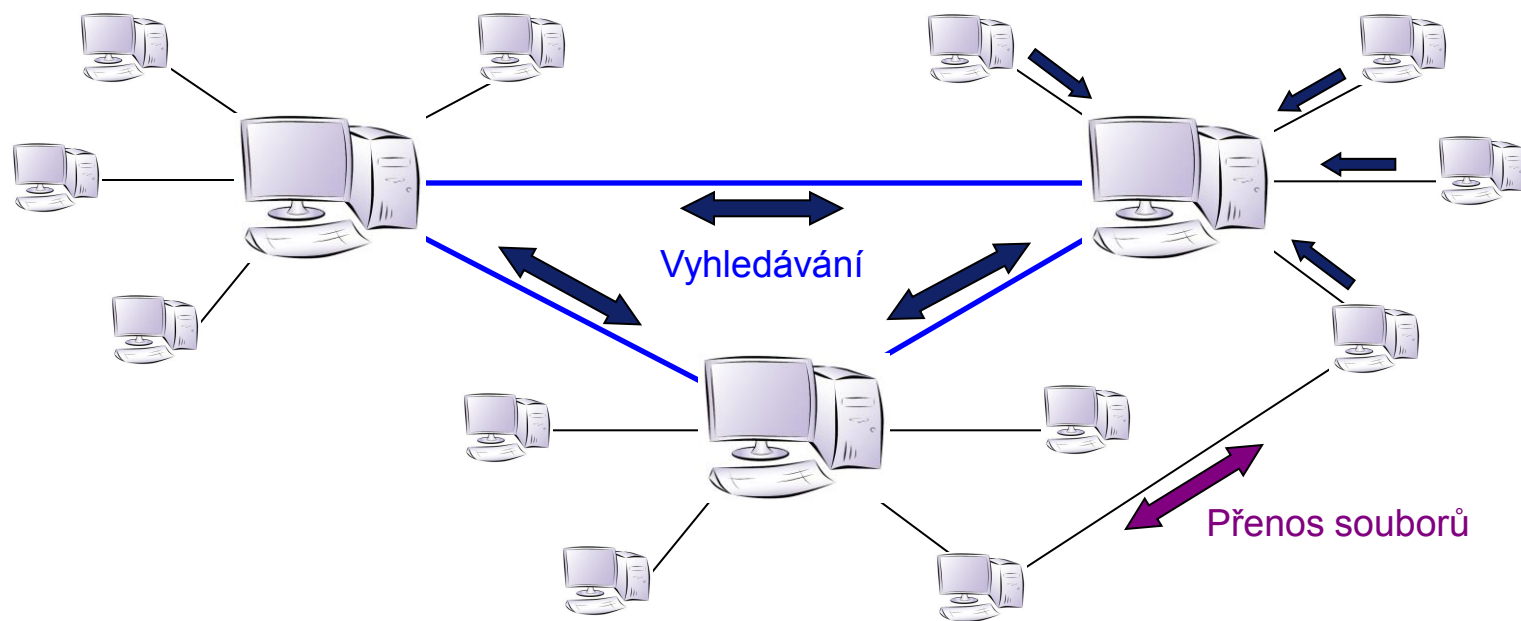
## 2. Decentralizované sítě



# Evoluce P2P sítí

## 3. Hybridní sítě

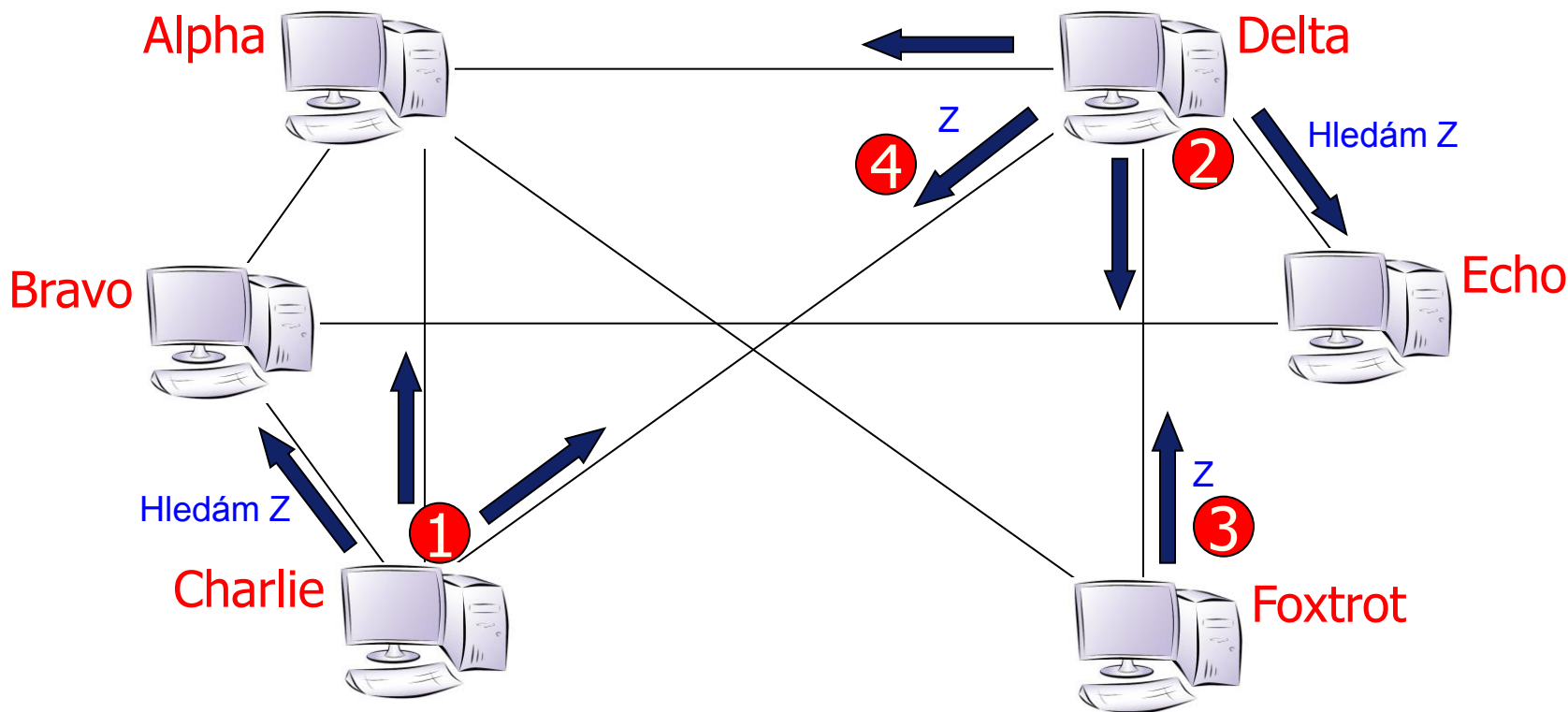
Dnes převládají: Skype, BitTorrent



# Evoluce P2P sítí

## 4. Proxy síť

Pomalé, neříliš spolehlivé, ale zcela anonymní.



# Jak P2P obchází firewall

Sítě s fixním číslem portu nijak

- Blokování je pak triviální
- Prakticky neexistují

Sítě s náhodným číslem portu

- Převládá-li některé číslo portu, lze to použít pro blokování
- Jinak velmi obtížně blokovatelné, nechci-li příliš omezit uživatele
- Lze nasadit analýzu obsahu paketu

Sítě s šifrováním nebo pseudošifrováním (obfuscation)

- Téměř neblokovatelné
- Statistická analýza provozu či adaptivní blokování
- Nebo FUP

# Adaptivní blokování

Použijeme proti P2P její vlastní zbraň – masovost.

- Jedna instance P2P typicky naváže desítky, spíše stovky až tisíce spojení (už to je podezřelé)
- Mnoho sítí sice používá náhodný port, ale některé klientské aplikace zkouší nejprve vždy stejný port.
- Algoritmus jednoduchého adaptivního blokování:
  1. Nejprve neblokujeme nic
  2. Počítáme si počet „podezřelých“ spojení
  3. Jak počet roste, začínáme blokovat víc a víc
  4. Nakonec propouštíme jen ne-P2P provoz
- Algoritmus nefunguje zcela na 100%, ale na významné omezení P2P zcela postačuje

# Lidský faktor

Zásady nastavení firewallu:

1. Zablokovat vše
2. Zjistit si, co je nutně potřeba povolit a to povolit
3. Provádět pravidelnou kontrolu politiky

Ale pozor:

- Čím restriktivnější politika, tím větší tendence lidí ji obcházet
- Zakážete Facebook a lidi začnou používat různé proxy anonymizéry
- Zakážete i ty a lidi se připojí přes mobil s internetovým tarifem
- Důležitější je vědět co se na síti děje než blokovat bezmyšlenkovitě všem všechno

# Děkuji

Jan Ježek, [jjezek@samepage.io](mailto:jjezek@samepage.io)