titulní strana

zadání

licenční smlouva jedna

licenční smlouva jedna

ANOTACE

Počítačové sítě se v současnosti používají mnohem více než před dvaceti lety. Počítače jsou využívány nejčastěji jako prostředek pro komunikaci, zábavu a skladování dat. Informace jsou dnes často uloženy pouze v elektronické podobě a proto je velice důležité jejich bezpečné skladování. Cílem této diplomové práce je popsat problematiku zabezpečení počítačových sítí.

První část práce je věnovaná popisu zabezpečení počítačové sítě a jeho kontrolování. Jsou zde diskutovány postupy, které se v praxi používají při kontrole zabezpečení i útocích na síť. Úvodní část popisuje možnosti skenování a filtrování dat v síti na jednotlivých vrstvách síťového modelu TCP/IP. Druhá část prezentuje jednotlivé druhy proxy spolu s jejich výhodami a nevýhodami. Dále je popsán Network Address Translation (NAT), který je jedním ze způsobů jak ochránit vnitřní síť a navíc efektivně nakládat s IP adresami. Na závěr této kapitoly je uveden stručný popis IPSec, VPN a základní typy útoků na počítačovou síť.

V druhé kapitole je uveden popis skriptů v programovacím jazyku Perl, které kontrolují některé oblasti zabezpečení počítačové sítě. Účelem skriptů není kontrolovat kompletní zabezpečení počítačové sítě, ale jsou navrženy tak, aby vyhověly aktuálním požadavkům IBM Global Services Delivery Centrum Brno. První ze skriptů skenuje spuštěné aplikace na síťových zařízeních. Jeho účelem je detekovat běh nepotřebných nebo neaktualizovaných aplikací. Druhý skript zjišťuje správnost konfigurace Cisco směrovačů s ohledem na předem definovaný seznam pravidel. Třetí ze skriptů načítá nastavení Nokia firewallu, který je na rozhraní vnitřní a vnější sítě IBM, a zjišťuje, zda je jeho konfigurace bezpečná. Výstupem prvních dvou skriptů je přehledný HTML dokument, poslední skript vypisuje všechny důležitá data na příkazové řádce.

V poslední části práce jsou uvedeny rady při konfiguraci Cisco síťových zařízení. Je zde uveden seznam bezpečnostních doporučení, která mohou být použita např. při konfiguraci směrovačů.

Součástí přílohy jsou dvě laboratorní úlohy, ve kterých mají studenti příležitost k seznámení s programy a postupy, které jsou používány v praxi IT odborníky pro kontrolu slabých míst jejich sítí.

Klíčová slova: zabezpečení počítačových sítí, proxy, firewall, hacking, Perl

ABSTRACT

Computer networks are used in much wider extent than 20 years ago. People use the computer mainly for communication, entertainment and data storage. Information is often stored only in electronic devices and that is why the security of the data is so important. The objective of my thesis is to describe network security problems and their solutions.

First chapter deals with the network security, security checks and attacks. It describes procedures used in practise. First part deals with traffic scanning and filtering at various layers of the TCP/IP model. Second part presents the types of proxy and its pros and cons. Network Address Translation (NAT) is a favourite technique of managing IP addresses of inside and outside network which helps to improve the security and lower the costs paid for IP addresses. NAT description, IPSec, VPN and basic attacks are described in this section.

The second chapter of the thesis presents set of Perl scripts for network security checking. The purpose of the project is not to check the whole network security. It is designed for contemporary needs of IBM Global Services Delivery Centrum Brno. The first script checks running applications on target object. The aim is to detect services that are not necessary to run or that are not updated. The second one checks the security of the Cisco device configuration. There is a list of rules that has to be kept. The third script inspects the Nokia firewall configuration which is on the border of IBM network. If some of the rule is broken, it shows the command that has to be proceeded at the particular device. The output of the first and the second script is an HTML file. The third script uses the command line for the final report.

The last part of this chapter gives advice to configure Cisco devices. It is a list of security recommendations that can be used by configuring e.g. routers.

The appendix presents two laboratory exercises. The aim is to give students an opportunity to learn something about programs and technologies which are used in practise by IT experts to check the weaknesses of their networks.

Keywords: network security, proxy, firewall, hacking, Perl

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Aplikace pro monitorování a kontrolu zabezpečení rozsáhlých počítačových sítí LAN a WAN“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne …… ……..
(podpis autora)

Poděkování

Na tomto místě bych chtěl poděkovat především vedoucímu své diplomové práce doc. Ing. Vítu Novotnému, Ph.D. a společnosti IBM Global Services Delivery Centrum Brno za všechen čas, který mi věnovali, za podnětné rady a připomínky k této práci.

Zkratky

ACL ..... Access Control List
AP ..... Access Point
ARIN ..... American Registry for Internet Numbers
BIND ..... Berkeley Internet Name Domain
DMZ ..... Demilitarizovaná zóna
FTP ..... File Transfer Protocol
ICMP ..... Internet Control Message Protocol
IDS ..... Intrusion Detection System
IMAP4 ..... Internet Message Access Protocol
LDAP ..... Lightweight Directory Access Protocol
NAT ..... Network Address Translation
NIDS ..... Network Intrusion Detection System
POP3 ..... Post Office Protocol version 3
PPTP ..... Point-to-Point Tunnelling Protocol
RIPE ..... Réseaux IP Européens
SMTP ..... Simple Mail Transfer Protocol
SSH ..... Secure Shell
SSL ..... Transport Layer Security
TCP ..... Transmission Control Protocol
TLS ..... Transport Layer Security
UDP ..... User Datagram Protocol
VPN ..... Virtual Private Network
HTTP ..... Hyper Text Transfer Protocol
ARP ..... Address Resolution Protocol
PAT ..... Port Address Translation
DHCP ..... Dynamic Host Configuration Protocol
SNMP ..... Simple Network Management Protocol

Obsah

Úvod

Počítačové sítě se staly v posledních letech nejdůležitějším komunikačním prostředkem i úložištěm dat. S rostoucím počtem uživatelů se zvyšují nejen požadavky na správný chod sítě, ale také na její bezpečnost. Většina citlivých firemních údajů je uložena v elektronické podobě na intranetu společnosti, což sebou nese velké požadavky na dostatečné zabezpečení. Správci sítí mají za úkol zajistit nejen bezchybný chod sítě, ale také bezpečnost uložených dat. Každý z uživatelů má určitá přístupová práva, která definují rozsah činnosti, ke které jsou oprávněni. Pokud by někdo mohl zacházet s daty, k nimž by neměl mít přístup, jedná se o chybu v bezpečnostní politice.

V diplomové práci je rozebírána problematika bezpečnosti počítačových sítí a možnosti jejího prověřování. První část je věnována možnostem skenování komunikace v počítačové síti, která je často využívaná nejen administrátory, ale i hackery. Lze tak zjistit, které části sítě nejsou dostatečně zabezpečené a sjednat příslušnou nápravu.

Následující část je věnována možnostem filtrování komunikace. Přenos dat mezi vnitřní a vnější sítí by měl být řízen nějakým systémem, který kontroluje oprávněnost přístupu do intranetu. V praxi existuje několik technických realizací, které tuto činnost zabezpečují. V kapitole jsou diskutována jak jednoduchá, tak složitá řešení. Liší se hardwarovou i softwarovou složitostí, náročností konfigurace a samozřejmě také cenou.

V praktické části práce jsem se věnoval tvorbě skriptů, které budou kontrolovat úroveň zabezpečení počítačové sítě. Projekt byl vytvářen pro IBM Global Services Delivery Centrum Brno, kde by měl zjednodušit v současnosti prováděné postupy. Nyní je kontrola prováděna v různých systémech a vyžaduje příliš mnoho úkonů od obsluhy. Navrhovaný systém by měl tuto práci zautomatizovat, aby byl počet úkonů obsluhy minimální.

Ve skriptech jsou kontrolovány například běžící služby na cílovém objektu, což představuje účinnou prevenci proti zbytečně spuštěným aplikacím, které by mohly představovat nebezpečí pro systém.

Počítačová síť je tvořená velkým množstvím směrovačů a přepínačů, které mají různé konfigurace. I přes odlišnost jejich nastavení je třeba kontrolovat určitá nastavení, která by za určitých okolností mohla představovat možnou hrozbu pro bezpečnost systému. Ve skriptech jsem se věnoval kontrole konfigurace směrovačů Cisco a firewallu od společnosti Nokia.

Na závěr své práce jsem uvedl dvě laboratorní úlohy, které slouží studentům k lepšímu pochopení problematiky zabezpečení. První z nich se věnuje možnostem skenování sítě a vytváření umělých systémů v programech honeyd a nmap.

V druhé úloze si studenti vyzkoušejí konfiguraci firewallu Cisco PIX 501 a tím budou mít možnost vyzkoušet v praxi často používaný nástroj pro zabezpečení.

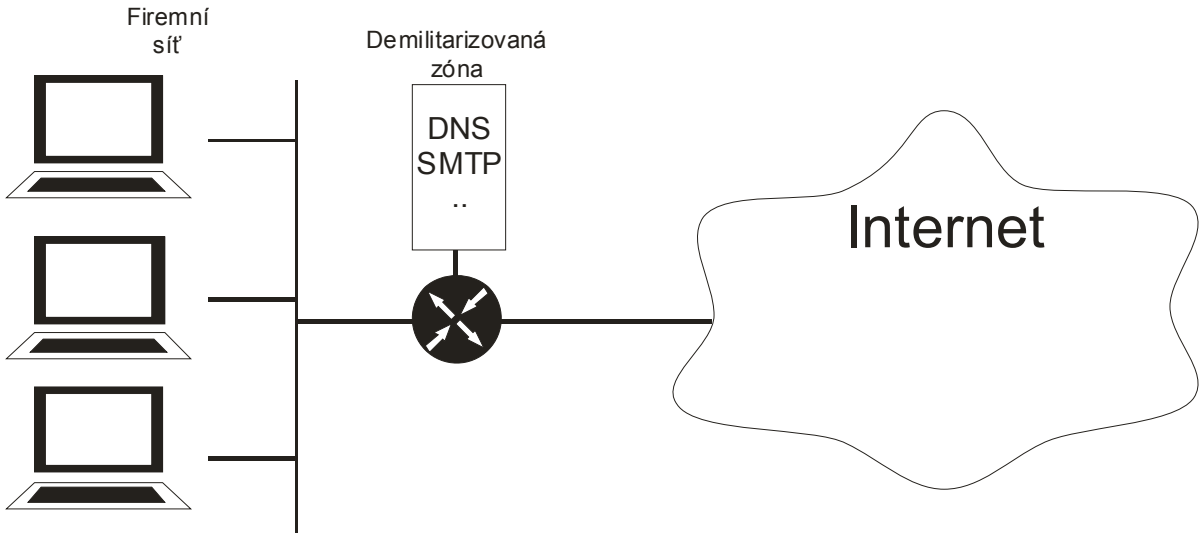
Požadavky na počítačové sítě se v posledních letech neustále zvyšují. Rychlost připojení do LAN i Internetu je dnes pro většinu firem i uživatelů uspokojivá. Počítačové sítě jsou dnes používány širokou veřejností, firmami i tajnými službami. Každá z těchto skupin má svoje individuální požadavky, v některých bodech se dokonce shodují. Každý chce mít svoje data v bezpečí, chráněná před neoprávněným přístupem cizích osob. Problematika bezpečnosti počítačových sítí tak tvoří jednu z klíčových oblastí, kterou se počítačoví experti zabývají. Rostoucí počet používaných systémů v oblasti IT si žádá i nové postupy při zabezpečování počítačových sítí.

1. Zabezpečení počítačových sítí

Kontrola zabezpečení sítě je nikdy nekončící proces, jehož účelem je testování odolnosti systému proti neoprávněnému přístupu. Skládá se z velkého množství testů, z nichž každý má za úkol ověřit bezpečnost určité části systému. Tímto způsobem se kontrolují všechny prvky sítě – od uzlů v páteřních sítích až po nastavení v uživatelských stanicích. Kontrolu bezpečnosti je třeba provádět na všech úrovních, protože pokud by byl některý z prvků sítě vynechán, může dojít k ohrožení celé sítě.

Útočníci využívají chyb, které jsou v operačních systémech nebo v konfiguraci síťových prvků. Tyto chyby jsou pravidelně zveřejňovány na internetových stránkách výrobců síťových komponent spolu s návodem, jak chybu odstranit. Úkolem síťových administrátorů je tyto informace sledovat a upravovat podle nich aktuální nastavení sítě.

Dalším zdrojem informací o bezpečnostních mezerách jsou weby spravované hackery. Na těch jsou vedle informacích o bezpečnostních chybách a jejich opravách také informace o tom, jak těchto chyb využít k průniku do systému. Poté, co útočník ovládne některou z částí sítě, může ji využít k vlastní potřebě. V ohrožení jsou tak především citlivá firemní data o zákaznících nebo samotná funkčnost sítě. Útočníci pak často využívají ovládnuté systémy pro útoky na jiné cíle. Ukázka připojení firemní sítě do Internetu s využitím demilitarizované zóny je znázorněna na **Obr. 1.1**.



Obr. 1.1 Připojení firemní sítě do Internetu s využitím demilitarizované zóny pro servery přístupné z Internetu

1.1 Hledání informací o cílovém systému

Před provedením jakéhokoliv útoku je třeba získat co největší množství informací o cílové síti, zařízení nebo počítači. Nejčastěji využívaným zdrojem je Internet.

Prvním krokem je důkladné prohlédnutí webových stránek firmy. Můžeme na něm získat informace o adresách poboček, telefonních číslech, emailových adresách nebo použitích technologiích pro intranet společnosti.

Využít lze samozřejmě i vyhledávací weby, např.:

- www.seznam.cz,
- www.google.com.

Pří získávání informací o síti nejdříve identifikujeme domény společnosti a jejich síťové adresy. Tyto informace jsou uloženy v whois databázích, např.:

Tab. 1.1 Whois databáze

Whois server	Adresa
Celosvětový	http://www.whois.net
	http://www.allwhois.com
Evropa	http://www.ripe.net
Asie a Pacifik	http://whois.apnic.net
U.S. armáda	http://whois.nic.mil
U.S. vláda	http://whois.nic.gov
Český	http://www.nic.cz

Informace z whois databází lze velice rychle získat z okna terminálu v Unixu příkazem `whois`.

Tímto postupem získáme informace o:

- subjektu, který doménu registroval,
- jméno domény,
- kontakt na administrátora (nejčastěji emailová adresa a telefon),
- datum vytvoření záznamu v databázi.

Vlastníka určité síťové adresy můžeme získat na webu ARIN (American Registry for Internet Numbers) - whois.arin.net nebo RIPE NCC – http://whois.ripe.net. Do vstupního pole zadáme IP adresu, výstupem nám bude název společnosti, která ji vlastní. Můžeme tak získat např. informaci o tom, zda společnost využívá pro webovou prezentaci svoji síť nebo využívá server hosting.

Další informace jsou uloženy v DNS serverech společnosti. DNS je databáze, která mapuje jména počítačů na IP adresy a naopak. Jednou z největších chyb při konfiguraci DNS je povolení přenosu zóny (informací o zóně) na libovolný počítač v Internetu.

Informace o zóně jsou přenášeny z primárního jmenného serveru (primary nameserver) na sekundární (secondary nameserver) kvůli zajištění redundance dat. Redundance je nutná v případě výpadku primárního jmenného serveru, kdy jeho funkce přebírá sekundární server. Přenos zóny by měl být povolen pouze na sekundární servery. Některé DNS servery dovolují přenos těchto informací na libovolný počítač, čehož může lehce využít útočník. To je nebezpečné zvláště v případě, kdy primární DNS server obsahuje informace o interní síti společnosti. Útočník pak může získat pouhým přečtením informací o zóně kompletní přehled serverů ve vnitřní síti společnosti. Také lze podle jmen počítačů odhadnout, k čemu konkrétnímu servery slouží. Podrobnější informace jsou uvedeny v [2].

Obranou proti zneužití záznamů v DNS je omezení přenosu zón pouze na autorizované servery. Ve verzi 9 programu BIND toho lze dosáhnout direktivou `allow-transfer` v konfiguračním souboru `named.conf`. V případě Microsoft DNS lze použít volbu Notify. Dalším krokem může být filtrování příchozích TCP spojení na portu 53 na hraničních směrovačích, čímž se zamezí přenosu zóny mimo intranet, pro což je potřeba TCP spojení, avšak neznemožní se provádění překladů, protože standardní DNS dotazy využívají protokol UDP.

Další technikou je průzkum sítě, který lze provádět pouze v případě, že známe adresy sítí společnosti. K tomu se využívají programy:

- **`traceroute`** v Unixu,
- **`tracert`** ve Windows.

Tento program zobrazuje cestu, kterou putuje IP paket na své cestě k cílové adrese. Využívá TTL pole v paketech, které postupně zvyšuje od jedné po jedničce a získává tak informace o jednotlivých směrovačích v cestě k cílové adrese. Je třeba zmínit, že existuje několik cest k cílovému adrese a že může být na některých portech směrovačů podle

nastavené polity v ACL blokován. Ve Windows jsou vysílány ICMP pakety, v Unixu standardně UDP, lze však použít přepínač `-I`, který zajistí, že budou využívány ICMP. Výsledky se mohou v obou případech lišit, protože na některých portech mohou být ICMP nebo UDP pakety blokovány. Podobné programy s grafickým rozhraním lze stáhnout z:

- `http://www.visualroute.com`,
- `http://www.neotrace.com`.

Proti průzkumu sítě útočníkem se lze pochopitelně do jisté úrovně bránit. Slouží k tomu komerční detektory průniku do sítě (NIDS – Network Intrusion Detection System). Další možností je použít volně šiřitelný program `snort` nebo nakonfigurovat hraniční směrovače tak, aby omezovaly přístup ICMP a UDP paketů ke specifickým cílům.

1.1.1 Skenování

Na základě výše uvedených postupů můžeme získat IP adresy důležitých směrovačů a serverů. Těchto znalostí využijeme k získání podrobnějších informací o zařízeních a jejich konfiguraci. Zde je základním krokem skenováni portů, abychom zjistili, na kterých z nich běží určité služby.

Prvním krokem mapování je použití příkazu `ping`, který nám dá informaci o tom, zda je cílový objekt živý. Pomocí nástroje ping se většinou netestuje pouze jedna adresa ale velké množství cílových objektů, jejichž adresy jsou zpravidla uloženy v nějakém souboru, ze kterého je ping čte. Pro zaslání žádosti využívá ping ICMP paket ECHO Typ 8. Pro odpověď od cílového objektu se využívá paket ICMP ECHO_Replay Typ 0.

Další programem, který se pro skenování často využívá je **`nmap`** `[3]`. Tento program jsem použil ve skriptech pro kontrolu zabezpečení počítačové sítě v praktické části diplomové práce. Nmap je velice schopný nástroj, který nám dává informaci o otevřených portech na cílovém objektu. Pro ně nám dále vypíše použitý protokol, běžící službu a především její verzi. Program nmap nám dává také informaci o použitém operačním systému. Toho lze využít při kontrole, zda jsou používány nejnovější verze softwaru, které obsahují záplaty chyb z předcházejících verzí. Pro tento postup jsem napsal skripty, které tento postup automatizují, tedy skenování nemusí být prováděno pro každé zařízení ručně. Těchto informací využívají útočníci, kteří tak mohou zjistit, zda není verze softwaru zastaralá a tudíž náchylnější k prolomení. Pro zkušeného útočníka pak není problém najít na Internetu návod,

jak chybu v dané verzi softwaru využít k průniku do systému. Úkolem správců sítí a systémů je pochopitelně dbát na aktuálnost použitého programového vybavení.

Příklad výpisu programu nmap:

```
C:\>nmap -P0 -sV -O scanme.nmap.org

Starting Nmap 4.20 ( http://insecure.org ) at 2007-03-28 17:36 Střední Evropa
Interesting ports on scanme.nmap.org (205.217.153.62):
Not shown: 1692 filtered ports
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 4.3 (protocol 2.0)
53/tcp  open   domain
70/tcp  closed gopher
80/tcp  open   http    Apache httpd 2.2.2 ((Fedora))
113/tcp closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.15-27-686 (Ubuntu Dapper, X86)
Uptime: 102.136 days (since Sat Dec 16 13:20:31 2006)

OS and Service detection performed. Please report any incorrect results at
http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 56.610 seconds
```

Ve výše uvedeném příkladu jsou použity tyto přepínače:

- **-P0** vynechává prověření cílového objektu pingem,
- **-sV** na otevřených portech se snaží zjistit spuštěnou službu a její verzi,
- **-O** prověřuje operační systém.

Dalšími nástroji jsou

- **`hping2`** (http://www.hping.org),
- **`icmpenum`** (http://www.portcullis-security.com).

Proti technikám hromadných pingů je třeba síť chránit. Mohou totiž prozradit nejen spoustu údajů o vnitřní síti, ale také je lze využít k zahlcení a pádu sítě jako takové. Využívá se tu technik, které jsou schopny detekovat útoky. Dalším krokem může být identifikace útočníka a filtrování paketů obsahujících v položce zdroje jeho IP adresu.

Dalším možným postupem je zakázání ICMP paketů. Ty jsou ale využívání pro monitorovací funkce v síti, takže tento postup je třeba důkladně zvážit. Lze také zakázat pouze některé typy ICMP paketů nebo definovat v ACL směrovačů a firewallů systémy, které mají povoleno ICMP pakety využívat.

Pokud se nám podaří detekovat skenování sítě, můžeme očekávat také útok na síť. Hlavní metodou detekce je použití IDS, například `snort` (http://www.snort.org), který lze získat zdarma.

Další vhodné programy jsou:

- **scanlogd** (http://www.openwall.com/scanlogd),
- **PortSentry** (http://www.psionic.com) – umí na skenování i aktivně reagovat,
- **Genius** od Independent Software pro Windows,
- **ZoneAlarm** – pracuje jako firewall i IDS a k osobnímu užití je zdarma.

Je vhodné dbát na to, aby na serveru zbytečně neběžely služby, které se momentálně nevyužívají.

1.2 Ochrana vnitřní sítě společnosti

Nejdůležitějším úkolem správců systémů při ochraně vnitřní sítě proti útokům „zvenku“ je zajištění správného nastavení filtračních pravidel při komunikaci z vnější sítí. Za tu lze považovat podnikovou síť jiné společnosti nebo Internet. K dosažení tohoto cíle se využívají především techniky filtrování komunikace prostřednictvím proxy a firewallů.

1.2.1 Filtrování

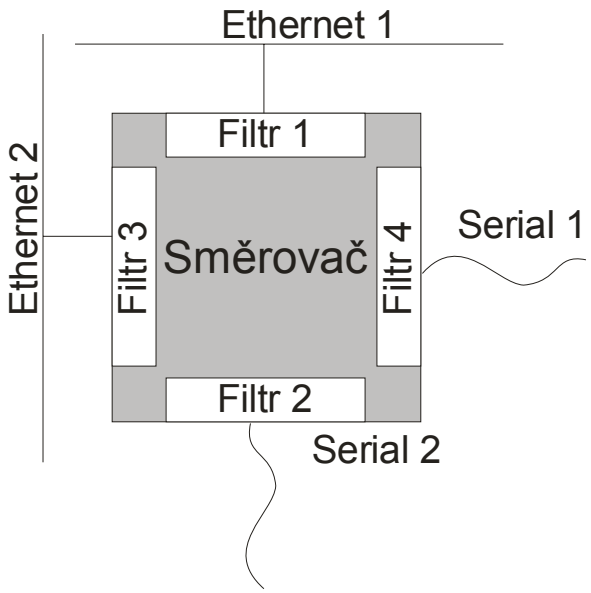
Filtrování je proces, při kterém dochází ke kontrole procházejících paketů na základě jejich obsahu. Podle něj se učiní rozhodnutí, jak se s paketem bude zacházet. Filtrování je možné provádět na různých vrstvách:

- **Linková vrstva** – realizováno nejčastěji přepínači, někdy firewally,
- **Filtrace protokolů IP a TCP** – realizováno na směrovačích a firewallech,
- **Filtrace aplikačních protokolů** – realizováno firewally.

Filtry na linkové vrstvě filtrují podle informací uvedených v záhlaví linkového rámce. Je tedy třeba aby filtr dokázal číst obsah linkových rámců. Stejný požadavek je i na filtraci na úrovni protokolu IP nebo TCP, kdy se rozhoduje podle záhlaví IP datagramu nebo UDP segmentu. V případě aplikačních protokolů musí filtr rozumět jak záhlaví, tak i přenášeným datům.

1.2.1.1 Filtrace na úrovni linkové vrstvy

Filtrací na přepínači v lokální síti můžeme zajistit kontrolu linkové adresy. Lze tak zabránit útokům z vnitřní sítě, kdy si by si útočník přinesl vlastní počítač. Tuto ochranu je možné obejít změnou linkové adresy na přineseném počítači, kterou umožňuje specializovaný software. Každé rozhraní směrovače má vlastní filtr, viz. **Obr. 1.2**.

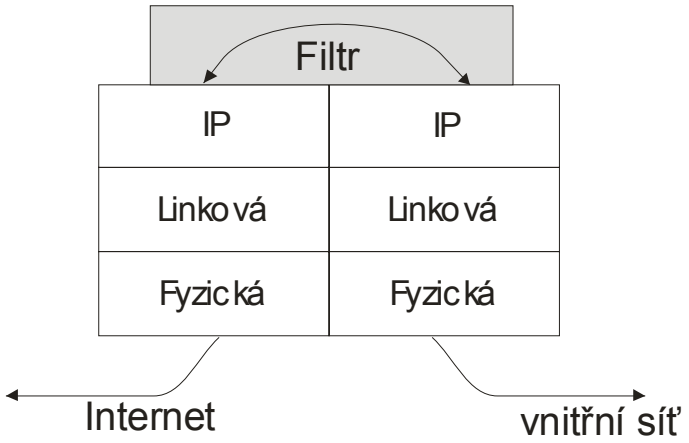


Obr. 1.2 Filtry na směrovači se přiřazují jednotlivým rozhraním podle [1].

1.2.1.2 Filtrace na úrovni protokolu IP

Tento druh filtru se umisťuje mezi vnitřní síť společnosti a Internet (viz. **Obr. 1.3**) a je většinou součástí přístupových směrovačů. Podle informací v záhlaví IP datagramu se rozhoduje, jak se bude s IP datagramem zacházet – zda bude předán na další rozhraní nebo bude zahozen. Tyto filtry jsou realizovány i na firewallech a serverech.

V záhlaví je nejdůležitější zdrojová a cílová IP adresa. Filtr samotný lze definovat tak, že vše je povoleno až na výjimky, které jsou v něm uvedeny a tvoří zákaz. Druhý a častější postup je definování pravidel pro to, co je povolené. Zbývající komunikace je zakázána.



Obr. 1.3 Filtr na úrovni protokolu IP na směrovači podle [1].

U směrovačů CISCO je několik druhů filtrů:

- **Standardní filtry** – filtrují podle adresy odesilatele IP datagramu. Využívají se například, když existují dvě vnitřní sítě a pouze jedna z nich má povolen přístup do Internetu.
- **Rozšířené filtry** – filtrují podle adresy odesilatele i příjemce IP datagramu. Dále umí filtrovat podle informací uvedených v záhlaví TCP segmentu.
- **Dynamické filtry** – umožňují specifické nastavení průchodu směrovačem pro uživatele, který se nejprve autentizuje (např. Telnetem). Praktické využití tohoto filtru je především pro správce systému, který se připojuje vzdáleně.
- **Reflexní filtry** – sledují relaci vyššího protokolu (TCP nebo UDP). Ve směru ven z vnitřní sítě povolují vznik relace. Ve směru do vnitřní sítě propouští pouze pakety, které patří již vzniklé relaci.

Samotný filtr je definován seznamem označovaným jako „Access list“, který se přiřadí určitému rozhraní.

Následující záznam zachycuje příklad konfigurace filtru pro směrovač od společnosti Cisco:

```
access-list 89 permit 10.80.78.0 0.0.0.255
access-list 89 deny 10.80.79.0 0.0.0.255

interface serial 1 ip access-group 89 out
```

V předcházejícím příkladu definujeme pravidla pro komunikace ven z vnitřních sítí v access listu číslo 89.

Spolu s filtrem na úrovni TCP tvoří výkonný systém pro ochranu vnitřní sítě před útoky z vnější sítě.

1.2.1.3 Filtrace na úrovni TCP

Zde se používá rozšířený filtr definující služby, pro které je komunikace na úrovni TCP nebo UDP povolena. Často se spojuje s filtrací protokolu ICMP, který informuje odesílatele o případném nedoručení některého ze segmentů na cílovou stanici. K rozhodování dochází podle obsahu TCP záhlaví.

1.2.1.4 Reflexní filtry

Reflexní filtr zpracovává odchozí pakety z vnitřní sítě. Pokud zachytí paket, kterým se zahajuje nová relace, pak v opačném směru vygeneruje položku dočasného filtru, která umožňuje průchod paketům téže relace z vnější sítě (např. Internetu). Položka v odchozím filtru obsahuje informaci o protokolu vyšší vrstvy, IP adresu a port příjemce a odesilatele.

Pokud není v síti spuštěna DMZ, je reflexní filtr definován pro vnější rozhraní směrovače. Jestliže je v síti DMZ spuštěna, definuje se reflexní filtr na vnitřním rozhraní směrovače.

Na směrovačích mohou být uloženy seznamy IP adres, se kterými je komunikace zakázaná. Vznikají ve chvílích, kdy se někdo z dané IP adresy snaží útočit na vnitřní síť.

1.2.1.5 Aplikační protokoly a jejich filtrace

Při komunikaci po síti je velice účinné filtrovat aplikační protokoly. V této kapitole jsou popsány některé z nich spolu s návrhem jejich zabezpečení. Bližší informace k jednotlivým typům proxy jsou uvedeny v kapitole 1.2.2.

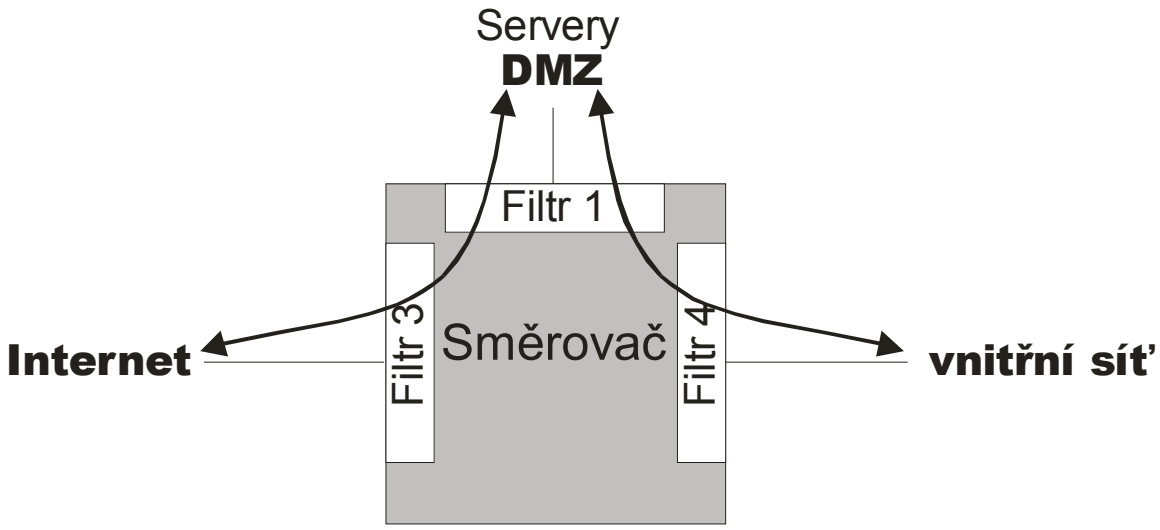
Aplikační protokoly [1]:

- **Telnet a SSH -** Protokol Telnet používá nezabezpečenou komunikaci, proto je lepší ho vůbec nepoužívat. Jako náhrada za něj je používán SSH, který využívá zabezpečený kanál a nabízí stejné možnosti jak Telnet. SSH používá port 22 a TCP.
- **FTP** - pokud používáme aktivní FTP, tak pro zabezpečení nestačí pouze reflexní filtr, ale je třeba použít proxy. Pro pasivní FTP lze bezpečnost komunikace ošetřit pomocí reflexního filtru.
- **HTTP** – většina webových prezentací je spuštěna na portu 80/tcp. Nevýhodou použití filtru je situace, kdy je na internetových stránkách použit odkaz na aplikací běžící na jiném portu než 80. Požadavek na tento port není veden z vnitřní sítě a tudíž komunikace na tomto portu zvenku nebude do vnitřní sítě propuštěna. V praxi se pro filtraci http nejčastěji používá reflexní filtr.
- **SSL a TLS** – pro komunikace prostřednictvím protokolu SSL a TSL se využívá na hraničním směrovači tunel, protože komunikace je šifrována a filtr nemůže vidět do přenášených dat. Používá se pro prohlížení Internetu, práci s emailem nebo třeba Internet Messaging.
- **SMTP** – pravidla pro filtrování komunikace protokolem SMTP lze definovat pouze v případě, že ve vnitřní síti je pouze klient SMTP využívající port 25/tcp. Pokud by běžel ve vnitřní síti SMTP server, tak nelze filtrací ošetřit příchozí emaily z vnější sítě. SMTP server se tak musí umístit do DMZ nebo Internetu.
- **POP3 a IMAP4** – pokud v síti běží klienti těchto protokolů a server je DMZ nebo Internetu, je jejich filtrace jednoduchá. Z proxy je nejvhodnější transparentní proxy, zvládne ji i generická proxy.

- **LDAP** – filtrování komunikace LDAP klienta z vnitřní sítě je bezproblémové. Komplikace mohou vzniknou v případě použití proxy. Nejvhodnější je opět transparentní proxy a dostačující je generická.

1.2.1.6 Servery

Pokud ve firmě potřebujeme provozovat servery, které mají být dostupné z vnější (veřejné) sítě, je nejlepším řešením jejich umístění do demilitarizované zóny (DMZ), viz. **Obr. 1.4**. Tou je fyzická nebo logická podsíť, která je přístupná z vnitřní i vnější sítě. Pokud by se útočníku podařilo tyto servery ovládnout, budou za použití demilitarizované zóny napáchané škody menší, než kdyby se tyto servery nacházely ve vnitřní síti.

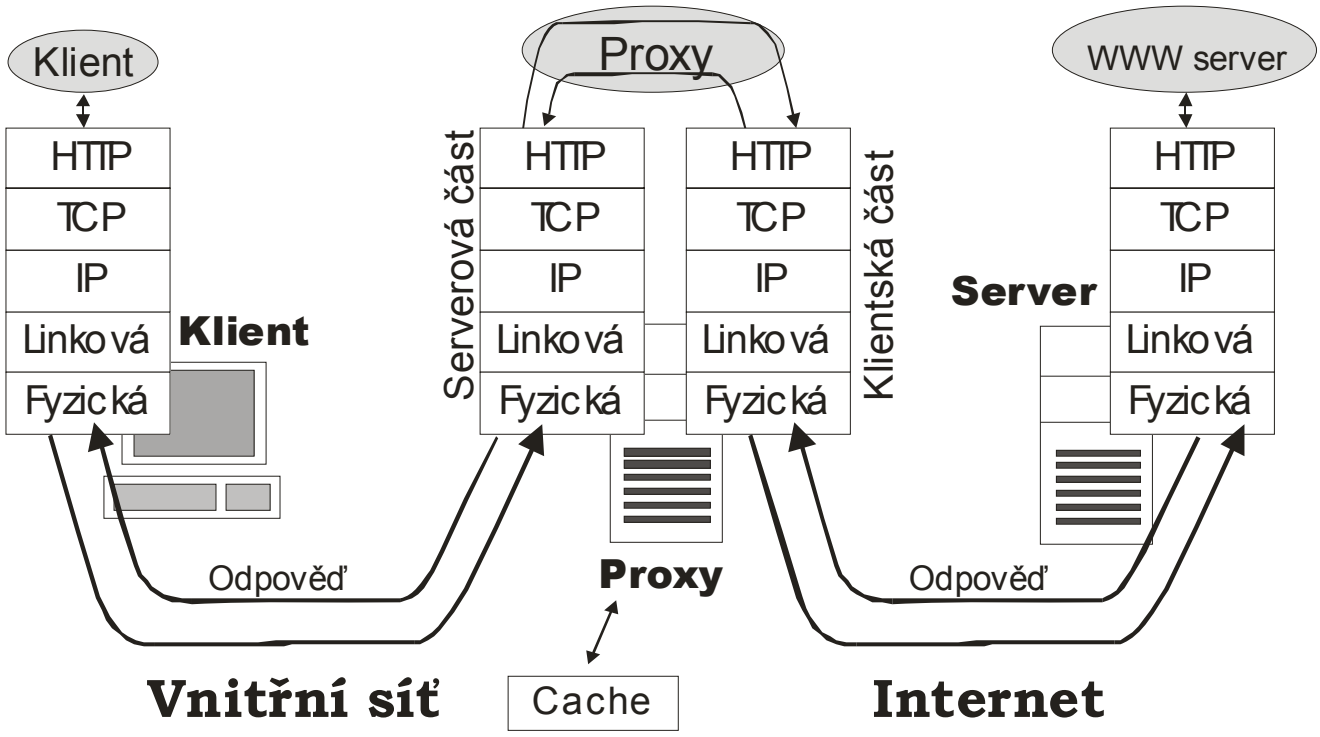


Obr. 1.4 Zapojení směrovače s DMZ

Toto řešení se používá pro servery SMTP, POP3, IMAP4, HTTP, HTTPS, DNS, FTP. Pro Telnet, SSH a LDAP pro komunikaci to Internetu se pak nastavují pouze filtry.

1.2.2 Proxy

Proxy je program pracující na aplikační úrovni a je složen ze dvou částí (pracuje jako server i jako klient). Serverová část na vnitřním rozhraní přijímá požadavky od klientů a předává je klientské části proxy na vnějším rozhraní, která jménem klientů předává požadavky cílovému serveru, viz. **Obr. 1.5**.



Obr. 1.5 Komunikace z vnitřní sítě do Internetu přes proxy [1]

Proxy si lze představit jako aplikací běžící na počítači se dvěma síťovými rozhraními – jedno je připojeno do vnitřní a druhé do vnější sítě. Existují i proxy s jedním síťovým rozhraním které mohou fungovat jako filtr nebo třeba cache.

Práce proxy spočívá ve vyměňování požadavků mezi serverovou a klientskou částí. Tato komunikace může obsahovat:

- **Filtraci** – definuje, co se smí předat dál nebo třeba, které počítače mající oprávnění komunikovat s druhou stranou. Proxy vidí do aplikačního protokolu, takže je možné definovat podrobnější pravidla pro komunikaci než u směrovačů.
- **Cache** – proxy může zpracované požadavky a jejich odpovědi ukládat na disk. V případě opakování požadavku pak může proxy využít informací z paměti cache.

U protokolů Telnet, SSH, FTP, POP3, IMAP4, LDAP a vlastních programů se vyskytuje problém se sdělováním jména cílového serveru, na který má klientská část proxy navázat spojení. Existují čtyři řešení:

- **klasická proxy**,
- **generická proxy**,
- **transparentní proxy**,
- **SOCKS**.

Stejný problém nastává u SMTP a NNTP, pokud nechceme na proxy spustit jejich server ale chceme spustit jednoduchou proxy, která by klientům vnitřní sítě umožňovala přístup na server ve vnější síti (Internetu).

1.2.2.1 Klasická proxy

Používá se pro FTP a Telnet. Například při použití Telnetu se uživatel nejdříve připojí na proxy, kde zadá příkaz connect, ve kterém určí název nebo IP adresu skutečného serveru. Klientská část proxy naváže spojení s cílovým serverem a začne předávat mezi klientem a tímto serverem data. Klientovi se pak jeví, jako by mezi ním a cílovým serverem žádná proxy nebyla.

Tab. 1.2 Komunikace přes klasickou proxy podle [1].

Směr	IP adresa odesílatele	Port odesílatele	IP adresa příjemce	Port příjemce	Proxy	IP adresa odesílatele	Port odesílatele	IP adresa příjemce	Port příjemce
ven	klient	> 1023	proxy.firma.cz	1500	→	proxy.firma.cz	> 1023	server	23
dovnitř	proxy.firma.cz	1500	klient	> 1023	←	server	23	proxy.firma.cz	> 1023

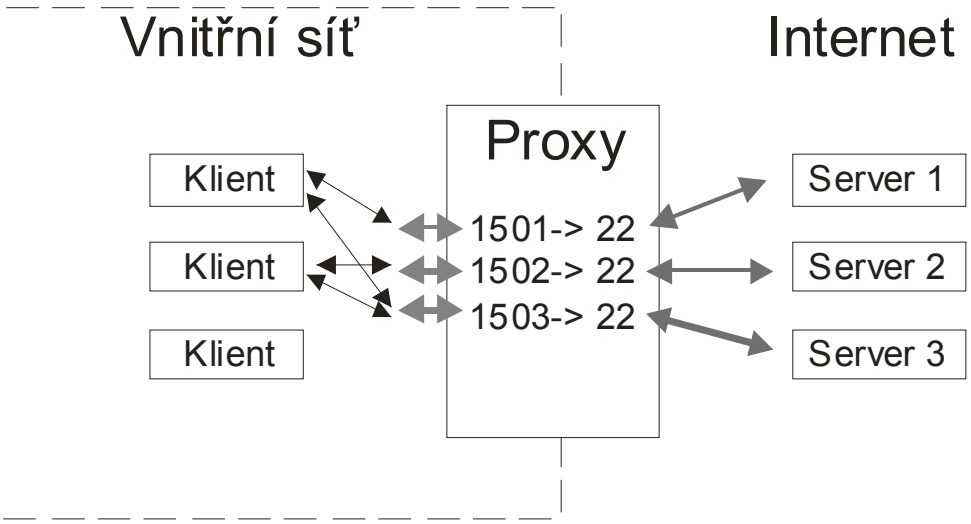
1.2.2.2 Generická proxy

Někteří klienti nemohou nebo nechtějí vést úvodní dialog s proxy, aby jí sdělili cílový server. Generická proxy je program, který může spouštět a konfigurovat správce sítě podle momentálních požadavků. Dokáže pracovat s TCP i UDP.

Serverová část proxy je spuštěná na konkrétním portu, který určil správce, a je připravena přijímat požadavky od klientů ve vnitřní síti. Klientská část proxy je pevně nastavena na jeden cílový server (viz. **Obr. 1.6**).

Jedna spuštěná generická proxy dokáže obsloužit jeden cílový server. Naráz jích pochopitelně může být spuštěných více.

Generické proxy se používají pro protokoly POP3, IMAP4, SSH, LDAP, SMTP A NNTP.



Obr. 1.6 Komunikace přes generickou proxy

1.2.2.3 Transparentní proxy

Klient naváže spojení s transparentní proxy a má pocit, že navázal spojení se skutečným serverem. Klientská část transparentní proxy hned navazuje spojení s cílovým serverem. Transparentní proxy nemusí vést s klientem žádný dialog.

Předpoklady pro práci transparentní proxy jsou:

- Klient ve vnitřní síti má možnost přeložit jméno cílového serveru nebo počítače v Internetu na IP adresu. Cílová adresa IP datagramu je adresa cílového serveru.
- IP datagram odeslaný do Internetu musí projít přes transparentní proxy.
- IP datagramy jednoho spojení musí procházet přes jednu transparentní proxy. Může existovat více transparentních proxy naráz.

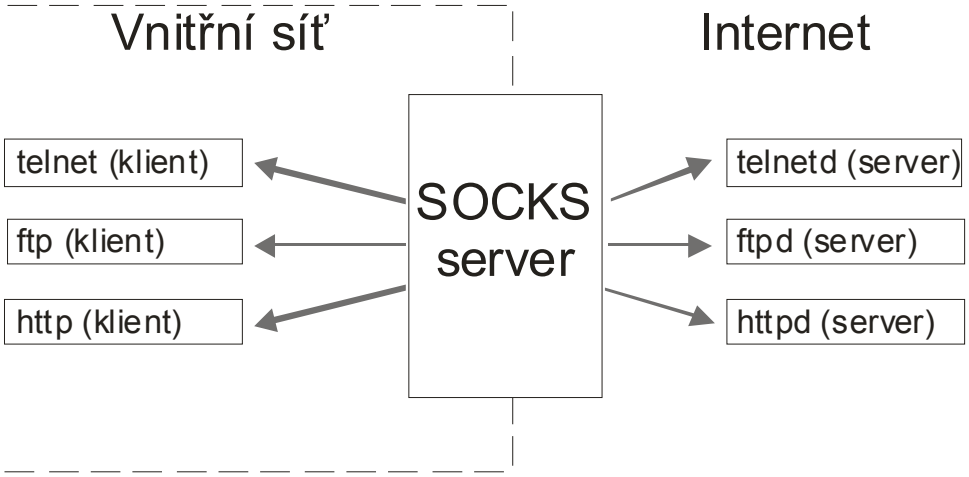
Používá se Telnet, FTP, NNTP, IMAP, POP a POP3.

Tab. 1.3 Komunikace přes transparentní proxy podle [1].

Směr	IP adresa odesílatele	Port odesílatele	IP adresa příjemce	Port příjemce	Proxy	IP adresa odesílatele	Port odesílatele	IP adresa příjemce	Port příjemce
ven	klient	> 1023	server	23	→	proxy.firma.cz	> 1023	server	23
dovnitř	server	1500	klient	> 1023	←	server	23	proxy.firma.cz	> 1023

1.2.2.4 SOCKS

Proxy se otevře jen pro klienty, kteří projdou autentizací. Ověření autentičnosti klienta probíhá podle IP adresy nebo jména a hesla nebo jednorázového hesla. SOCKS server (viz. **Obr. 1.6**) je proxy společná pro všechny aplikační protokoly.



Obr. 1.7 Komunikace přes SOCKS server

Mezi serverem a klienty je řízena komunikace protokolem SOCKS a má za úkol otevřít příslušnou proxy na SOCKS serveru. Řídící kanál se SOCKS serverem je na portu 1080/tcp. Tímto kanálem obdrží klient IP adresu a port serverové části proxy, kterou otevírá pro klienta. Na tento port se připojí pomocí protokolu TCP.

Zde je postup připojení na SOCKS:

1. **Dohoda na autentizační metodě** a zabezpečení komunikace (někdy je komunikace mezi klientem a SOCKS serverem šifrována).
2. **Autentizační dialog** (může být ale vynechán).
3. **Požadavek na zřízení příslušné proxy** – klient vysílá požadavek CONNECT nebo BIND pro TCP proxy a požadavek ASSOCIATE pro UDP proxy.
4. **Datová komunikace na cílový server přes proxy.**

1.2.3 NAT

NAT (Network Address Translator) překládá IP adresy na směrovači mezi vnitřní a vnější sítí. Jedná se o směrovač na straně firmy, a ne o směrovač na straně poskytovatele Internetu. V případě, že je vnitřní síť připojena do Internetu dvěma směrovači, je třeba dbát na stejnou konfiguraci obou. Pokud jsou IP adresy přidělování dynamicky, existuje nebezpečí, že vznikne konflikt.

Pro firmu znamená použití NAT úsporu finančních prostředků, protože si může od poskytovatele připojení k Internetu pronajmout méně IP adres než je počet počítačů v síti. Druhou výhodou je, že se tak šetří IP adresy v celosvětové měřítku. Počet IP adres verze 4 totiž v současnosti přestává stačit nárokům na počet připojených zařízení do Internetu. Tento problém řeší až plné zavedení IP verze 6.

NAT je výhodné použít, existuje-li v síti nějaký server, na který je veden velký počet dotazů. Jeden server by takový provoz nezvládl a tak je za NAT umístěno několik serverů, které jsou do vnější sítě spojeny přes jeden směrovač. Ten zajišťuje, že požadavky z vnější sítě jsou vedeny na jednu adresu a jsou rovnoměrně rozděleny mezi servery ve vnitřní síti, které mají pochopitelně odlišné IP adresy. Dojde tak k rozdělení zátěže.

V praxi se používá několik typů NAT, které jsou popsány v následující části.

1.2.3.1 Jednoduchý NAT

Slouží pro navázání spojení z vnitřní sítě do Internetu. Směrovač obsahuje seznam IP adres vnitřní sítě a seznam IP adres, které jsou mu přiděleny poskytovatelem připojení k Internetu. Při odchozím spojení se paketu změní IP adresa z vnitřní sítě na IP adresu do vnější sítě. Po příchodu paketu zpět se provede opačný převod. Ve vnitřní síti jsou směrovatelné adresy jak vnitřní tak vnější sítě.

Příklad zápisu pro zařízení firmy CISCO:

```
ip nat inside source static 10.0.0.1 196.65.84.4
ip nat inside source static 10.0.0.2 196.65.84.5

interface ethernet 0 ...
ip nat inside

interface serial 1 ...
ip nat outside
```

Přidělování IP adres může na směrovači probíhat staticky i dynamicky.

1.2.3.2 Rozšířený NAT

Network Address and Port Translation (NAPT) dokáže efektivněji nakládat s IP adresami, které jsou firmě přiděleny pro připojení do Internetu. Několika adresám z vnitřní sítě dokáže přidělit jednu adresu do vnější sítě.

NAPT používá položky tabulky NAT rozšířené o čísla portů a typ protokolu.

Konfigurace CISCO směrovače:

```
ip nat pool pytel 196.67.79.16 196.67.79.31 netmask 255.255.255.240
access-list 1 permit 10.0.0.0 0.255.255.255
ip nat inside source list 1 pool pytel overload

interface ethernet 0 ...
ip nat inside

interface serial 1 ...
ip nat outside
```

1.2.3.3 Dvojitý NAT

Slouží k řešení kolizních IP adres, což nastává v případě, že se ve vnitřní síti používají IP adresy, které jsou v Internetu přiděleny někomu jinému, tzv. Overlapping Addresses. Podrobnější popis je v [1].

1.2.4 Firewall

Firewall je systém sloužící k bezpečnému oddělení vnitřní sítě od Internetu. Používá filtraci, wrappery, proxy, brány nebo třeba SOCKS.

Firewall ukládá během své práce informace o provozu do souborů (logů), ze kterých je možné vyčíst dílčí akce i vytvářet hlášení, neboli tzv. reporty.

V nastavení firewall jsou definována pravidla, kdy určitá událost může spustit nějakou akci (alert). Akcí se rozumí spuštění nebo ukončení nějakého programu, např.:

- **odeslání zprávy o určité události** – prostřednictvím SMS, emailu apod.,
- **ukončení proxy nebo filtru**, na kterém došlo k události,
- **zapsání IP adresy potenciálního útočníka na černou listinu**,

- **ukončení práce celého systému**,
- **spuštění libovolného programu**.

Je snaha zajistit, aby firewall dokázat zabezpečit i komunikaci na linkové vrstvě, a to stavy, kdy:

- **IP datagramy s adresou odesilatele z vnitřní sítě** přicházejí z Internetu.
- **IP datagramy s adresou odesilatele z Internetu** přicházejí z vnitřní sítě.
- **Uživatelé vnitřní sítě se pokoušejí přes firewall napadnout servery ve vnitřní síti**, na které běžně nemají přístup.

Rozhraní firewallu se dělí:

- **vnitřní**,
- **vnější**,
- **rozhraní pro DMZ**.

NAT, filtraci nebo proxy provádí firewall sám. Některé jiné služby (např. kontrola přítomnosti virů) využívají aplikací specializujících se na určitou oblast. Tyto aplikace mohou být spuštěny na stejném nebo jiném počítači jako firewall. Tyto počítače jsou umístěny ve vnitřní síti.

Základem každého firewallu je buď proxy (např. MS Proxy) nebo filtrace (např. CISCO PIX). Některé firewally podporují proxy i filtraci.

Seznam známých firewallů:

- Raptor http://www.symantec.com
- Gauntlet http://www.pgp.com
- Firewall Toolkit http://www.fwtk.org
- Cyberguard KnightStar http://www.cyberguard.com
- MS Proxy http://www.microsoft.com
- CISCO PIX http://www.cisco.com
- Firewall-1 http://www.checkpoint.com

1.2.5 IPsec

IPsec je skupina protokolů, které se snaží zabezpečit přenos na úrovni IP, tedy jednotlivé IP datagramy. Zabezpečení provádí operační systém nebo hraniční směrovač. Data přenášená mezi aplikacemi nebo uživateli na jednom počítači jsou nezabezpečená.

IPsec obsahuje dva základní režimy: transportní a tunel.

IPsec se dělí podle toho, v jaké části komunikačního kanálu mezi koncovými zařízeními jsou data šifrována:

- **Mezi koncovými počítači**.
- **Mezi dvěma směrovači** – může se jednat o hraniční směrovače mezi dvěma LAN sítěmi na dvou pobočkách jedné firmy.
- **Mezi počítačem a směrovačem** – používá se pro vzdálený přístup zaměstnance do vnitřní sítě odkudkoliv z Internetu.

Jednotlivá řešení lze kombinovat.

1.2.5.1 Transportní režim

IP datagram je rozšířen o bezpečnostní záhlaví, které je vloženo mezi záhlaví IP a záhlaví vyšší vrstvy. Bezpečnostní hlavička pak udává, jak je daná část IP datagramu zabezpečena.

1.2.5.2 Tunel

Tunel vezme původní IP datagram, zabezpečí jej a vloží do nového IP datagramu. Za IP záhlavím nového IP datagramu následuje bezpečnostní záhlaví.

V IP datagramu se tak nacházejí dvě záhlaví, které se označují jako vnitřní a vnější. Internet se tak stává pouze přenosovým prostředím. V praxi se toto řešení používá například ve firmách s více pobočkami, kdy je touto technologií zabezpečena komunikace mezi jednotlivými pobočkami.

1.2.5.3 Zabezpečení IP datagramu

Existují dva protokoly pro zabezpečení IP datagramu. První je IP Authentication Header (AH) a druhý IP Encapsulating Security Payload (ESP).

AH zabezpečí:

- integritu přenášených IP datagramů,
- autentizaci odesilatele IP datagramu,
- ochranu proti útoku zopakováním přenášených dat.

ESP zabezpečí narozdíl od AH ještě přenášená data šifrováním.

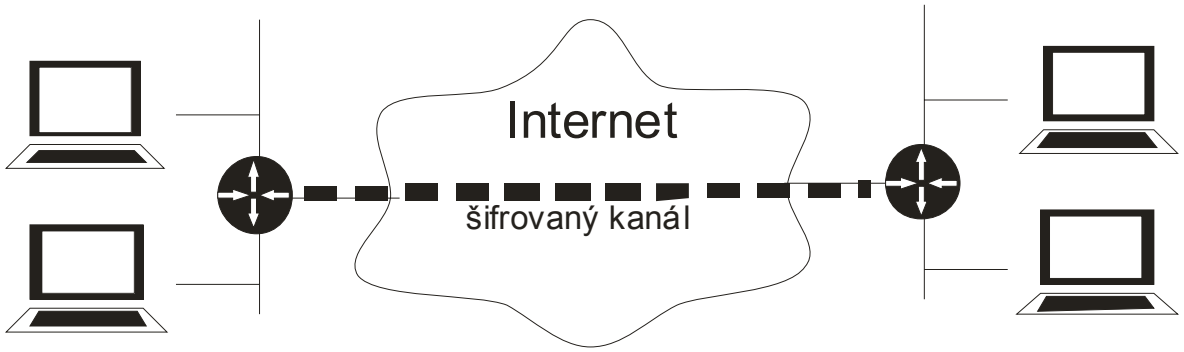
1.2.6 VPN

Virtual Private Network (VPN) je síť, které se používá především ve společnostech, které mají několik poboček. Jejím úkolem je zabezpečit komunikaci mezi jednotlivými částmi společnosti během přenosu dat přes Internet nebo síť poskytovatele připojení (**Obr. 1.8**). Jedná se o ekonomicky nepříliš náročné řešení.

Komunikační prostředí lze rozdělit na bezpečnou síť a méně důvěryhodnou síť. První z nich je lokální síť společnosti, do které jsou připojeny počítače zaměstnanců. Druhou je vnější síť.

Mezi sítí s důvěrou a bez ní se nachází firewall. Nejprve se klient (zaměstnanec) připojí na tento firewall a autentizuje se. Během autentizace se určí zabezpečení pro jeho spojení. Následuje odeslání dat na cílovou stanici, které je za firewallem již šifrováno.

Je moudré šifrovat veškerou komunikaci mezi interními sítěmi. Snižuje se tak riziko napadení koncové stanice útočníkem z vnější sítě.



Obr. 1.8 VPN mezi dvěma sítěmi

Za bezpečné protokoly VPN se považují [12]:

- **IPsec** (IP security).
- **SSL a TSL** – používá se pro tunelování celé sítě nebo zabezpečení web proxy. Výhodou VPN založené na SSL je, že ji lze používat z více míst, než ostatní protokoly. Ty nemusí být v AP vždy dovoleny.
- **OpenVPN** – klienti a servery jsou na Internetu volně ke stažení pro většinu operačních systémů. Pro šifrování používá OpenSSL.
- **L2TPv3** – Layer 2 Tunneling Protocol version 3 a novější.
- **VPN Quarantine** – koncové zařízení VPN komunikace se může stát cílem útoku, proto je třeba kontrolovat správnost jeho identity. Toto zajišťuje Microsoft ISA Server 2004/2006 spolu s VPN-Q 2006 od společnosti Winfrasoft nebo aplikace QSS (Quarantine Security Suite).
- **MPVPN** (Multi Path Virtual Private Network).

1.3 Útoky na počítačovou síť

1.3.1 DoS útok

Následkem útoku Denial of Service (DoS - odepření služby) je nedostupnost systému nebo ztráta dat. Existuje několik útoků typu DoS [2]:

- **Obsazení přenosové kapacity spoje** – následkem tohoto útoku je plně vytížený spoj a tudíž ostatní pokusy o přenos dat po něm selhávají. V případě

že se jedná o jediný spoj pro připojení k Internetu, tak je komunikace mimo LAN znemožněna.

- **Přivlastnění si systémových zdrojů** – tento typ útoků se snaží o vyčerpání systémových zdrojů cílového počítače (procesorový čas, paměti, diskový prostor). Následkem je nedostupnost služby nebo restartování operačního systému cílové stanice.
- **Chyby v programech** – útočník se snaží využít chyb v programových kódech serverových aplikací, a tak způsobit pád aplikace a nepřístupnost poskytované služby.
- **Útoky na systémy přesměrováním paketů** – změnou směrovacích tabulek dojde k znepřístupnění cílového počítače nebo sítě. Směrovací protokoly mají jen velice slabou ochranu proti útoku, neobsahují například autentizaci, a tak může lehce dojít k podvrhnutí administrativních dat.
- **Útoky na DNS** – v cache DNS serveru se změní údaje. Jmenný server potom dává chybný údaj o IP adrese cílového serveru.

K **DDoS** (Distributed Denial of Service) útoku se využívá velkého množství stanic zasílajících své dotazy na cílový server. Zde je nutné zmínit, že uživatelé stanic, z nichž tyto dotazy odcházejí, o této aktivitě jejich počítačů často vůbec neví.

2. Kontrola zabezpečení počítačové sítě

V praktické části své práce jsem se zaměřil na programování skriptů, které kontrolují zabezpečení sítě. Jako programovací jazyk jsem si zvolil Perl. Soubor skriptů je určen pro operační systém Unix i Windows.

Tento projekt jsem vyvíjel s ohledem na provozní potřeby IBM Global Services Delivery Centrum Brno, kde by mohl být v budoucnu použit. Toto středisko se zabývá dohledem sítí a koncových zařízení po celém světě. Základní služba poskytovaná zákazníkům je dohled nad funkčností těchto sítí. Druhým úkolem Delivery Centra je kontrolovat zabezpečení síťových prvků. Dosud byl tento problém řešen několika samostatným skripty nebo ručně. Tento postup je ale vzhledem k počtu kontrolovaných zařízení (tisíce) neefektivní. Proto mi byl zadán projekt na vytvoření souboru skriptů, které by dokázaly řešit tyto úlohy automaticky. Došlo by tak k výraznému urychlení kontroly zabezpečení sítí.

Jako platforma byl zvolen Unix, který se používá na počítačích určených pro management sítí.

Vstupem pro testovaní zabezpečení je seznam zařízení nebo konfiguračních souborů uložených v textovém souboru, u nichž je třeba prověřit úroveň nastavené bezpečnosti.

Výstupem je zpráva, která obsahuje celkový výsledek testu a vypisuje seznam nastavení, které by mohly být nebezpečné pro síť.

2.1 Skripty na kontrolu spuštěných aplikací pro komunikaci po síti

V této části projektu bylo mým úkolem napsat seznam skriptů, který bude kontrolovat zabezpečení koncových zařízení. Těmi se rozumí především počítače a servery.

2.1.1 Nmap

Pro skenování cílové stanice jsem zvolil program Nmap. Jedná se o program, který je určen pro platformu Unix, Windows i jiné. Nejdříve byl vyvinut pouze pro Unix, ale pro svou velkou oblíbenost byl upraven tak, aby ho bylo možné používat i v jiných operačních systémech.

Tento program je jedním z nejoblíbenějších programů sloužících pro skenování například firewallů, přepínačů, směrovačů nebo filtrů. Jedná se open source software, který je možné si zdarma stáhnout z Internetu.

Příklad výstupu programu nmap je uveden v kapitole 1.1.1:

2.1.2 Skript pro skenování sítě

V síti se nachází velké množství síťových zařízení od páteřních směrovačů až po uživatelské počítače. Pokud je některý z těchto prvků nedostatečně zabezpečen, může ohrozit bezpečnost celé sítě.

Nmap dokáže určit operační systém, který je spuštěn na cílovém zařízení včetně použité verze. Slouží k tomu přepínač `-O`.

Nmap provádí skenování portů cíle TCP i UDP pakety, podle parametrů zadaných při spuštění.

Tento program je používán jak hackery, tak správci sítí. Obě tyto skupiny se snaží najít málo zabezpečená místa sítí. První skupina využívá těchto informací k útokům a mnohdy i k páchání trestných činů. Úkolem správců sítí je naopak těmto situacím předcházet a starat se o správnou konfiguraci sítě, aby nebylo možné nebo alespoň velice obtížné se do ní neoprávněně připojit.

2.1.3 Skript provádějící sken

Tento skript má za úkol provádět sken těchto zařízení pro zjištění stavu jejich zabezpečení při připojení do sítě.

Název skriptu: **`port_scan.pl`**

Tento skript využívá pro skenování program nmap, kterým získává následující informace:

- **operační systém** a jeho verze na cílovém objektu,
- **otevřené porty**,
- **protokol** běžící na jednotlivých portech,
- **název služby** spuštěné na určitém portu,
- **verze služby** běžící na určitém portu,
- **uptime** zařízení.

Tyto informace jsou načteny a zpracovány skriptem. Důležité jsou všechny výše uvedené údaje, protože informují o aktuálním stavu rozhraní do sítě. Získané údaje jsou podrobeny kontrole stavu zabezpečení podle doporučení IBM pro bezpečnost sítě.

Spuštění skriptu:

```
perl port_scan.pl --list_of_objects=seznam_zarizeni_na_skenovani.txt
--output_file=report.html
```

Skript umožňuje testování pouze některých portů. To výrazně zvyšuje rychlost prováděné operace ve srovnání se skenem všech portů. Skript bude v praxi provádět skenování několika tisíc síťových zařízení, a tak bylo třeba věnovat čas jeho odladění. Seznam testovaných portů mi byl zadán od IBM. Vzhledem k tomu, že seznam testovaných portů se pro jednotlivá spouštění skriptu nebude lišit, rozhodl jsem se ho vnést do skriptu jako proměnnou a nezadávat ho prostřednictvím parametrů během spuštění skriptu. Seznam portů je uveden na začátku skriptu jako jeden z parametrů při spuštění nmapu. Skript napsaný v Perlu je možné editovat jednoduše i v poznámkovém bloku, seznam portů je oddělen čárkami a tak i případná změna seznamu portů nebude pro systémového inženýra žádný problém. Může ji jednoduše provést.

Testování portů probíhá jak pakety protokolu TCP, tak i UDP. Je to dáno požadavky na testování jednotlivých služeb na specifických portech. Celkem tak dochází během skenu jednoho zařízení k dvěma spuštěním programu **nmap[3]**.

2.1.4 Načtení informací o číslech portů a přiřazených službách z Internetu

Ve skriptu `port_scan.pl` je přiřazení služeb spuštěných na určitých portech vyřešeno programem nmap, který ve svém výpisu tuto informaci uvádí.

V praxi se však může vyskytnout situace, kdy nmap pro skenování nebude možné použít. Důvodem může být platforma, která nmap nepodporuje nebo třeba striktní bezpečnostní pravidla, která zakazují instalaci programů do managementových počítačů. Tento problém není zdaleka neřešitelný. Je třeba napsat skript, který bude sám provádět sken určitých portů. Výstupem bude seznam portů, které jsou na koncovém objektu otevřené.

Dalším krokem je přiřazení služeb k otevřeným portům. Některé z portů jsou předem určeny pro určité protokoly (např. port 80 pro http). Toho lze využít a k seznamu otevřených portů přiřadit službu, která na něm spuštěna.

Informace o aplikacích běžících na specifických portech lze získat z této webové adresy:

`http://www.iana.org/assignments/port-numbers [5]`

Soubor je možné stáhnout na disk a skriptem z něj načíst požadované informace.

2.1.5 Kontrola verzí spuštěných aplikací

Pro komunikaci po síti jsou otevřeny na cílovém objektu porty, z nichž každý slouží jiné aplikace. Z hlediska bezpečnosti není důležité pouze číslo otevřeného portu ale mnohem důležitější je aplikace a zejména její verze, která je na daném portu spuštěna.

Každá aplikace může představovat určité bezpečnostní riziko pro cílový objekt. Lze ji totiž využít pro nedovolený průnik do systému.

První otázkou, kterou je třeba si položit je, zda je nezbytné, aby byla na cílovém objektu tato aplikace skutečně spuštěna. Často se lze setkat se situací, kdy jsou na zařízeních spuštěny služby, které nejsou pro jejich provoz nezbytné nebo jsou dokonce dlouhodobě nevyužívány. Zvyšují tak potom zbytečně možnost úspěšného průniku útočníka do systému.

Skript `port_scan.pl` zjišťuje z výpisu programu nmap verzi spuštěné aplikace a porovnává ji s databází zranitelných (vulnerable) aplikací, kterou lze stáhnout z Internetu. Databáze obsahuje seznam verzí pro různé aplikace, které nejsou zcela bezpečné a jejichž

provoz na síťových zařízeních nelze považovat za bezpečný. V takovém případě je třeba nahradit stávající verzi nějakou novější. To, zda nová verze již není zařazena na seznamu vulnerable aplikací, může správce systému jednoduše zjistit na Internetu.

Skript bude používán především pro testování zařízení v sítích zákazníků. Pokud zaměstnanec IBM provádějící testy zákaznických sítí zjistí nějaký bezpečnostní problém, doporučí správci dané sítě, aby provedl aktualizaci problémové aplikace.

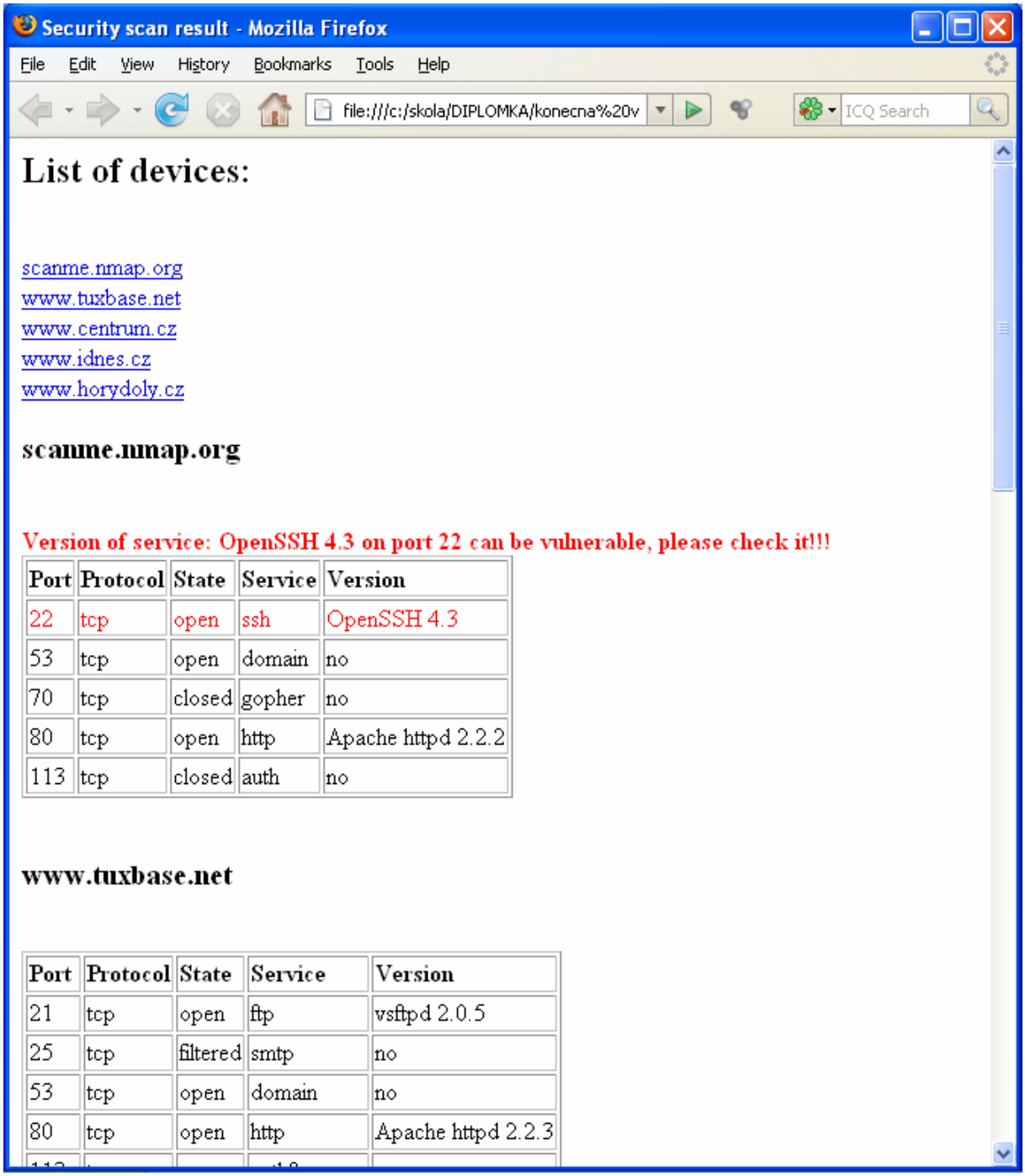
Seznam vulnerable aplikací lze stáhnout z:

`http://nvd.nist.gov/download/nvd_dictionary.txt`

`port_scan.pl` načítá seznam vulnerable aplikací ze souboru nvd_dictionary.txt, který je umístěn ve stejném adresáři jako on sám. Z výstupu programu nmap získá verzi aplikací běžících na skenovaném objektu a ty pak porovnává se seznamem získaným ze souboru nvd_dictionary.txt. Pokud je zjištěno, že některá z aplikací není aktuální, podá o tom informaci ve svém výstupním souboru.

Při spuštění skriptu se do příkazové řádky uvádí soubor, který obsahuje seznam zařízení, které mají být testovány.

Výstupem skriptu `port_scan.pl` je html soubor, který informuje o výsledku bezpečnostního skenu. Aplikace, které jsou považovány za nedostatečně bezpečné jsou zvýrazněny.


Security scan result - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///c:/skola/DIPLOMKA/konecna%20v
ICQ Search
List of devices:
scanme.nmap.org
www.tuxbase.net
www.centrum.cz
www.idnes.cz
www.horydoly.cz
scanme.nmap.org
Version of service: OpenSSH 4.3 on port 22 can be vulnerable, please check it!!!
Port Protocol State Service Version
22 tcp open ssh OpenSSH 4.3
53 tcp open domain no
70 tcp closed gopher no
80 tcp open http Apache httpd 2.2.2
113 tcp closed auth no
www.tuxbase.net
Port Protocol State Service Version
21 tcp open ftp vsftpd 2.0.5
25 tcp filtered smtp no
53 tcp open domain no
80 tcp open http Apache httpd 2.2.3

2.2 Kontrola konfigurace firewall Nokia

V síti IBM je několik firewallů od společnosti Nokia. Jejich konfigurace je vyjádřena v textovém módu a je uložena v souboru initial. Ten obsahuje několik set řádků určujících jednotlivá pravidla.

Prvním krokem před spuštěním mého skriptu je vytvoření html souboru obsahujícího přehledně zpracovanou konfiguraci firewallu. Tato konverze se používá i v praxi, protože usnadňuje hledání informací v konfiguraci. Údaje jsou rozděleny do jednotlivých sekcí, ve kterých jsou uvedeny v tabulkách všechny podstatné informace.

Tento html soubor je vytvořen programem `fw1rules 7.3`. Jedná se o freeware, který je volně k stažení z Internetu. Ten zpracovává do přehledné formy údaje z těchto třech souborů, které obsahují konfigurační údaje firewallu:

- `objects_5_0.C`,
- `rulebases_5_0.fws`,
- `Sernam.W`.

Po vytvoření html souboru může být spuštěn můj skript, který získá všechny podstatné informace se kterými dále pracuje. Název skriptu je:

`check_Nokia_parse_html.pl`

Jako parametry příkazové řádky při spuštění skriptu jsou uvedeny názvy souborů s konfigurací firewallu Nokia, html souborem a názvem výstupního souboru. Výstupní zpráva je ve formátu html a informuje o výsledku kontroly konfigurace firewallu.

Skript se spouští tímto příkazem:

```
perl check_Nokia_parse_html.pl --Nokia_configure_file=initial --html_file=ruleset_name.html
```

Prvním krokem je práce se souborem `initial`, ze kterého jsou získány IP adresy rozhraní firewallu a je kontrolován jeho obsah.

V souboru `initial` je kontrolováno:

- konfigurace TACPLUS,
- zda SSH a telnet využívají TACPLUS.

Při kontrole konfigurace TACPLUS musí být v souboru `initial` obsaženy tyto řádky:

```
aaa:auth_profile:base_TACPLUS_authprofile:tacplus_srv:100:host ip_adresa
aaa:auth_profile:base_TACPLUS_authprofile:tacplus_srv:100:secret heslo
```

Pokud nejsou tyto údaje v souboru uvedeny, jsou vypsány tyto chybové hlášky:

```
C:\diplomka\Nokia>perl check_Nokia_parse_html.pl --Nokia_configure_file=initial
--html_file=ruleset_name.html
Security error in file:"initial" there is not included line:
aaa:auth_profile:base_TACPLUS_authprofile:tacplus_srv:100:host
```

nebo

```
C:\diplomka\Nokia>perl check_Nokia_parse_html.pl --Nokia_configure_file=initial
--html_file=ruleset_name.html
Security error in file:"initial" there is not included line:
aaa:auth_profile:base_TACPLUS_authprofile:tacplus_srv:100:secret
```

Při kontrole zda SSH a telnet využívají TACPLUS musí být v souboru `initial` obsaženy řádky:

```
aaa:profile:base_prof_sshd:auth_profile:1:name base_TACPLUS_authprofile
aaa:profile:base_prof_httpd:auth_profile:1:name base_TACPLUS_authprofile
```

Pokud tyto údaje chybí, vypíše se:

```
C:\diplomka\Nokia>perl check_Nokia_parse_html.pl --Nokia_configure_file=initial
--html_file=ruleset_name.html
Security error in file:"initial" there is not included line:
aaa:profile:base_prof_sshd:auth_profile:1:name base_TACPLUS_authprofile
```

nebo

```
C:\diplomka\Nokia>perl check_Nokia_parse_html.pl --Nokia_configure_file=initial
--html_file=ruleset_name.html
Security error in file:"initial" there is not included line:
aaa:profile:base_prof_httpd:auth_profile:1:name base_TACPLUS_authprofile
```

Po vytvoření seznamu IP adres rozhraní firewallu je načten soubor html. V tomto souboru je tabulka, ve které jsou definovány protokoly pro komunikaci mezi dvěma objekty, které jsou určeny IP adresami. Pokud je jako cílová adresa uvedena IP adresa firewallu a pro komunikace se používají zabezpečené protokoly https a ssh, tak je třeba zkontrolovat

zdrojový objekt. Může jím být host nebo skupina neobsahující síť. V definici zdrojového objektu se nesmí vyskytnout síť.

Tato konfigurace je nařízena z bezpečnostních důvodů, kdy k přístupu do sítě může dojít pouze z předem povolených IP adres a ne ze sítě. Pro útočníka je totiž mnohem jednodušší stát se členem určité sítě než získat konkrétní IP adresu.

Dále se kontroluje, zda žádný objekt vnitřní sítě nevystupuje ve vnější síti s adresou příslušející vnitřní síti. Všechny IP adresy vnitřní sítě musí být před přístupem do vnější přeloženy na jiné. Zde se jedná o standardní překlad adres pomocí NAT.

2.3 Kontrola konfiguračního souboru Cisco routeru

Cisco je největším výrobcem síťových zařízení na světě. Jejich směrovače jsou pravděpodobně nejpoužívanější, a proto se nacházejí i v sítích, které budou kontrolovány mou sadou skriptů.

Pro kontrolu konfiguračních souborů Cisco routerů je určen:

check_configure_file.pl

Jeho účelem není kontrolovat celý konfigurační soubor, ale pouze vybrané pasáže, které jsou z hlediska bezpečnosti nejdůležitější a které mi byly zadány od IBM.

Spuštění skriptu:

```
perl check_configure_file.pl --device_list=seznam-konfiguraku.txt
--output_html=report.html --output_csv=report.csv
```

Při spuštění programu je jako první parametr v příkazové řádce uveden textový soubor obsahující seznam konfiguračních souborů, které mají být zkontrolovány. Na každém řádku je uvedena cesta ke konfiguračnímu souboru. Na počtu položek nezáleží, může jich být velké množství.

Druhým parametrem je název výstupního souboru ve formátu html. Na jeho začátku je uveden seznam testovaných zařízení. Jednotlivé položky jsou odkazy, při jejichž stisknutí se uživatel dostane na zprávu o požadovaného zařízení. Pro testování jedno či dvě zařízení není vzhledem k délce výpisu odkazování potřeba. S ohledem na to, že tímto skriptem bude

kontrolováno velké množství zařízení, je tato funkce rychlého přístupu k danému zařízení uživatelsky přívětivá.

Poslední parametr je název výstupního souboru ve formátu csv. Ten je výhodný zejména kvůli svém jednoduché formě a často se využívá jako vstupní soubor pro zpracování dalšími skripty.

Za seznamem zařízení je v souboru uvedena tabulka obsahující výsledek testu pro jednotlivá zařízení. Pokud zařízení vyhovělo požadavkům na zabezpečení, je ve sloupečku Description uvedeno OK a řádek má modrou barvu.

Pokud je však v konfiguračním souboru bezpečnostní problém, je do sloupce *Description* uveden popis problému. Řádek s chybou má pro zvýraznění červenou barvu.

V následujícím odstavci *Fix* je příkaz, kterým lze chybnou konfiguraci zařízení opravit.

Adresář se spouštěným skriptem obsahuje také soubor `rules_conf.csv`, ve kterém jsou pravidla pro konfiguraci routeru a příkazy pro opravu nedostatečného zabezpečení. Pokud je třeba pro načtení pravidel použít jiný soubor než `rules_conf.csv` použije se při spuštěn přepínač `--config=conf.csv`.

Konfigurační soubor je pak postupně prohledán, zda některou z těchto konfigurací neobsahuje. Pokud ano, je zařízení s touto konfigurací vyhodnoceno jako nedostatečně zabezpečené a je uvedeno ve výpisu na červeném řádku.

Jako první je v konfiguračním souboru testována kvalita hesla. Pokud je použito kryptování podle typu 7, znamená to malé zabezpečení, v konfiguračním souboru je uveden řádek:

```
enable secret 7
```

V takovém případě je vyhodnocen konfigurační soubor jako nevyhovující požadavkům na zabezpečení.

Správně by mělo být uvedené heslo kryptováno podle MD5. Dále je testována jeho síla, tedy to, zda splňuje základní požadavky na sílu hesla. Tento skript kontroluje, za heslo obsahuje:

- velké i malé znaky,
- číslo,
- je delší než osm znaků.

Na kvalitu hesla mohou být kladeny i další požadavky jako speciální znaky, posloupnost písmen stejné velikosti ne delší než čtyři aj.

Úkolem správce je také zřídit dostatečně bezpečný způsob přístupu k zařízení. Chybné je například povolit přístup přes SSH i Telnet. V takovém případě je zbytečné povolovat přístup přes Telnet. SSH je zabezpečené, zatímco Telnet není. Pokud je možné k zařízení přistupovat přes SSH, tak by případné použití Telnetu znamenalo zbytečné bezpečnostní riziko. V takovém případe se doporučuje Telnet deaktivovat.

Pro přístup do snmp community se používá heslo a uvádí se skupina, která má tento přístup povolen. Na stejném řádku je definováno, zda daná skupina může pouze číst, nebo zda má právo i na zápis. Skript testuje, zda v definici skupiny není dovolen přístup komukoliv.

Správná konfigurace snmp community:

```
snmp-server community meR1al6nmp02ro RO 10
access-list 10 permit 20.85.108.13
access-list 10 permit 20.85.108.11
```

Chybná konfigurace snmp community z bezpečnostního hlediska:

```
snmp-server community meR1al6nmp02ro RO 10
access-list 10 permit any
```

Ve výše uvedeném příkladu konfigurace došlo ke konfiguraci snmp community s právy pro čtení (RO) a access list s číslem 10. Heslo je: `meR1al6nmp02ro`. Toto heslo vyhovuje požadavkům na kvalitu hesla.

SNMP (Simple Network Management Protocol) slouží pro spojení mezi dvěma entitami: managerem a agentem. Cisco router je v tomto rozdělení agent, manager je program běžící na počítači. Agent uchovává informace o zařízení v tzv. MIB (Management Information Base). SNMP slouží k výměně informací uvedených v MIB mezi zařízeními. MIB má hierarchickou stromovou strukturu. Objekty v databázi jsou volány přes OID (Object Identifier). Tyto informace jsou vyměňovány mezi objekty v síti, slouží k informování o problémech nebo k vzájemnému dotazování na nastavení. Jedná se výkonný nástroj pro management sítí, který je ale potřeba dostatečně zabezpečit.

Proto se doporučuje nastavit router tak, aby k těmto informacím měli přístup jen určité entity. K tomu slouží `snmp-server community`, kde se definuje access list pro přístup k těmto datům.

SNMP existuje ve třech verzích:

- **SNMPv1** – první verze SNMP, nesplňuje současné požadavky na bezpečnost. Byl určen pro krátkodobé vzdálené spravování sítě.
- **SNMPv2** – má stejně jako předcházející verze příliš nízké zabezpečení, data jsou sítí posílána nešifrovaně.
- **SNMPv3** – obsahuje tři úrovně zabezpečení:
  - ***noAuthNoPriv*** – autentizace nešifrovaně, žádné šifrování komunikace,
  - ***AuthNoPriv*** – autentizace šifrovaně využívající HMAC-MD5 nebo HMAC-SHA, komunikace nešifrována,
  - ***AuthPriv*** - autentizace šifrovaně využívající HMAC-MD5 nebo HMAC-SHA, komunikace šifrována 56-bitovým DES.

Ukázka výstupního html souboru:

List of devices

FRA02-C2950-120-13.cfg
FRA02-C2950-120-15.cfg
FRA02-C2950-120-17.cfg
FRA02-C2950-120-19.cfg

Details:

Device	Description	Fix
FRA02-C2950-120-13.cfg	Disable Cisco Discovery Protocol (CDP) service	router(config)# no cdp run
FRA02-C2950-120-13.cfg	Disable bootp server.	router(config)# no ip bootp server
FRA02-C2950-120-13.cfg	Require login banner	!router(config)# motd-banner YOUR_BANNER
FRA02-C2950-120-13.cfg	Use centralized AAA system.	!router(config)# tacacs-server host IP_ADDRESS;!or ;!router(config)# radius-server host IP_ADDRESS
FRA02-C2950-120-13.cfg	Use AAA authentication methods for login authentication (with fall-back).	router(config)# aaa authentication login default group tacacs+ local enable
FRA02-C2950-120-13.cfg	Use AAA authentication methods for enable authentication (with fall-back).	! router(config)# aaa authentication enable default group tacacs+ enable
FRA02-C2950-120-15.cfg	**OK**	
FRA02-C2950-120-17.cfg	Disable Cisco Discovery Protocol (CDP) service	router(config)# no cdp run
FRA02-C2950-120-17.cfg	Disable bootp server.	router(config)# no ip bootp server
FRA02-C2950-120-17.cfg	Use centralized AAA system.	!router(config)# tacacs-server host IP_ADDRESS;!or ;!router(config)# radius-server host IP_ADDRESS
FRA02-C2950-120-17.cfg	Use AAA authentication methods for login authentication (with fall-back).	router(config)# aaa authentication login default group tacacs+ local enable
FRA02-C2950-120-17.cfg	Use AAA authentication methods for enable authentication (with fall-back).	! router(config)# aaa authentication enable default group tacacs+ enable
FRA02-C2950-120-19.cfg	Disable Cisco Discovery Protocol (CDP) service	router(config)# no cdp run
FRA02-C2950-120-19.cfg	Disable bootp server.	router(config)# no ip bootp server
FRA02-C2950-120-19.cfg	Use centralized AAA system.	!router(config)# tacacs-server host IP_ADDRESS;!or ;!router(config)# radius-server host IP_ADDRESS

V následující kapitole jsou diskutována kontrolovaná nastavení v konfiguračních souborech.

2.4 Bezpečnostní pravidla při konfiguraci Cisco routeru a firewallu

Tato kapitola nepopisuje všechna potřebná nastavení routeru pro zajištění dostatečné bezpečnosti. V této kapitole jsou zmíněny pouze některá. Návody mají sloužit jako úvod do některých oblastí týkajících se bezpečné konfigurace sítě. V této kapitole jsou podrobněji popsána nastavení, která jsou kontrolována ze souboru **`rules_conf.csv`** ve skriptu **`check_configure_file.pl`**.

Vzdálený přístup k routeru nebo firewallu je třeba řádně nakonfigurovat. V níže uvedeném příkladu dochází ke konfiguraci routeru pro připojení přes ssh z access listu číslo 80. Z bezpečnostních důvodů je nastavena i hodnota exec session timeout, která ukončí spojení po určité době nečinnosti.

```
Central(config)# ip telnet source-interface loopback0
Central(config)# access-list 80 permit 12.8.9.1 log
Central(config)# access-list 80 permit 12.8.6.6 log
Central(config)# access-list 80 deny any log
Central(config)# line vty 0 4
Central(config-line)# access-class 99 in
Central(config-line)# exec-timeout 5 0
Central(config-line)# transport input telnet
Central(config-line)# login local
Central(config-line)# exec
Central(config-line)# end
```

V rámci zabezpečení zařízení je třeba pečlivě uvážit, které služby mají být povoleny. Například finger protokol slouží na dotazování přihlášených uživatelů. Je tedy naprosto zbytečné, aby nepřihlášený uživatel takto zjišťoval, kdo je aktuálně k zařízení připojen. Zde je příklad zakázání této služby spolu s výpisem informací před a po deaktivaci protokolu finger:

```
Central# connect 14.2.9.250 finger
Trying 14.2.9.250, 79 ... Open
This is the CENTRAL router; access restricted.
Line User Host(s) Idle Location
```

```
130 vty 0 14.2.9.6 00:00:00 goldfish
*131 vty 1 idle 00:00:00 central
[Connection to 14.2.9.250 closed by foreign host]
Central# config t
Enter configuration commands, one per line. End with CNTL/Z.
Central(config)# no ip finger
Central(config)# no service finger
Central(config)# exit
Central# connect 14.2.9.250 finger
Trying 14.2.9.250, 79 ...
% Connection refused by remote host
Central#
```

V následující části jsou popsány další služby, které je lepší deaktivovat. Pro podrobnější konfiguraci je třeba se podívat do manuálu routeru nebo na internetové stránky výrobce [14].

Na tomto místě je třeba zmínit, že i jedna špatně nakonfigurovaná služba může představovat pro síť velké riziko.

Cisco Discovery Protocol se používá pouze ve speciálních případech, takže při konfiguraci běžných sítí je možné jej vypnout. Slouží pro vzájemnou identifikaci Cisco zařízení v rámci lokální sítě. Pro jeho vypnutí se používá příkaz:

```
no cdp run
```

IP Source routing je prostředek, kterým mohou jednotlivé pakety určovat směrování. Při jeho aktivování na routeru může dojít k několika druhům útoků. Pokud však není daná síť postavena právě na principu tohoto druhu směrování, měl by být IP Source Routing na všech routerech vypnutý, provede se to příkazem:

```
no ip source-route
```

Protokoly TCP a UDP obsahují doporučený seznam jednoduchých služeb, které by měli hostitelé poskytovat. Ve většině případů však není nutné, aby právě tyto služby byly na routerech povoleny. K jejich deaktivaci se používají následující příkazy:

```
no service tcp-small-serv
no service udp-small-serv
```

Další službou, kterou je doporučeno deaktivovat je finger, její podrobnější popis již byl uveden.

```
no ip finger
no service finger
```

Bootp je protokol používaný některými systémy ke stažení operačního systému přes síť nebo dalších konfiguračních parametrů. Cisco routery dokáží pracovat jako Bootp servery pro jiná Cisco zařízení. V praxi se tak může stát, že na jednom zařízení je uložen software IOS pro ostatní. V praxi se tato možnost však používá velice zřídka, takže je doporučeno ji vypnout. Případnému útočníkovi by se v případě jejího používání mohlo podařit stáhnout IOS software pro síťová zařízení. Deaktivace se provede:

```
no ip bootp server
```

Packet assembler/disassembler (PAD) je služba podporující X.25 spojení. Tato volba je implicitně povolena, ale je zbytečné ji mít povolenou v případě, pokud se X.25 v síti nevyužívá. Vypíná se v globálních konfiguračním módu:

```
no service pad
```

Některé verze Cisco routerů dovolují vzdálenou konfiguraci prostřednictvím HTTP (Hyper Text Transfer Protocol) protokolu. Jsou účinným prostředkem pro monitorování, konfiguraci ale i útok na router. Nevýhodou je také textový způsob komunikace, takže například hesla jsou přenášena nekryptovaná. Sám přístup přes HTTP není nezbytně nutný, proto se doporučuje jeho deaktivování:

```
no ip http server
```

Dále se doporučuje dekativovat name server příkazem:

```
no ip name-server
```

Cisco IOS podporuje vyhledávání v DNS záznamech. Standardně je požadavek na zjištění některého z údajů posílán na adresu 255.255.255.255. Pokud byste chtěli tuto služby vypnout, lze tak provést příkazem.

```
no ip domain-lookup
```

Konfiguraci routeru lze nastavit i tak, že zasílá požadavek na přeložení jména v DNS pouze na určitou adresu a nepoužívá broadcast adresu. Pokud je v síti alespoň jeden jmenný server, lze konfiguraci provést těmito příkazy:

```
MyRouter(config)# ip name-server 18.10.1.2 18.7.9.1
MyRouter(config)# ip domain-lookup
```

Lze také nastavit název doménového serveru, to je nezbytné pro zprovoznění SSH v síti:

```
MyRouter(config)# ip domain-name mydnsserver.com
```

Automatické načtení konfigurace (*Configuration Auto-Loading*) dovoluje při spuštění routeru načíst konfiguraci z lokální paměti nebo síťového zařízení. Načítání přes síť je možné povolit pouze v případě, že je celá síť důvěryhodná. Deaktivace této služby se provede příkazy:

```
no boot network
no service config
```

V textu výše je zmíněna kontrola snmp-community prostřednictvím skriptu kontrolujících konfigurační soubory. SNMP však nabízí i spoustu jiných prostředků, které je třeba během konfigurace routeru řádným způsobem ošetřit. Protokol SNMP (Simple Network Management Protocol) je standardní internetový protokol sloužící pro vzdálené monitorování a administraci. Pokud se však SNMP v síti nepoužívá, měly by být na routeru provedeno následující nastavení:

```
Central# show running-config | include snmp
Building configuration...
snmp-server community public RO
snmp-server community admin RW
Central#
```

```
Central# config t
Enter configuration commands, one per line. End with
CNTL/Z.
Central(config)# ! erase old community strings
Central(config)# no snmp-server community public RO
Central(config)# no snmp-server community admin RW
Central(config)#
Central(config)# ! disable SNMP trap and system-shutdown
features
Central(config)# no snmp-server enable traps
Central(config)# no snmp-server system-shutdown
Central(config)# no snmp-server trap-auth
Central(config)#
Central(config)# ! disable the SNMP service
Central(config)# no snmp-server
Central(config)# end
```

Při vypsání konfigurace routeru je pak uvedeno:

```
no snmp-server enable traps
no snmp-server system-shutdown
no snmp-server trap-auth
```

A při úplném vypnutí SNMP:

```
no snmp-server
```

Pro komunikaci po síti je nezbytné znát nejen IP adresu, ale i MAC adresu. Překlad IP adres na fyzické adresy MAC obstarává protokol ARP (Address Resolution Protocol). Vyhledávání fyzické adresy je standardně prováděno pouze v jednom LAN segmentu. Pokud je třeba hledat ve více segmentech, dokáže Cisco router tuto komunikace zprostředkovat. Tato služba se nazývá APR proxy a měla by být povolena pouze na dvou stejně zabezpečených segmentech LAN, aby nedošlo k narušení bezpečnostních pravidel. Jedná se o komunikaci na druhé vrstvě.

Cisco router má implicitně povolen Proxy ARP na všech rozhraních. Vypnutí této služby se provádí na každém rozhraní zvlášť příkazem:

```
no ip proxy-arp
```

Broadcast vysílání je normálně prováděno pouze na jednom segmentu LAN. Hraniční router tyto pakety do dalšího segmentu nepropouští. Mohou se však vyskytnout situace, kdy bude právě toto potřeba (např. NetBios). Zakázání přeposílání broadcast vysílání do dalšího segmentu se provede příkazem:

```
no ip directed-broadcast
```

Internet Control Message Protocol (ICMP) je důležitý pro správné řízení směrování v síti. Zprávy protokolu ICMP jsou běžně generovány a vysílány Cisco routerem. Některé ze zprávy však mohou byt zneužity útočníky. Jedná se o zprávy typu:

- **Host unreachable**,
- **Redirect**,
- **Mask Replay**.

ICMP je však v naprosté většině případů používán pouze pro účel ke kterému byl určen. Zde je tabulka, která popisuje všechny jeho možnosti:

Tab. 2.1 Typy ICMP

Typ	Funkce	Poznámka
0	Echo	
3	Nedoručitelný IP-datagram	Paket byl zahozen - informace pro odesílatele
4	Sniž rychlost odesílání	Síť je někde přetížena
5	Změn směrování	Provede dynamické změny ve směrovací tab.
9	Odpověď na žádost o směrování	
10	Žádost o směrování	
11	Čas vypršel	
12	Chybný parametr	
13	Časová synchronizace	Žádá cílový počítač o čas
17	Maska podsítě	

Tab. 2.2 Obsahu nedoručeného paketu

Kód	Význam
0	síť nedostupná (net unreachable) — směrovač nenašel cestu k síti
1	stanice nedostupná (host unreachable) — nelze se spojit se stanicí
2	protokol nedostupný (protocol unreachahle)
3	port nedostupný (port unreachable)
4	fragmentace nutná, ale byl nastavený příznak DF (Don‘t fragment)
5	špatné zdrojové směrováni (source route failed)

Obr. 2.1 Zapouzdření ICMP paketu

Na rozhraních se sítěmi, které nelze považovat za bezpečné, lze generování zneužitelných zprávy zakázat příkazy:

```
MyRouter(config-if)# no ip unreachables
MyRouter(config-if)# no ip redirects
```

```
Myrouter(config-if)# no ip mask-reply
MyRouter(config-if)# no ip directed-broadcast
```

Maintenance Operations Protocol (MOP) se používal pro systémové utility v DECnet protokolech. Standardně je tato volba povolena na všech rozhraních Ethernetu. Pokud je třeba ji deaktivovat, provede se to příkazem:

```
no mop enabled
```

Network Time Protocol (NTP) se používá pro synchronizaci času na síťových zařízeních. Tuto volbu je vhodné použít, pokud je třeba v záznamech o provozu v síti mít stejný čas na všech zařízeních.

V některých sítích však není NTP provozován, a tak je zbytečné mít jej mít povolený. Vypnutí této služby se provádí na jednotlivých rozhraních příkazem:

```
ntp disable
```

Ukázka výpisu konfigurace routeru pro dvě rozhraní:

```
interface eth 0/0
description Outside interface to 14.1.0.0/16 net
no ip proxy-arp
no ip directed-broadcast
no ip unreachable
no ip redirect
no mop enabled ntp disable
interface eth 0/1
description Inside interface to 14.2.9.0/24 net
no ip proxy-arp
no ip directed-broadcast
no ip unreachable
no ip redirect
no mop enabled
ntp disable
interface loopback0


description Loopback interface for service bindings
no ip proxy-arp
no ip directed-broadcast
no ip unreachable
```

Authentication, Authorization, Accounting (AAA) jsou metody, které se používají při ověřování uživatele během přístupu k zařízení a zaznamenávají jeho činnost. Authentication je postup pro identifikaci uživatelů před přístupem k síťovým zařízením. Authorization slouží k definování toho, co má uživatel povoleno. Accounting umožňuje zaznamenávání činnosti uživatele.

Zprávy z logů mohou být použity pro síťový management, bezpečnostní analýzu nebo třeba reporty. Logy ze zařízení jsou odesílány na server. Pro nahrávání logů ze zařízení jsou povoleny pouze servery TACACS+ a RADIUS.

Existuje několik typů AAA accountings:

- **Network accounting** – informace z protokolů PPP, SLIP, ARAP o počtu přenesených paketů a bajtů.
- **EXEC accounting** – informace o EXEC spojení na routeru. Obsahuje uživatelské jméno, čas začátku a konce spojení a IP adresu.
- **Connection accounting** – spojený směřující pryč provedené síťového access serveru.
- **Command accounting** – zaznamenává příkazy zadané do EXEC shellu. Loguje informace o uživateli, času a prováděném příkazu.
- **System** – uchovává systémové zprávy.

3. Laboratorní úlohy

3.1 Seznámení s honeyd a nmap

Laboratorní úloha je zaměřené na seznámení s démonem **honeyd** [4] a programem **nmap** [3]. V první části úlohy provedou studenti skeny různých serverů na Internetu a seznámí se tak se základními parametry programu nmap.

Úvodní část nedefinuje servery, které mají studenti prozkoumávat, proto bude jen na nich, které se zvolí. Mohou tak v praxi zjistit, že spousta objektů v síti je nastavena tak, aby na tyto dotazy neodpovídaly. Na druhou strany mohou zjistit, že správci některých serverů nechávají spuštěných zbytečně moc portů. Toho mohou využít útočníci pro průnik do systému.

Studenti mají za úkol stáhnout seznam aplikací, které nejsou považovány za bezpečné. Jedná se o soubor:

`http://nvd.nist.gov/download/nvd_dictionary.txt`

Poté budou analyzovat výpisy z první části laboratorní úlohy a hledat spuštěnou službu, jejíž provozování není považováno za bezpečné. V době psaní této práce jsem našel neaktuální verzi SSH na serveru `www.horydoly.cz`.

V druhé části laboratorní úlohy si studenti vyzkouší práci s démonem `honeyd`. Ten se používá především ve firemních sítích, kde slouží jako virtuální síť, která má za úkol odlákat útočníky od skutečných cílů. Zařízení v této virtuální síti mohou být servery i routery. Jsou na nich definovány služby a porty jimž jsou přiřazeny. Dále může být nastaven operační systém a další volby. Tyto parametry jsou přiřazeny určité IP adrese. Pokud útočník provádí sken těchto objektů, může si myslet, že se jedná o skutečná zařízení. Monitorovací systémy tak získávají informace o útočníkovi a při následujícím pokusu o přístup do sítě je komunikace s ním odmítnuta. Jedná se tak o velice efektivní a především levný způsob, jak chránit síť před útočníky. V některých sítích je falešná síť postavena ze skutečných zařízení. Toto řešení je ovšem finančně náročné, protože vyžaduje velké investice do hardware i software.

Honeyd obsahuje konfiguraci těchto zařízení v souboru:

`honeyd.conf`

Démon je spuštěn skripty:

```
./start-arpd.sh
./start-honeyd.sh
```

Po jejich spuštění lze ověřit běh virtuálních systémů pomocí pingu na cílové objekty. Po tomto prověření následují skeny cílových objektů programem nmap s různými parametry. Studenti mají za úkol vyzkoušet různé parametry při jeho spuštění a analyzovat jeho výstupy. Získají tak informace o cílovém objektu, které mají za úkol uvést v závěru laboratorní úlohy.

V poslední části budou studenti měnit konfigurační soubor `honeyd.conf` a tím se seznamovat více se spuštěným démonem. Mohou tak realizovat vlastní představy a zkoušet možnosti, které tato aplikace umožňuje.

3.2 Cisco PIX 501

V této laboratorní úloze se studenti seznámí se zařízením Cisco PIX 501 patřícím do skupiny Cisco PIX Firewalls (více [14]). Používá se pro zabezpečení menších počítačových sítí v soukromém i firemním prostředí. Je určen pro vysokorychlostní spoje, které jsou užívány v nepřetržitém provozu. Jeho výhodou je jednoduchá správa, uživatelská přívětivost a malé rozměry.

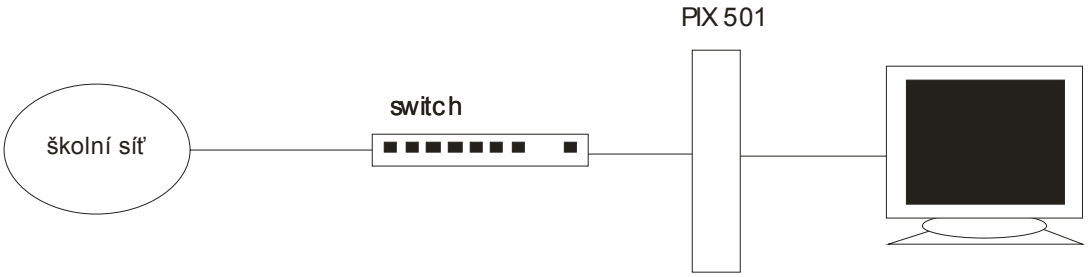
Technická specifikace:

- čtyřportový 10/100 FastEthernet přepínač,
- procesor 133-MHz AMD SC520,
- 16 MB SDRAM,
- 8 MB Flash paměť,
- Sběrnice – single 32 bitová a 33 MHz PCI.

Poskytuje spoustu služeb, které přispívají k bezpečnosti sítě. Poskytuje tyto funkce:

- stavový firewall,
- kontrola obsahu paketů pro jednotlivé protokoly,

- VPN,
- NAT,
- PAT,
- SNMP,
- DHCP,
- detekce průniků,
- ochrana multimediálních přenosů,
- ochrana přenosu hlasu.



Obr. 3.1 Zapojení laboratorní úlohy

Firewall využívá Cisco Adaptive Security Algorithm, který podporuje stavovou kontrolu paketů. Snaží se tak zabránit neoprávněným přístupům do vnitřní sítě například z Internetu. Obsahuje pravidla pro několik desítek aplikačních protokolů a aplikací, které jsou použity při prověřování komunikace stavovým firewallem.

Zařízení lze konfigurovat více způsoby. Zkušení administrátoři většinou preferují konzolovou konfiguraci. Ta je u PIX prováděna přes sériový port. Tento způsob jsem zvolil i v laboratorní úloze. Studenti se s ním z velkou pravděpodobností setkají v praxi. Firewall je možné konfigurovat také přes webové rozhraní využívající Javu. Osobně preferuji první variantu pro její přehlednost.

Pro vzdálený přístup může být využit i Cisco Secure Policy Manager, který je součástí Cisco VPN/Security Management Solution.

V praxi se často využívá kombinace překladu síťových adres a portů. Počítačová síť pak může používat pouze jednu veřejnou IP adresu. Výhodou tohoto řešení je finanční úspora,

protože provozovatel počítačové sítě nemusí platit velké množství veřejných IP adres. Druhou výhodou je počítače v Internetu nevystupují se svou vlastní IP adresou.

Na začátku laboratorní úlohy se studenti seznámí se způsobem výpisu konfigurace zařízení. Jejich úkolem je výpis důkladně prostudovat a pochopit význam jednotlivých příkazů. V této části využijí příkazů:

- `show configure`
- `show interface`
- `show version`

Zařízení je již předem nakonfigurováno, což je poznat právě na výpisech konfigurace. Pro uvedení zařízení do firemního nastavení je v další části zadat příkaz:

```
pixfirewall(config)#configure factory-default
```

Tím jsou do restartu zařízení vymazána všechna použitá nastavení. Studenti dále nastaví IP adresu počítače na:

```
IP adresa:        192.168.1.2
maska:            255.255.255.0
gefault gateway:  192.168.1.1
DNS server:       147.229.144.10
```

Nyní vyzkoušejí, zda je možné prohlížet webové stránky nebo provést ping za firewall. To se jim pochopitelně nepodaří a jejich úkolem v následujících minutách bude tyto služby zprovoznit. Toho dosáhnou vhodnou konfigurací firewallu, na kterém tento druh provozu povolí.

Studenti si tímto v praxi vyzkoušejí hlavní důvod použití firewallu – omezení, resp. povolení určitého druhu provozu pro omezený počet IP adres z vnitřní sítě.

Druhý úkolem studentů je použít DHCP u Cisco PIX 501. Vyzkoušejí si tak jednu z mnoha funkcí, které zařízení dovoluje použít.

Závěr

S počítačovými systémy se dnes setkáváme takřka na každém kroku. Jsou používány ve firmách, státní správě, univerzitách a v posledních letech velice často i v domácnostech. Každá z těchto skupin využívá počítačové sítě k odlišným účelům, přesto jsou některé jejich potřeby shodné. Vedle požadavku na bezproblémový chod, rychlé připojení do LAN a Internetu je s rostoucím množstvím informací skladovaných v elektronické podobě stále více diskutována otázka bezpečnosti. Přístup k informačním zdrojům je třeba chránit, aby je mohli užívat pouze oprávnění uživatelé. Do této problematiky se zahrnuje nejen oprávnění ke čtení nebo změně uložených dat, ale i ochrana informací během přenosu po síti.

Ve své práci jsem se zabýval problematikou zabezpečení počítačových sítí a jeho kontrolou. Sestavil jsem soubor několika skriptů, které tyto úkony automatizují. Jsou určeny pro IBM Global Services Delivery Centrum Brno a jejich úkolem není kontrolovat celkové zabezpečení počítačových sítí, ale soustředit se na předem vymezené oblasti.

Jeden ze skriptů provádí skenování spuštěných aplikací na cílových objektech. Obsluha pak rozhodne, zda nejsou některé aplikace spuštěny zbytečně. Dále je třeba sledovat, zda nejsou některé z nich neaktualizované, což by bylo z bezpečnostního hlediska nepřípustné.

Pro přístup do vnitřní sítě je doporučeno použít firewall. Jeho hlavním úkolem je zamezit neoprávněnému přístupu z vnější sítě. U IBM se používají firewally společnosti Nokia a v rámci kontroly zabezpečení je třeba důkladně kontrolovat jejich konfiguraci. Jeden z přiložených skriptů je určen právě pro tuto kontrolu.

Další skript se zabývá kontrolou konfigurace Cisco routerů. Je v něm ověřováno několik předem definovaných pravidel, jejichž dodržování je z hlediska bezpečnosti nezbytné.

V následující kapitole je podrobně rozepsáno, jaké konfigurace Cisco zařízení mohou být z bezpečnostních důvodů rizikové. Zařízení od firmy Cisco jsou v současnosti nejpoužívanější a proto jsem si v této kapitole věnoval právě jim.

V příloze práce jsou umístěny dvě laboratorní úlohy, ve kterých se studenti seznámí s programy honeyd a nmap a firewallem Cisco. V první úloze budou pracovat s démonem honeyd, který lze použít pro testování nebo jako falešný systém pro zmatení útočníka. V druhé úloze se studenti seznámí s konfigurací firewallu Cisco PIX 501, který se používá pro zabezpečení menších počítačových sítí.

Bezpečnost počítačových sítí je rozsáhlá oblast, kterou se zabývá velké množství počítačových odborníků. S rostoucím objemem dat uchovávaných v elektronické podobě se zvyšují požadavky na jejich zabezpečení. Výzkum v této oblasti neustále pokračuje a bude určitě zajímavé sledovat jeho výsledky. Po skončení studia na vysoké škole bych se rád věnoval právě bezpečnosti a konfiguraci počítačových sítí.

Seznam použité literatury

[1] Dostálek, L. a kolektiv *Velký průvodce protokoly TCP/IP Bezpečnost.* Computer Press, ISBN 80-7226-849-X, ČR, 2003

[2] McClure, S., Scambray, J., Kurtz, J. *Hacking bez tajemnství*. Computer Press, ISBN 80-722-6948-8, ČR, 2003

[3] *Nmap Reference Guide*, [online] [cit. 2008-04-08]
URL: <http://www.insecure.org/nmap/man/>

[4] *Honeyd*
URL: < http://www.honeyd.org> [cit. 2008-03-02].

[5] IANA, *Port Numbers*, [online] [cit. 2008-05-17]
URL: <http://www.iana.org/assignments/port-numbers>

[6] Karel Burda *Bezpečnost informačních systémů.* VUT v Brně, ČR, 2007

[7] NOVOTNÝ, V. *Architektura sítí*. FEKT VUT v Brně, 2002.

[8] *Windows Security*
URL: <www.windowsecurity.com>

[9] *Network Security Journal*
URL: <www.networksecurityjournal.com>

[10] *CERT*
URL: <www.cert.org>

[11] Matt Curtin *Introduction to Network Security.*
URL: <www.interhack.net/pubs/network/security>

[12] *Wikipedia - VPN*
URL: < http://en.wikipedia.org/wiki/Virtual_private_network> [cit. 2008-01-05].

[13] Vanessa Antoine, Raymond Bongiorni a kolektiv *Router Security Configuration Guide.* National Security Agency, 2005

[14] *Cisco*, zabezpečení síťových prvků
URL: < http://www.cisco.cz> [cit. 2008-04-26].

[15] Ivo Herman *Pokročilé komunikační techniky*. VUT v Brně, ČR, 2007

Seznam Příloh

Přílohy

Laboratorní úloha - Seznámení s honeyd a nmap

Popis úlohy

Laboratorní úloha je zaměřena se na seznámení s programy honeyd a nmap. Oba jsou často používány správci systémů i hackery. Honeyd slouží pro simulaci síťových zařízení, nmap pro skenování. Cílem práce studentů je vytvořit programem honeyd virtuální hosty a ty poté testovat s různými parametry programu nmap.

Vybavení pracoviště

PC s OS Linux a instalací nmap a honeyd

Úkoly

1. Seznamte se s programem nmap.
2. Seznamte se s programem honeyd.
3. Skenujte síťová zařízení simulované v prostředí honeyd programem nmap.

Teoretický úvod

Honeyd

Honeyd je démon, který vytváří virtuální hosty na síti. Host může být nakonfigurován jako libovolná služba nebo třeba proxy, která běží na určitém operačním systému. Honeyd umožňuje aby měl jeden host více IP adres.

Jedná se o open source projekt, který zaštiťuje GNU General Public Licence. Jeho hlavním výhodou je schopnost testování situací, které mohou v síti nastat.

Poskytuje mechanismy, kterými lze detekovat průnik do systému nebo třeba maskování reálných systémů prostřednictvím virtuálních.

Honeyd podporuje testování prostřednictvím příkazu ping nebo traceroute. Každá služba provozovaná v honeyd je definovaná jednoduchým textovým souborem. Ten díky své jednoduchosti umožňuje rychlou konfiguraci hostů.

Honeyd může být použit pro virtuální honeyd sítě nebo pro monitorování sítě. Umožňuje vytvoření virtuální síťové topologie včetně specifických směrování. To může být definováno například zpožděním nebo ztrátovostí, což činí vytvořený model reálným.

Zde je vlastnosti démona honeyd:

- Simulace tisíců virtuálních hostů.
- Nastavování libovolných vlastností u aplikací:
  - spojení přes proxy,

    - pasivní fingerprinting pro identifikování vzdálených hostů.
- Simulování operačního systému používajícího protokolovou sadu TCP/IP:
- Nastavování libovolných směrovacích topologií:
    - nastavení zpoždění a ztrátovosti,
    - asymetrické směrování,
    - začlenění hardwarových zařízení do topologie,
    - distribuce Honeyd prostřednictvím GRE tunelování.
- Virtualizace subsystémů:
    - spouštění skutečných unixových aplikací pod IP adresami v Honeyd, napař. webserver, ftp server nebo jiné,
    - dynamické přiřazování portů ve virtuální adresovém prostoru, vytváření spojení na pozadí aj.

Jednou z nevýraznějších výhod tohoto démona je možnost spouštět aplikace operačního systému UNIX na IP adresách Honeydu. Tyto aplikace pak mohou navazovat síťová spojení, přiřazovat porty apod. Subsystém zachytává pakety směřující na tyto adresy a předává je programu honeyd, který je dále doručí cílové aplikaci. Tímto systémem může být simulován například provoz na pozadí jako je čtení emailů nebo stahování web stránek.

V prostředí tohoto programu je umožněno simulovat hardwarová zařízení a definovat jejich směrování, zpoždění či jiné vlastnosti, jako je tomu v konfiguračním souboru níže:

```
route entry 10.0.0.1 network 10.0.0.0/8
route 10.0.0.1 link 10.0.0.0/24
route 10.0.0.1 add net 10.4.0.0/14 tunnel "thishost" "honeyd-b"
route 10.0.0.1 add net 10.1.0.0/16 10.1.0.1 latency 55ms loss 0.1
route 10.0.0.1 add net 10.2.0.0/16 10.2.0.1 latency 20ms loss 0.1
route 10.0.0.1 add net 10.3.0.0/16 10.2.0.1 latency 20ms loss 0.1
route 10.1.0.1 link 10.1.0.0/24
route 10.2.0.1 link 10.2.0.0/24
[...]
route 10.2.0.1 add net 10.3.0.0/16 10.3.0.1 latency 10ms loss 0.1
route 10.3.0.1 link 10.3.0.0/24
route 10.3.0.1 add net 10.3.1.1/24 10.3.1.1 latency 10ms
route 10.3.0.1 add net 10.3.240.0/20 10.3.240.1 latency 5ms
route 10.3.1.1 link 10.3.1.1/24
route 10.3.240.1 link 10.3.240.0/20
route 10.3.240.1 add net 0.0.0.0/0 10.3.0.1 latency 40ms loss
0.5
[...]
bind 10.2.0.243 to fxp0
bind 10.3.1.15 to fxp0
```

Nmap

Nmap ("Network Mapper") je open source, který slouží k prozkoumávání sítě. Je vhodný pro skenování rozsáhlých sítí. Jeho prostřednictvím lze získat velké množství informací o cílovém síťovém prvku. Jedná se především o seznam spuštěných aplikací, jejich verzi a port na kterém běží nebo operační systém.

Program je spouštěn z příkazové řádky a nabízí velké množství nastavitelných parametrů. Jejich seznam je uveden v příloze. Podrobně je lze prostudovat na stránkách `http://insecure.org/nmap`, odkud lze tento program také stáhnout. V této laboratorní úloze se seznámíte se základními druhy skenů.

Postup

1) Spusťte si příkazovou řádku a proveďte ping na několik serverů v Internetu. Máte za úkol najít alespoň jeden, který na ping odpovídá, a jeden, který na ping neodpovídá. Zdůvodněte.

2) Proveďte sken několika serverů v Internetu pomocí příkazu nmap. Výpis programu z libovolného si uložte a v závěru okomentujte. Zjistěte verzi běžících služeb a použitý operační systém.

   Např. `nmap –sV –O www.idnes.cz`

3) Stáhněte si soubor:

   http://nvd.nist.gov/download/nvd_dictionary.txt

V něm jsou uloženy verze služeb, které nejsou považovány za bezpečné. Analyzujte předcházející výpisy a pokuste se najít službu, která je náchylná k prolomení (vulnerable).

4) Nmap umožňuje skenování jen určitých portů. Tento postup je výhodný zvláště při testování velkého počtu zařízení, u kterých nás zajímá spuštění pouze některých služeb. Jeho volby nastudujte v příloze a uveďte zápis tohoto příkazu.

5) Další možností je používat v nmapu pro skenování buď UDP nebo TCP, nastudujete.

V druhé části laboratorních prací budete pracovat s démonem honeyd. Jedná se o program, který umožňuje simulovat síťová zařízení a především vytvářet virtuální sítě v reálné síti. Toho lze využít například pro zmatení útočníků, kteří by se snažili napadnout počítačovou síť. Budou pak útočit na virtuální cíle, ve kterých nemohou napáchat škody. Během jejich činnosti o nich systém nasbírá dostatečné množství informací a při opětovném přístupu nebude útočník do systému vůbec vpuštěn.

6) Zkopírujte obsah souboru `honeyd_puvodni.conf` do souboru `honeyd.conf`, pokud nejsou oba soubory totožné.

7) Spusťte tyto dva skripty:

   `./start-arpd.sh`

```
./start-honeyd.sh
```

Tím je spuštěn honeyd s konfigurací uloženou v souboru `honeyd.conf`.

8) Zkontrolujte, zda jsou cílové objekty spuštěny:

```
ping 147.229.151.73
ping 147.229.151.74
ping 147.229.151.75
```

9) Proveďte základní nmap sken pro všechny tři zařízení a analyzujte výstup:

```
nmap 147.229.151.73
nmap 147.229.151.74
nmap 147.229.151.75
```

10) Zjistěte operační systém na cílových zařízeních:

```
nmap -O 147.229.151.73
nmap -O 147.229.151.74
nmap –O 147.229.151.75
```

11) Standardně jsou skeny prováděny s protokolem TCP. Někdy je třeba použít UDP, k tomu slouží přepínač –sU.

```
nmap –sU 147.229.151.73
nmap -sU 147.229.151.74
nmap –sU 147.229.151.75
```

12) Ve zbývajícím čase nastudujte soubor `honeyd.conf` a zkuste jej změnit. Proveďte stejné skeny, které jste spustili na stroji s původní konfigurací.

13) V závěru uveďte co nejvíce informací o systémech, které jste skenovali jako první.

14) Vraťte změny v souboru `honeyd.conf`, případně jej nahraďte souborem `honeyd_puvodni.conf`, který je uložen ve stejném adresáři.

Kontrolní otázky

1. K čemu se využívá program `nmap`?
2. K čemu se využívá program `honeyd`?
3. Proč je program `honeyd` oblíbený?

```
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sP: Ping Scan - go no further than determining if host is online
  -P0: Treat all hosts as online -- skip host discovery
  -PS/PA/PU [portlist]: TCP SYN/ACK or UDP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idlescan
  -sO: IP protocol scan
  -b <ftp relay host>: FTP bounce scan
  --traceroute: Trace hop path to each host
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080
  -F: Fast - Scan only the ports listed in the nmap-services file)
  -r: Scan ports consecutively - don't randomize
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=safe,intrusive
  --script=<lua scripts>: <lua scripts> is a comma separated list of dirs or scripts
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database. Only performed if -sC or --
script was also given.
OS DETECTION:
  -O: Enable OS detection (try 2nd generation w/fallback to 1st)
  -O2: Only use the new OS detection system (no fallback)
  -O1: Only use the old (1st generation) OS detection system
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
```

```
TIMING AND PERFORMANCE:
  Options which take <time> are in milliseconds, unless you append 's'
  (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  -T[0-5]: Set timing template (higher is faster)
  --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
  --min-parallelism/max-parallelism <time>: Probe parallelization
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
      probe round trip time.
  --max-retries <tries>: Caps number of port scan probe retransmissions.
  --host-timeout <time>: Give up on target after this long
  --scan-delay/--max-scan-delay <time>: Adjust delay between probes
FIREWALL/IDS EVASION AND SPOOFING:
  -f; --mtu <val>: fragment packets (optionally w/given MTU)
  -D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
  -S <IP_Address>: Spoof source address
  -e <iface>: Use specified interface
  -g/--source-port <portnum>: Use given port number
  --data-length <num>: Append random data to sent packets
  --ip-options <options>: Send packets with specified ip options
  --ttl <val>: Set IP time-to-live field
  --spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
  --badsum: Send packets with a bogus TCP/UDP checksum
OUTPUT:
  -oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
     and Grepable format, respectively, to the given filename.
  -oA <basename>: Output in the three major formats at once
  -v: Increase verbosity level (use twice for more effect)
  -d[level]: Set or increase debugging level (Up to 9 is meaningful)
  --open: Only show open (or possibly open) ports
  --packet-trace: Show all packets sent and received
  --iflist: Print host interfaces and routes (for debugging)
  --log-errors: Log errors/warnings to the normal-format output file
  --append-output: Append to rather than clobber specified output files
  --resume <filename>: Resume an aborted scan
  --stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
  --webxml: Reference stylesheet from Insecure.Org for more portable XML
  --no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
  -6: Enable IPv6 scanning
  -A: Enables OS detection and Version detection
  --datadir <dirname>: Specify custom Nmap data file location
  --send-eth/--send-ip: Send using raw ethernet frames or IP packets
  --privileged: Assume that the user is fully privileged
  --unprivileged: Assume the user lacks raw socket privileges
  -V: Print version number
  -h: Print this help summary page.
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sP 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -P0 -p 80
```

Laboratorní úloha - Firewall Cisco PIX 501

Úkoly

1. Seznamte se zapojením pracoviště.
2. Seznamte s firewallem Cisco PIX 501.
3. Nastavte pravidla na firewallu povolujících prohlížení internetových stránek a provedení pingu.
4. Nastavte DHCP na firewallu Cisco PIX 501.

Vybavení pracoviště

PC s OS Windows, Cisco PIX 501, propojovací kabely

Úvod

Firewall je síťové zařízení, které slouží k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti nebo zabezpečení. Zjednodušeně se dá říct, že slouží jako kontrolní bod, který definuje pravidla pro komunikaci mezi sítěmi, které od sebe odděluje.

Používá se tam, kde je třeba omezit nebo zakázat určitý druh provozu. Používá se ve velkých podnikových sítích i v domácnostech. V obou případech jsou na ně kladeny jiné nároky, ale vždy ma za úkol chránit vnitřní síť před neoprávněným přístupem z vnější sítě. Při konfiguraci sítě je nutné dbát na to, aby všechna spojení vnitřní sítě s vnější sítí probíhala přes firewall. Pokud by se tak nestalo, nebyla by síť dostatečně chráněná. Nejjednodušší je směřovat všechna spojení do a ze sítě přes jeden firewall.

Rozhraní firewallu se dělí:

- vnitřní,
- vnější,
- rozhraní pro DMZ.

V nastavení firewallu jsou definována určitá pravidla, kdy určitá událost může spustit nějakou akci (alert). Akcí se rozumí spuštění nebo ukončení nějakého programu, např.:

- odeslání zprávy o určité události – prostřednictvím SMS, emailu apod.,
- ukončení proxy nebo filtru, na kterém došlo k události,
- zapsání IP adresy potenciálního útočníka na černou listinu,
- ukončení práce celého systému,
- spuštění určitého programu.

Existuje několik typů firewallů:

- paketové filtry,
- aplikační brány,

- stavové paketové filtry,
- stavové paketové filtry s kontrolou protokolů a IDS.

Paketové filtry určují z jaké adresy a portu na jakou adresu a port může být doručen procházející paket. Výhodou tohoto řešení je rychlost, nevýhodou naopak nízká kontrola zabezpečení spojení.

Příkladem tohoto systému je paketový filtry typu ACL (Access Control List) u Cisco routerů.

Aplikační brány jsou někdy také nazývány Proxy firewall. Jejich úkolem je úplné oddělení dvou sítí. Veškerá spojení probíhají 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 přijatá aplikační branou od serveru jsou předána klientovi.

Při použití aplikační brány se používá NAT – překlad síťových adres. Server tak nezná IP adresu klienta, ale pouze IP adresu firewallu.

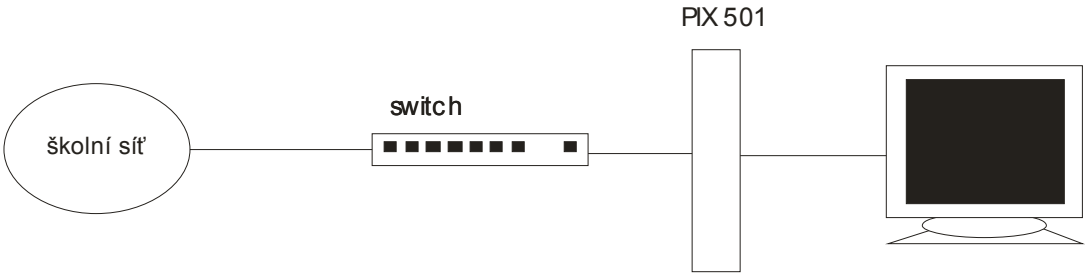
Kontrola se provádí na 7. úrovni OSI modelu. Výhodou tohoto řešení je vysoká efektivnost, nevýhodou zase vysoké požadavky na HW a SW.

Stavové paketové filtry provádějí stejnou kontrolu jako paketové filtry. Narozdíl od nich ale ukládají informace o povolených spojeních. Toho se využívá při kontrole příchozích dat, u kterých se zjišťuje, zda patří k již existujícímu spojení nebo k nějakému novému.

Stavové paketové filtry s kontrolou protokolů a IDS umožňují dynamicky otevírat porty pro různá řídící a datová spojení i 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 nejedná o požadavek na WWW server, ale tunelování jiného protokolu, což často využívají klienti P2P sítí.

Umožňují také použití IDS (Intrusion Detection Systems – systémy pro detekci útoků). Ty dokáží odhalit vzorce útoků i ve zdánlivě nesouvisejících pokusech o spojení.

Zapojení



Obr. 1 Zapojení laboratorní úlohy

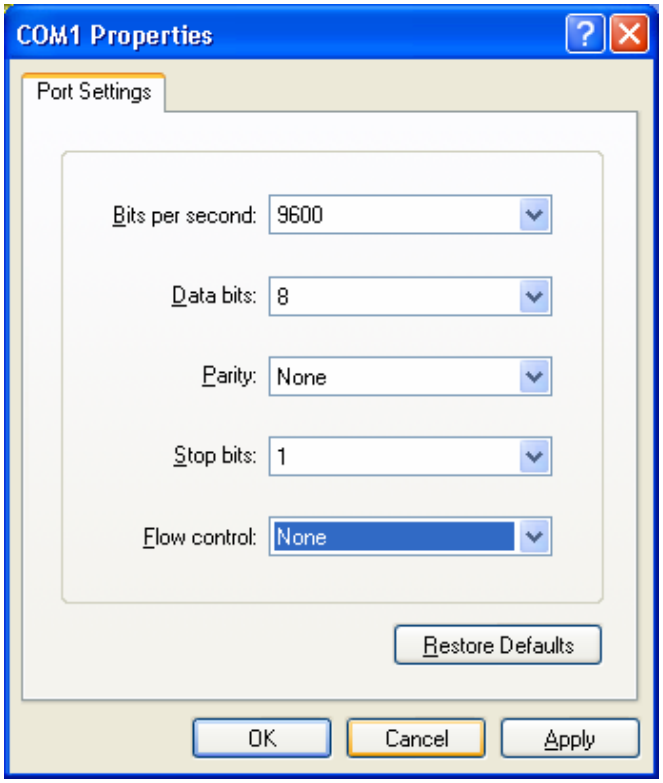
Postup

1. Zkontrolujte zapojení laboratorní úlohy podle obrázku. Ujistěte se, že firewall je připojen do zdroje el. energie, switche a síťové karty počítače.
2. Dále zkontrolujte připojení firewallu se sériovým portem počítače světle modrým plochým kabelem. Toto spojení se využívá pro konfiguraci zařízení. Spusťte Hyperterminál:

Start->Programy->Příslušenství->Komunikace->HyperTerminal.

Hyperterminál je součástí Windows a vy jej využijete ke konzolové práci s firewallem. Příkazy budete zadávat v textové formě, která je preferována zkušenými administrátory pro svou přehlednost.

Nyní je třeba vytvořit spojení mezi sériovým portem počítače a firewallem. Spojení libovolně pojmenujte a vyberte sériový port, do kterého je PIX připojen. Další požadované položky vyplňte podle následujícího obrázku.



Obr. 2 Konfigurace spojení přes sériové port

3. Pokud se vám spojení podařilo navázat, tak v terminálovém okně vidíte po stisku klávesy Enter úvodní prompt:

```
pixfirewall>
```

4. Nyní se nacházíte v neprivilegovaném režimu. Pro přístup do privilegovaného je třeba zadat příkaz:

```
pixfirewall>enable
```

a heslo:

```
cisco
```

Seznam dostupných příkazů můžete kdykoliv vyvolat příkazy:

- `help`
- `?`

Do konfiguračního módu se přistupuje příkazem:

```
pixfirewall#configure terminal
```

Zpět do privilegovaného režimu se přistupuje příkazem

```
pixfirewall(config)#exit
```

5. Nyní se pomocí výše zmíněných příkazů dostaňte do privilegovaného módu a nechte si vypsal konfiguraci:

```
show configure
show version
```

Výpis podrobně prostudujte. Pokud některým konfiguracím nerozumíte, podívejte se na internetové stránky `www.cisco.com`, kde jsou vysvětlena jednotlivá nastavení.

Dále se podívejte na stav jednotlivých rozhraní:

```
show interface
```

Výpis opět důkladně prostudujte.

6. Nastavte následující parametry ve vlastnostech připojení počítače:

```
IP adresa:        192.168.1.2
maska:            255.255.255.0
gefault gateway:  192.168.1.1
DNS server:       147.229.144.10
```

V konfiguračním módu firewall zadejte následující příkaz:

```
pixfirewall(config)#configure factory-default
```

Tím se firewall dostane do firemního nastavení, kdy nejsou uplatňována nakonfigurovaná pravidla.

7. Ověřte komunikace počítače s okolím. Spusťte Explorer a otevřete nějakou internetovou stránku, proveďte ping na jiný počítač v laboratoři.

Ani v jednom z těchto pokusů nebude komunikace navázána, protože není na firewallu povolena.

8. Příkazem `sh int` zkontrolujte, že adresa ethernet1 „inside“ na firewallu je 192.168.1.1. Nyní nakonfigurujeme access list, kterým povolíme komunikace pro surfování na Internetu a spuštění pingu. Access list poté přiřadíme do access group:

```
pixfirewall(config)# access-list muj permit icmp any any
pixfirewall(config)# access-list muj permit tcp 192.168.1.0 255.255.255.0 any eq www
pixfirewall(config)# access-group muj in interface outside
```

9. Nyní se pokuste v Exploreru podívat na libovolnou stránku. Proveďte ping na sousední počítač nebo do Internetu.

Komunikace proběhne v obou případech standardním způsobem, protože je již povolena na firewallu.

10. Nyní nakonfigurujete přístup na firewall prostřednictvím Telnetu.

```
pixfirewall(config)# telnet 192.168.1.2
```

ip adresa udává počítač, ze kterého se bude možné připojit

```
pixfirewall(config)# password heslo89
```

nastavení hesla pro připojení přes Telnet

Z příkazové řádky počítače se připojte Telnetem na firewall:

```
C:\>telnet 192.168.1.1
```

Pozn. Komunikace prostřednictvím Telnetu probíhá v nezabezpečené formě.

11. Druhá část laboratorní úlohy slouží k ukázce použití firewallu jako **DHCP serveru**. PIX 501 tuto možnost nabízí a tak vaším následujícím úkolem bude tuto funkci ověřit.

    Nastavte na počítač ve vlastnostech připojení, aby získával IP adresu automaticky pomocí DHCP:

    `Obtain an IP address automatically`

12. Přes konzoli se na firewallu přepněte do konfiguračního režimu a zadejte následující příkazy. Nejprve nastavte rozsah IP adres pro vnitřní síť:

```
pixfirewall(config)# ip add inside 147.229.151.97 255.255.255.248
```

13. Nastavte IP adresu dvou DNS serverů:

```
pixfirewall(config)#dhcpd dns 147.229.72.10 147.229.3.10
```

14. Nastavte rozsah IP adres, které budou DHCP serverem přiřazovány:

```
pixfirewall(config)#dhcpd address 147.229.151.98-147.229.151.99 inside
```

15. Nastavte adresy DNS serverů:

`pixfirewall(config)`#dhcpd dns 147.229.72.10 147.229.3.10

16. Nastavte dobu, po kterou bude IP adresa přidělena (údaj je v sekundách):

`pixfirewall(config)`#dhcpd lease 3000

17. Spusťte démon DHCP serveru na firewallu:

`pixfirewall(config)`#dhcpd enable inside

18. Klikněte na ikonu Internetového připojení na počítači ve vnitřní síti a zvolte Obnovit (Repair).
19. Zkontrolujte síťová nastavení příkazem `ipconfig` v příkazové řádce počítače ve vnitřní síti.

Kontrolní otázky

1. K čemu se využívá firewall a jaké jsou největší výhody jeho použití?
2. Co je to NAT a proč se používá?
3. Znáte nějakého výrobce SW firewallů?