VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITORING ZABEZPEČENÍ LAN SÍŤĚ

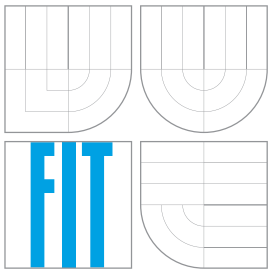
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MATĚJ GRÉGR

BRNO 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITORING ZABEZPEČENÍ LAN SÍŤĚ

LAN SECURITY MONITORING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MATĚJ GRÉGR
AUTHOR

VEDOUCÍ PRÁCE — Ing. PAVEL OČENÁŠEK
SUPERVISOR

BRNO 2007

Zadání bakalářské práce/4841/2006/xgregr01

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Grégr Matěj**
Obor: Informační technologie
Téma: **Monitoring zabezpečení LAN sítě**
Kategorie: Počítačové sítě

Pokyny:

1. Seznamte se s principy a protokoly používanými při komunikaci v rámci lokální sítě (LAN).
2. Analyzujte požadavky na zabezpečení koncových připojených prvků (např. počítačů) na různých vrstvách ISO/OSI. Seznamte se s útoky realizovatelnými v takové síti.
3. Navrhněte aplikaci, která bude monitorovat dostupnost a vybrané bezpečnostní parametry připojených zařízení. Návrh proveďte pomocí UML.
4. Do návrhu zahrňte také statistiky dostupnosti a zabezpečení a specifické upozorňování administrátora sítě a/nebo uživatele sítě.
5. Navrženou aplikaci implementujte a její funkčnost otestujte.
6. Implementujte webové rozhraní tak, aby bylo možné získat aktuální a statistické údaje také na bráně připojené do Internetu.
7. Diskutujte získané znalosti a možnosti dalšího rozšíření projektu.

Literatura:

- C.Hunt: TCP/IP Network Administration. O'Reilly Press, 2002.
- A.S.Tanenbaum: Computer Networks, Forth Edition, Prentice Hall, 2003.
- F.Halsall: Computer Networking and the Internet, Fifth Edition, Addison Wesley, 2005.
- Další související internetové zdroje

Při obhajobě semestrální části projektu je požadováno:

- Body 1 - 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Očenášek Pavel, Ing.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2
L.S.

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Matěj Grégr**
Id studenta: 84390
Bytem: U nemocnice 864, 757 01 Valašské Meziříčí
Narozen: 16. 12. 1984, Chomutov
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Monitoring zabezpečení LAN sítě
Vedoucí/školitel VŠKP: Očenášek Pavel, Ing.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:
tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☐ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

....

Nabyvatel Autor

Abstrakt

Tato práce se zabývá bezpečností a monitorováním dostupnosti portů v sítích LAN. Popisuje útoky, které mohou nastat v sítích LAN a obranu vůči nim. Zabývá se také technikami skenování portů. V praktické části je implementována sada funkcí, které mohou být použity k testování dostupnosti portů. Implementace je realizována pomocí jazyku C, knihovnami Libnet, Pcap a sadou PHP a bash skriptů.

Klíčová slova

LAN, bezpečnost, monitorování, útoky v sítích LAN, skenování portů

Abstract

This bachelor thesis deals with security in LAN networks and monitoring security and availability of selected ports. The thesis gives a brief overview of the LAN attacks, defense techniques that are used to prevent them and port scanning techniques. In the practical part of my thesis I have implemented the set of functions collected in library. These functions are ready to be used for testing of ports availability and analysis of open services. The implementation is realized in C language, Libnet and Pcap libraries and bash scipts and the web control panel is implemented in HTML and PHP.

Keywords

LAN, security, monitoring, LAN attacks, port scanning

Citace

Matěj Grégr: Monitoring zabezpečení LAN síťě, bakalářská práce, Brno, FIT VUT v Brně, 2007

Monitoring zabezpečení LAN síťě

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Pavla Očenáška

.....
Matěj Grégr
11. května 2007

Obsah

Kapitola 1

Úvod

Počítače jsou nasazovány do více a více odvětví. Kvůli potřebě sdílet mezi sebou data jsou spojovány do sítí. Vzrůstající počet těchto sítí s sebou nese také zvyšující se riziko různých útoků, které mohou způsobit odcizení dat nebo například výpadek sítě. Je nutné tedy klást čím dál tím větší důraz na bezpečnost a monitorování těchto sítí, jelikož následky různých útoků mohou způsobit nemalé finanční škody.

Za hlavní bezpečnostní rizika můžeme v sítích LAN[1] považovat viry[2], trojské koně[3] a hackery[4]. Důsledkem těchto útoků je potenciální ztráta nebo poškození dat, případně výpadek nebo zahlcení sítě.

1.1 Cíl bakalářské práce

Hlavním cílem bakalářské práce je seznámit se s principy a protokoly používanými při komunikaci v rámci lokální sítě. Analyzovat požadavky na zabezpečení koncových připojených prvků (např. počítačů) a seznámit se s útoky realizovatelnými v takové síti. Dále navrhnout aplikaci, která bude monitorovat dostupnost připojených zařízení.

V kapitole 2. je popsána síťová architektura pomocí referenčních modelů. Kapitola 3 se zabývá popisem základních protokolů a principů komunikace v sítích LAN. Kapitola 4 se věnuje problematice útoků v sítích LAN a obraně proti nim. V kapitole 6 jsou popsány techniky skenování portu a 7. kapitola se věnuje vlastní implementaci.

[1] Local Area Network

[2] programy šířící se bez vědomí uživatele

[3] program, který předstírá užitečnost, ale obsahuje skrytý kód, který může ohrozit počítač

[4] počítačový odborník, který se pokouší proniknout do počítačových systémů

Kapitola 2

Síťová architektura

Jelikož komunikace v sítích je komplexní, obtížná záležitost a navrhnout ji tak,aby správně fungovala je problém, používá se dekompozice problému. Tedy jeho rozdělení do více snadněji zvládnutelných podproblémů, tzv. vrstev. Aby byla komunikace univerzální je nutná potřeba jednoho standardu pro vzájemné propojení síťových prvků a jejich komunikaci. Vznikly tak modely síťové komunikace.

2.1 Referenční model ISO/OSI

Standard s názvem Reference Model of Open Systems Interconnection (Referenční model propojování otevřených systémů) zkráceně RM OSI nebo ISO/OSI byl přijat mezinárodní organizací pro standardizaci ISO. Definuje 7 vrstev (viz. obrázek 2.1) [7]

Obrázek 2.1: Referenční model ISO/OSI

Fyzická vrstva (Physical Layer)

Tato vrstva zajišťuje fyzickou komunikaci mezi zařízeními. Definuje navázání a ukončení spojení a veškeré další elektrické a fyzikální vlastnosti zařízení (velikost napěťové úrovně logické 1, konektory atp.).Příkladem může být protokol X.25.

Linková vrstva (Data Link Layer)

Linková vrstva zabezpečuje bezchybný přenos bloků dat (rámců) mezi dvěma body počítačové sítě. Kontroluje, jestli rámce byly přeneseny správně. Příkladem je protokol X.25 nebo LAPB[1].

Síťová vrstva (Network Layer)

Vrstva č. 3. Stará se o směrování packetů mezi více uzly sítě. Zajišťuje volbu vhodné cesty mezi mezilehlými uzly v síti. Příkladem mohou být protokoly CLNS[2] nebo PLP[3].

Transportní vrstva (Transport Layer)

Tato vrstva zprostředkovává při odesílání rozdělení dat na jednotlivé pakety a při přijímání je skládá ve správném pořadí do původního tvaru. Zajišťuje přenos libovolně dlouhých zpráv. Příkladem jsou protokoly TP0, TP1, TP2, TP3, TP4[4]

Relační vrstva (Session Layer)

Úkolem této vrstvy je navazování, udržování a rušení spojení (relací) mezi koncovými účastníky. Dále také synchronizace a obnovení spojení a oznamování výjimečných stavů. Příkladem je např. protokol X.225.

Prezentační vrstva (Presentation Layer)

6. vrstva. Zajišťuje nezávislost aplikacím z pohledu prezentace dat. Účelem je dosáhnout stejné reprezentace dat na více platformách.Příkladem může být ISO 8327 a X.225.

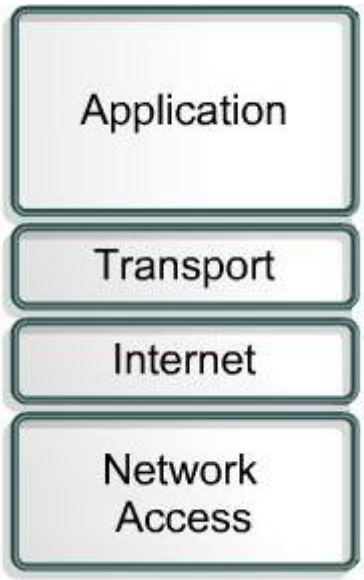
Aplikační vrstva (Application Layer)

Účelem této 7. vrstvy je zajistit interakci s uživatelem a umožnit aplikacím přístup ke komunikačnímu systému. Příkladem jsou protokoly X.500 DAP[5]

[1] Link Access Protocol, Balanced
[2] Connectionless Network Service
[3] Packet-Layer Protocol
[4] Transport Layer Protocols Class 1,2,3,4
[5] Directory Access Protocol

2.2 Referenční model TCP/IP

Na rozdíl od modelu ISO/OSI je model TCP/IP rozdělen do 4 vrstev (viz. obrázek 2.2). Byl vytvořen na zakázku pro ministerstvo obrany USA, které si ho nechalo vyvinout, aby pro potřeby válečného ohrožení získalo decentralizovaný, robustní systém, nezávislý na médiu.



Obrázek 2.2: Referenční model TCP/IP

Vrstva fyzického rozhraní (Network Access Layer)

V modelu ISO/OSI by odpovídala fyzické a linkové vrstvě. Nejnižší vrstva, která umožňuje přístup k fyzickému přenosovému médiu. Je specifická pro každou síť v závislosti na její implementaci. Příklady sítí: Ethernet, Token ring, PPP, FDDI.

Internetová (Síťová) vrstva (Internet Layer)

Plní funkci síťové vrstvy modelu ISO/OSI. Zajišťuje tedy síťovou adresaci a směrování datagramů. Její součástí jsou např. protokoly ARP a RARP (neoperují přímo na síťové vrstvě, ale mezi vrstvou síťovou a fyzickou), ICMP, IPv4, IPv6, IGMP a další.

Transportní vrstva (Transport Layer)

Odpovídá transportní vrstvě ISO/OSI. Vytváří logické spojení mezi procesy a zajišťuje přenos mezi dvěma koncovými účastníky. Transportní vrstva zahrnuje protokoly TCP a UDP.

Aplikační vrstva (Application Layer)

Zahrnuje relační, prezentační a aplikační vrstvu modelu ISO/OSI. Aplikační protokoly přímo komunikují s transportní vrstvou a využívají jejích služeb. Pro rozlišení aplikačních protokolů se používají tzv. porty. Příkladem aplikačních protokolů jsou např. Telnet, FTP, HTTP, DHCP, DNS, SSL, IRC, NNTP, IMAP, POP3 a další.

Kapitola 3

Protokoly, porty

3.1 Porty

Protokol TCP a UDP používá k rozlišení aplikací čísla portů. Číslo portu je 16-bitové bezznaménkové číslo. Existuje tedy 65535 různých portů, které mohou být přiděleny aplikaci. Porty se dělí do 3 skupin. Dobře známé porty (well-known ports) jsou v rozsahu 0 – 1023, registrované porty (registered ports) v rozsahu 1024 – 49151 a dynamické/privátní porty (dynamics/private ports) s rozsahem 49152 – 65535. Seznam, které aplikace běží na kterém portu, je spravován organizací IANA[1]. Privátní porty jsou určeny pro nejpoužívanější aplikace. Aplikace používající registrované porty by je měla zaregistrovat u IANA. Dynamické/privátní jsou určeny pro soukromé použití. Porty slouží k rozlišení síťových služeb a upřesnění identifikace pro odesílatele a příjemce. Kdyby se nepoužívaly, na počítači (serveru apod.) by mohla běžet pouze jediná síťová služba (např. FTP). Každá komunikace v síti je určena zdrojovým a cílovým portem.

3.2 UDP

Protokol UDP[2] je protokolem transportní vrstvy modelu TCP/IP. Slouží k přenášení datagramu mezi počítači. Nezaručuje doručení, pořadí paketů a zda pakety nedorazí vícekrát. U protokolu UDP se neustanovuje spojení (connectionless), má menší režii v hlavičce paketu a je bezestavový, což je vhodné pro servery, které odpovídají na velké množství jednoduchých dotazů. Je používán pro aplikace u kterých nevadí občasná ztráta paketů. Např. streamovaná média, přenos hlasu, videa, online hry.

3.3 TCP

TCP[3] protokol se používá u většiny aplikací. Pracuje na transportní vrstvě modelu TCP/IP. Protokol garantuje také spolehlivé doručení, doručení paketů ve správném pořadí a správnost dat, která je zabezpečena kontrolním součtem. Komunikace probíhá ve 3 fázích. První je navazování spojení, následuje komunikace a poslední fází je ukončení spojení. Nazývá se také “three-way handshake”. Při navazování spojení se posílá paket s příznakem SYN. Druhá strana odpoví paketem s nastavenými příznaky SYN a ACK. Navazovatel potvrdí spojení

[1] Internet Assigned Numbers Authority
[2] User Datagram Protocol
[3] Transmission Control Protocol

paketem s nastaveným příznakem ACK a spojení je navázáno. Příkladem jsou například protokoly HTTP, SSH, IMAP a další.

3.4 ARP

Pokud chtějí v ethernetové síti mezi sebou komunikovat 2 body (počítače), potřebují znát fyzickou (ethernetovou, MAC) adresu bodu se kterým chtějí komunikovat. Vyšší vrstvy (vrstva síťová) ale používají pro adresaci IP adresu. Protokol ARP[4] slouží ke získání fyzické adresy z adresy IP. Počítač, který získává MAC adresu druhého počítače posílá broadcastem dotaz ARP Request. Tento dotaz obdrží všechna zařízení v dané podsíti. Porovnají IP adresu se svojí a pokud nastala shoda posílají odpověď ARP Reply, jinak paket zahodí. Počítač, který chtěl komunikovat nyní zná cílovou MAC adresu a může poslat data. Aby se nemusel dotazovat před každým pokusem o odeslání dat, ukládá si MAC adresu na určitou dobu do tzv. ARP Cache. Po vypršení této doby je záznam vymazán a překlad musí proběhnout znovu.

3.5 DHCP

Pro identifikaci v síťového rozhraní zařízení se používá protokol IP. Nezbytné nastavení síťového rozhraní musí obsahovat IP adresu, síťovou masku a případně IP adresu brány pro přístup do jiné sítě. IP adresu můžeme nakonfigurovat staticky nebo dynamicky. Pro dynamickou konfiguraci slouží protokol BOOTP (starší, už málo rozšířený) a DHCP[5]. Pomocí DHCP můžeme nastavit více parametrů než BOOTP protokolem, např. IP adresu, masku podsítě, primární a sekundární DNS server, maximální dobu přidělení IP adresy (lease time) a další. Klient, který komunikuje na UDP portu 68, se připojí do sítě a vyšle broadcastem DHCP Discover, na který server, naslouchající na portu 67, odpovídá paketem DHCP Offer, jenž obsahuje nabizené IP adresy a další nastavení. Klient si vybere z nabídky a pošle serveru DHCP Request, ve kterém jej žádá o vybranou adresu. Server potvrzuje přijetí volby pomoci DHCP Ack.

3.6 FTP

Protokol pro přenos souborů, FTP[6], je protokolem aplikační vrstvy modelu TCP/IP. K přenosu souborů využívá, pro svou spolehlivost a garanci doručení paketů ve správném pořadí, protokolu TCP. Naslouchá na portu 21 (příkazy). Port 20 slouží k posílání dat. Přenos může být pasivní nebo aktivní. Při aktivním přenosu jsou přenášena data z portu 20 a připojení pro přenos navazuje server. Při pasivním přenosu klient požádá server o přepnutí do pasivního režimu. Server oznámí klientovi číslo datového portu pro příjem dat a FTP klient otevírá datový port pro příjem dat.

[4] Address Resolution Protocol
[5] Dynamic host configuration protocol
[6] File Transfer Protocol

3.7 ICMP

ICMP[7] je protokolem internetové vrstvy modelu TCP/IP2.2. Pracuje nad vrstvou IP a slouží ke zpětné vazbě pro protokol IP. Každý uzel, který má implementovaný protokol IP, jej musí podporovat. Primárním účelem tohoto protokolu je předávání síťových informací, zejména chybových hlášení. Nejpoužívanějšími ICMP datagramy jsou:

Typ	Zpráva
0	Echo Reply
3	Destination Unreachable
5	Redirect Message
8	Echo Request
11	Time Exceeded
13	Timestamp Request
14	Timestamp Reply

Kombinaci ICMP zprávy typu 8, kódu 0(Echo Request) a zprávy typu 0, kódu 0 (Echo Reply) používá příkaz ping k jednoduchému otestování dostupnosti síťového uzlu.

ICMP zpráva typu 5 (Redirect) se používá k optimalizaci routování paketů po síti. Existují 4 podtypy (kódy). Nejrozšířenější je kód 1, Redirect datagrams for the Host, který používá uzel k informování zdrojového uzlu, že existuje lepší cesta pro posílání paketů.

3.8 NetBIOS

Zkratka NetBIOS znamená Network Basic Input/Output System. Tento softwarový interface (API) vyvinula firma IBM a později jej převzala společnost Microsoft. NetBIOS umožňuje přístup k síťové komunikaci na úrovni relační vrstvy modelu ISO/OSI (2.1). V nynějších sítích je používán NetBIOS implementovaný jak nad protokoly TCP, tak UDP. Používají se hlavně tyto služby[3]:

- **Name service**: NetBIOS pracuje na relační vrstvě. Nepoužívá proto adresy uzlů síťové vrstvy, ale vlastních logických jmen, která jsou v podobě až 16 znakových řetězců a stanice si je mohou samy volit a měnit. Pokud si počítač hodlá přiřadit/změnit určité jméno, informuje o tom pomocí broadcastu a rozhraní NetBIOS ostatní počítače v síti. Pokud mu nikdo v časovém limitu neodpoví (požadované jméno vlastní), smí ho nastavit jako své logické jméno. Z tohoto principu přidělování logických jmen je patrné, že na větších sítích LAN může dojít k zahlcování sítě broadcast dotazy a také že NetBIOS nelze použít v sítích WAN (L3 zařízení jako např. router filtrují broadcast). Proto se používá komunikace se jmenným serverem. Ten bývá implementován např. pomocí WINS[8] nebo DDNS[9]. Jména se registrují a vyhledávají u name serveru a ten brání jejich duplicitě a také urychluje přidělení nebo změnu jména. Můžeme použít tyto základní příkazy: `Add Name`, `Add Goup Name`, `Delete Name` a `Find Name`. Name Service pracuje na UDP a TCP portu 137.
- **Session service**: je spolehlivé, spojované (connection-oriented) propojení 2 NetBIOS aplikací. Využívá TCP portu 139. Na začátku spojení se posílá paket `SESSION`

[7]Internet Control Message Protocol
[8]Windows Internet Name Service
[9]Dynamic DNS

`REQUEST`, kterým se žádá o zahájení relace (session). Na tento paket může přijít odpověď `POSITIVE SESSION RESPONSE` (pozitivní odpověď, TCP spojení je přijato pro přenos dat), `NEGATIVE SESSION RESPONCE` (negativní odpověď, TCP spojení je odmítnuto a ukončeno) a `SESSION RETARGET RESPONSE` (relace je přípustná, ale musí být na jiné IP adrese a portu). Po navázání a ustanovení spojení mohou být posílána/přijímána data. Pro detekci selhání připojení jsou posílány také keep-alive pakety. Základní příkazy této služby: `Call`, `Listen`, `Hang Up`, `Send`, `Receive`, `Session Status`.

- **Datagram service**: je nespolehlivé, nespojované (connectionless) spojení. Pracuje na UDP portu 138. Každý paket je posílán samostatně a jeho maximální velikost je 512 bajtů. Kontrola doručení musí být implementována v aplikaci. Datagramy mohou být poslány na specifické jméno, nebo jako broadcast. Základní příkazy: `Send Datagram`, `Send Broadcast Datagram`,`Receive Datagram`, `Receive Broadcast Datagram`.
- **Miscellaneous functions**: jsou ostatní funkce NetBIOSu pro kontrolu síťového zařízení. Základní příkazy: `Reset`, `Cancel`, `Adapter Status`, `Unlink`.

Kapitola 4

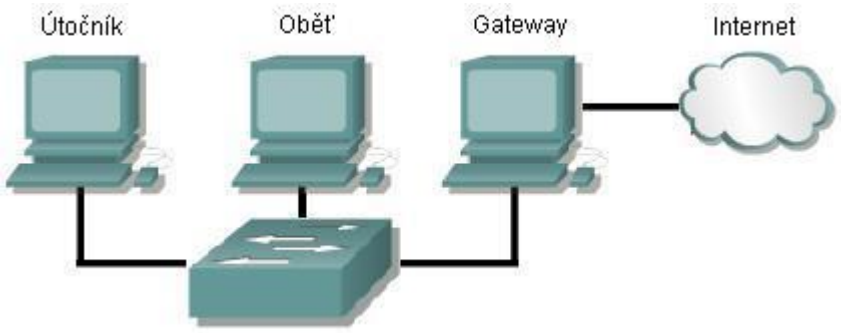
Útoky v sítích LAN

V sítích LAN je potencionálně největší hrozbou odposlech dat a jejich zneužití. Jelikož je nyní většina sítí realizována skrz routery a switche, je nutné vůbec získat možnost data odposlouchávat. K tomuto účelu se používají různé techniky.

4.1 ARP Cache poisoning

Útok

V 3.4 je popsáno fungování ARP protokolu. Jelikož ARP se jako většina protokolů vyvíjel v době, kdy nebyl kladen velký důraz na bezpečnost, můžeme využít některých jeho slabin. Když si protokol ARP uloží záznam do své cache paměti, je relativně snadné změnit hodnotu MAC adresy. Protokol si totiž nehlídá, jestli o data žádal, a pomocí ARP Reply můžeme tedy změnit záznam v cache tabulce. Tento útok může být využit následujícím způsobem.



Pro demonstraci si zvolíme IP a MAC adresy počítačů.

Počítač	IP adresa	MAC adresa
Gateway	176.10.16.1	AA:AA:AA:AA:AA:AA
Oběť	176.10.16.2	BB:BB:BB:BB:BB:BB
Útočník	176.10.16.3	CC:CC:CC:CC:CC:CC

Před začátkem útoku vypadá ARP Cache table oběti a brány takto:

Oběť			Gateway		
176.10.16.1	=	AA:AA:AA:AA:AA:AA	176.10.16.2	=	BB:BB:BB:BB:BB:BB
176.10.16.3	=	CC:CC:CC:CC:CC:CC	176.10.16.3	=	CC:CC:CC:CC:CC:CC

Nyní útočník odešle ARP Reply paket. Příjemce bude Gateway a v paketu uvede, že MAC adresa Oběti je CC:CC:CC:CC:CC:CC. Další ARP Reply paket útočník pošle Oběti a uvede, že Gateway má MAC adresu CC:CC:CC:CC:CC:CC. ARP Cache tabulky nyní budou vypadat takto:

Oběť			Gateway		
176.10.16.1	=	CC:CC:CC:CC:CC:CC	176.10.16.2	=	CC:CC:CC:CC:CC:CC
176.10.16.3	=	CC:CC:CC:CC:CC:CC	176.10.16.3	=	CC:CC:CC:CC:CC:CC

Pokud nyní bude oběť chtít komunikovat s internetem (bránou), zašle data počítači útočníka. Ten je může analyzovat a poslat teď už na bránu. Pokud bude brána odpovídat na dotaz oběti, pošle svůj paket také útočníkovi, ten jej opět zanalyzuje a přepošle oběti. Pomocí tohoto útoku má útočník přehled o kompletní komunikaci mezi obětí a internetem.

Obrana

Obrana proti tomuto útoku se odvíjí od implementace sítě. Mohou být zavedeny statické položky v ARP Cache tabulce. Statickým položkám nevyprší platnost jako u položek dynamických a nemohou být změněny pomocí ARP Reply paketu. Toto je ale pro větší sítě neefektivní. Existuje i obrana přímo na switchi, např. DAI[1], která využívá DHCP Snoopingu viz.4.2 a tabulek, které jsou díky němu vytvářeny. Pokud přijde dotaz ARP Request z untrusted portu, je zkontrolováno, zda MAC a IP počítače žádajícího o překlad IP adresy patří k sobě. Pokud to je paket ARP Reply, kontroluje se i zda k sobě patří IP a MAC počítače, který odpovídá na ARP Request. Dále se tomuto útoku dá bránit např. ignorováním ARP Reply paketu, pokud k němu neexistuje ARP Request, nebo pravidelným porovnáváním ARP Cache se zapamatovanou hodnotou.

4.2 DHCP Spoofing

Útok

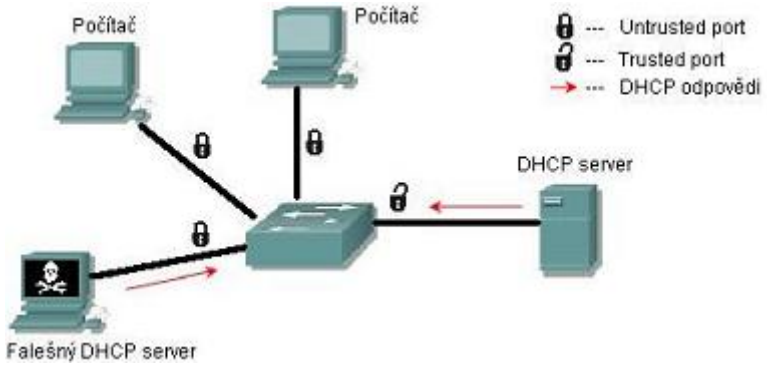
Jak funguje protokol DHCP je popsáno zde 3.5. Při DHCP Spoofingu se využije toho, že na síti může být více DHCP serverů. Aby oběť dostala paket DHCP Offer od útočníkova DHCP serveru, tak útočník musí odpovědět rychleji na dotaz DHCP Discover. Toho může docílit stálým zasíláním paketu DHCP Offer, nebo také vyčerpáním adres, které DHCP server může nabídnout. Pokud DHCP server nemůže přidělit žádné volné IP adresy, neodpovídá na DHCP Discover a na tento paket může odpovědět falešný DHCP server a podvrhnout tak údaje, posílané v paketu DHCP Offer jako např. gateway nebo DNS servery.

Obrana

Bránit se proti tomuto útoku lze použitím tzv. DHCP Snoopingu. Princip je ten, že porty switche jsou rozděleny na trusted a untrusted (viz 4.1). Administrátor definuje, za kterými porty se nachází DHCP server nebo jiný switch, a tento port označí jako trusted. Pokud je na portu připojen počítač, je port označen jako untrusted. DHCP odpovědi jsou pak povoleny jen na trusted portech. Pokud bude falešný DHCP server chtít zaslat DHCP odpovědi, switch zjistí, že odpověď přišla z untrusted portu a paket zahodí. Této techniky se ještě využívá k získání informací z DHCP paketů a vybudování tabulky, kde jsou uvedeny

[1] Dynamic ARP Inspection

vazby mezi MAC adresou, IP adresou, dobou přidělení IP adresy a portem přepínače. Této tabulky se pak využívá k další obraně.



Obrázek 4.1: DHCP Snooping

4.3 MAC flooding

Útok

Switch posílá data pouze tomu počítači, kterému jsou určena. Jelikož pracuje na linkové vrstvě, kam poslat data zjišťuje podle MAC adres v hlavičce linkového rámce. K tomu, aby věděl, na který svůj port má data poslat, využívá CAM[2] tabulku. Zde si ukládá, která MAC adresa patří ke kterému portu. Tato tabulka může obsahovat tisíce položek. Po zaplnění CAM tabulky závisí na implementaci switche, jak bude postupovat dále. Některé by měly dále fungovat jako HUB, tzn. přeposílat data na všechny své porty kromě příchozího. Další se mohou chovat jako HUB, dokud nepřijde paket s MAC adresou, kterou ve své CAM tabulce mají. Ten pak pošlou pouze na cílový port. Útočník tedy vygeneruje množství paketů a zaplní CAM tabulku switche. Jelikož záznamy v CAM tabulce mají pouze omezenou platnost a po určité době se mažou, může toho využít a vložit do CAM tabulky své podvržené údaje.

Obrana

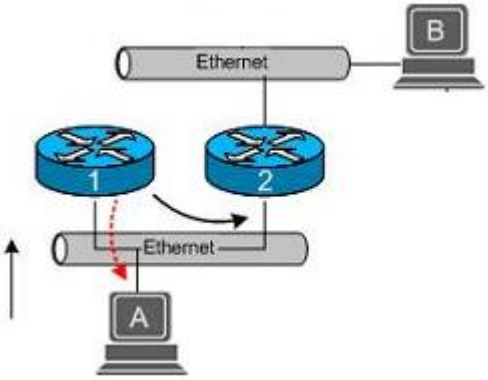
Na obranu proti MAC flooding se používá několik technik. Definování oprávněných MAC adres, které mohou používat port. Stejně jako u statických položek v ARP Cache, ani zde není nastavení omezení MAC adresy na port použitelné řešení pro větší sítě, kde je tento způsob administrátorsky nezvládnutelný. Některé switche umožňují nastavit maximální počet MAC adres, které se switch může naučit z daného portu. Další řešení je v kombinaci a využití DHCP Snoopingu. Z počátku je blokována veškerá komunikace kromě DHCP paketů. Počítač připojený k portu dostane platnou (kontrolováno DHCP Snoopingem) IP adresu. Po obdržení IP adresy je na portu vytvořen Access list, který zajistí omezení provozu pouze na počítače, které mají přidělenu důvěryhodnou adresu. Jakýkoliv provoz s adresou, která není v access listu, je odfiltrován.

[2]Content Addressable Memory table

4.4 ICMP Redirect

Útok

ICMP Redirect využívá ICMP zprávy typu 5 kódu 1. Touto zprávou informuje router odesilatele, že existuje lepší cesta k cíli než přes něj a odesílatel by si tedy měl upravit svou routovací tabulku.



Obrázek 4.2: ICMP Redirect

Na obrázku 4.2 chce počítač A poslat poslat data počítači B. Počítač A má nastaveno standardní směrování do sítě na router 1. Když router 1 přijme paket s cílovou adresou počítače B, zjistí podle své routovací tabulky, že paket musí přeposlat routeru 2, který je předá počítači B. Dále zjistí, že cílové rozhraní a síť , přes které musí odeslat data, je stejné jako rozhraní, na kterém data přjal. Pošle tedy počítači A paket ICMP Redirect, ve kterém ho informuje o tom, že data odesílaná počítači B by měla být posílána na router 2. Sníží se tak počet skoků (hops) a vytížení routeru 1. Postup při útoku je stejný jako na příkladu výše. Útočník využije zfalšovaných paketů ICMP a může tak poškodit topologii sítě, nebo se vydávat za defaultní bránu a odposlouchávat tak data oběti.

Obrana

V dnešní době většinu ICMP paketů blokují firewally. Lze mu také zabránit podobně jako MAC Floodingu pomocí IP Source Guardu. Podvržený ICMP paket totiž potřebuje mít zfalšovanou zdrojovou IP adresu, což ale IP Source Guard neumožní a data budou na switchi zahozena. Záleží také na použitém OS. V ICMP RFC 792 [4] je definován ICMP paket typu 5 tak, že musí obsahovat prvních 64 bitů z paketu, který ICMP Redirect vyvolal. Ne vždy je však toto vyžadováno. Musíme si také uvědomit, že ICMP Redirect by se ve správně navržené topologii sítě neměl vůbec objevit.

4.5 Denial of Service

Denial of Service (odmítnutí služby, dále pouze DoS) jsou útoky, které mají za cíl znepřístupnit danou službu, server nebo síť. Znepřístupnit danou službu může útočník např. zahlcením linky oběti. Tedy ve vygenerování co největšího datového toku. Pokud je linka uživatele zahlcena, pravidla firewallu ho neochrání, protože linka bude zahlcována pořád. Pravidla musí být nastavena u providera, což v sobě zahrnuje problém komunikace aj. Navíc

útok může být jednoduše modifikován změnou útočících IP adres. K zahlcení můžeme použít jak protokoly UDP a TCP, tak např. protokol ICMP.

Nynější servery jsou připojeny k internetu tak rychlými linkami, že jeden počítač je může těžko zahltit. Proto se používají distribuované útoky DoS (DDoS). Dříve to bylo o domluvě několika útočníků. V dnešní době je trend využívání botnetů.

Botnet jsou sítě čítající od několika desítek po tisíce počítačů. Na všech těchto počítačích bývá nainstalován program (bot), který dokáže plnit příkazy z jiného počítače (nejčastěji zasílané přes protokol IRC). Počítače bývají označovány jako *zombies*. Útočník pak používá těchto zombies k záplavovému útoku. Obrana bývá velice obtížná, jelikož je útok prováděn z velkého množství IP adres. U providera musí dojít k zablokování těchto adres nebo k zablokování celých skupin adres. V nedávné době byl takto veden útok třeba na zákazníky společnosti Netgroup, u kterých má server např. i jabber.cz.

DoS útoků může být daleko víc. SYN Flood, který zasílá pakety s nastaveným příznakem SYN. Server na tyto pakety odpovídá paketem s příznaky SYN+ACK. Při poslání velkého množství paketů je server zahlcován nejen příchozími spojeními, ale také odchozími, jak se snaží na dané pakety odpovídat.

Lokální DoS útoky mohou využít i příliš striktní bezpečnostní politiky. Ve firemní LAN síti, kde po několika neúspěšných přihlášeních je účet uživatele zablokován, může být během útoku zablokovaná celá řada uživatelských účtů. Zaregistrováním e-mailové adresy na velkém počtu zahraničních reklamních serverů nebo pornoserverů může vést k zahlcení a zaplnění e-mailové schránky oběti. DoS útoky také využívají exploitů, zveřejněných chyb v programech, systémech či protokolech.

Kapitola 5

Skenování portů

Co jsou to porty je vysvětleno v kapitole 3.1. Skenování portů je technika využívající toho, že porty mohou být v následujících stavech.

- **open,accepted**: Pokud je port open (otevřený), znamená to, že na něm běží síťová služba a je možno s ním navázat spojení.
- **closed,denied**: closed port značí uzavřený port. Na pokus o připojení k takovému portu je u TCP portu poslán zpět paket s nastavenými příznaky RST a ACK, v případě portu UDP je poslán ICMP paket typu 3 kódu 3 (port unreachable).
- **filtered,blocked**: Při pokusu o kontaktování tohoto portu nebyla zjištěna odpověď (kladná ani záporná)

Pokud chce útočník napadnout počítač v síti, potřebuje o něm získat co nejvíce informací např. jaké služby na počítači běží. Oskenuje tedy porty a po zjištění aktivních, běžících služeb může využít k jejich zneužití exploit, sociálního inženýrství nebo jiné techniky.

5.1 Techniky skenování TCP a UDP portů

5.1.1 TCP

U skenování TCP portů můžeme využít více technik. Většina využívá různých nastavených příznaků v TCP paketu a toho, že je TCP connection-oriented (navazuje a ustanovuje se spojení). Při navazování spojení jde o tzv. three-way-handshake (viz. 3.3). Pokud port odpoví tak jak má (příznaky SYN/ACK), je ve stavu open. Při odpovědi RST je ve stavu closed a při žádné odpovědi ve stavu filtered/blocked.

SYN scan

SYN scan bývalo relativně neodhalitelné skenování, jelikož nikdy nedokončí TCP spojení. Pro zjištění toho, zda je port otevřen, mu stačí poslat pouze paket s příznakem SYN a čekat na odpověď. Pokud přijde SYN/ACK, útočník ví, že port je otevřený. Podle three-way-handshakingu by měl poslat paket s příznakem ACK, ale ten nepošle a kompletní spojení tedy nenaváže. Je větší pravděpodobnost, že nekompletní navázání spojení nebude logováno, avšak většina aplikací kontrolující dění v síti již tento způsob skenování dokáže odhalit.

Connect scan

Connect scan je vlastně klasické kompletní navázání spojení. Pokud je port ve stavu open, proběhne celý tcp handshake. Tato skenovací technika není příliš využívána, protože poskytuje stejné informace jako SYN scan (5.1.1), ale je pomalejší a kompletní spojení bývávají logována.

TCP Null, FIN a Xmas scan

Tyto způsoby skenování využívají znění TCP RFC 793 [5], kde je řečeno, že pokud je port ve stavu closed, na každá příchozí data, která neobsahují příznak RST, má odpovědět paketem s příznakem RST. Pokud je port ve stavu open, má na jakékoliv pakety, kde nejsou nastaveny ani jedny z příznaků SYN, RST, ACK, zareagovat zahozením paketu.

- Null scan posílá pakety bez nastavení jakéhokoliv příznaku.
- FIN scan nastavuje u paketu pouze příznak FIN.
- Xmas nastavuje příznaky FIN, URG a PSH.

Ack scan

Tento způsob skenování má za účel zjistit pouze to, zda je port filtrován nebo ne. Pokud je port nefiltrován, tak open i closed port by na paket s příznakem ACK měl odpovědět paketem s příznakem RST. Pokud je port filtrován, tak neodpoví.

5.1.2 UDP

U UDP protokolu se využívá pouze jediného skenování. Pro oskenování UDP portu se posílá UDP paket pouze s hlavičkou a neobsahující žádná data. Pokud se vrátí jako odpověď ICMP paket typu 3, kódu 3, je port označen jako closed. Pokud se vrátí ICMP paket jiného kódu (1,2,9,10,13), můžeme port označit jako filtered. Při otevřeném portu jako odpověď přijdou nějaká data z dotazovaného UDP portu.

5.1.3 Rizikové porty

Následující tabulka popisuje nejrizikovější služby pro vzdálené zneužití a porty, na kterých většinou běží.

Port	Služba	Popis
21	FTP	FTP server. Pokud je špatně zabezpečen může být terčem útoku pomocí různých exploitů nebo využit ke stažení špatně zabezpečených souborů.
23	Telnet	Protokol pro vzdálený přístup. Je nekódován. Pomocí odposlechu mohou být zjištěna důvěrná data.
25	SMTP	Služba pro odeslání pošty. Při špatném nastavení může být použita ke generování spamu, nebo útok na poštovní server.
80	HTTP	Nejpoužívanější služba internetu. Webový server. Lze napadnou pokud je špatně nastaven a neaktulizován.
110	POP3	Protokol pro stahování pošty. Lze odposlouchávat a zjistit tak hesla k emailovým schránkám.
139	NetBIOS session	Ustanovení spojení pomocí NetBIOSu. Pokud je povolena mohou být přístupné soubory na disku.
161	SNMP	Slouží k získání informací o síťových prvcích. Informace získáné pomocí tohoto protokolu mohou být použita pro další útok.
443	HTTPS	Šifrovaný WWW server. Pokud je špatně nakonfigurován nebo neaktulizován může být napadnut.
445	SMB	Služba pro sdílení souborů a tiskáren. Stejné nebezpečí jak u NetBIOS session
1080	SOCKS	Proxy služba. Pokud je přístupná, může se útočník vydávat za jiný počítač.
1494	Citrix	Služba pro vzdálené ovládání plochy aplikačního serveru.
1723	PPTP	Vzdálený přístup do sítě. Možnost prolomení hesla.
3389	Remote desktop	Vzdálené připojení pomocí grafického terminálu. Možnost prolomení hesla k uživatelskému účtu.

Toto je pouze stručný souhrn portů, které je dobré monitorovat a které jsou útočníkem skenovány jako první. Tento výčet není zdaleka kompletní. Další možné porty ke skenování můžeme převzít z online skenerů, nebo prostudováním dokumentace programu nmap, který v základním nastavení skenuje přes 1600 nejpoužívanějších portů.

Kapitola 6

Implementace

Podle pokynů vedoucího práce jsem navrhl sadu funkcí. Tyto funkce jsou obecné a je možno je použít i v jiných projektech. Pomocí těchto funkcí jsem implementoval aplikaci, která monitoruje vybrané porty počítače, zjišťuje jejich odezvu, statistiky si zaznamenává a generuje graf za zvolené časové období. V této kapitole jsou popsány jednotlivé implementační prostředky a vlastní implementace.

6.1 Implementační prostředky

Tato část se zabývá popisem jednotlivých implementačních prostředků, které byly použity.

6.1.1 Libnet

Libnet je knihovna pro práci s pakety. Má následující vlastnosti:

- Napsána převážně v jazyce C. Je abstrakcí nad rozdílnými architekturami a poskytuje jednotné rozhraní pro práci s pakety.
- Programátor má kompletní kontrolu nad konstrukcí jednotlivých částí paketu.
- Je přenositelná. Umožňuje pracovat se stejným rozhraním na platformách Linux, FreeBSD, Solaris a Windows NT.
- Umožňuje vytvářet pakety pro velké množství protokolů.

Použití Libnet pro vytváření paketů

Při vytváření a posílání paketů pomocí knihovny Libnet se postupuje následujícím způsobem.

1. Inicializace síťového rozhraní
2. Alokace paměti pro paket
3. Konstrukce paketu
4. Výpočet kontrolních součtů
5. Poslání paketu

Inicializace síťového rozhraní

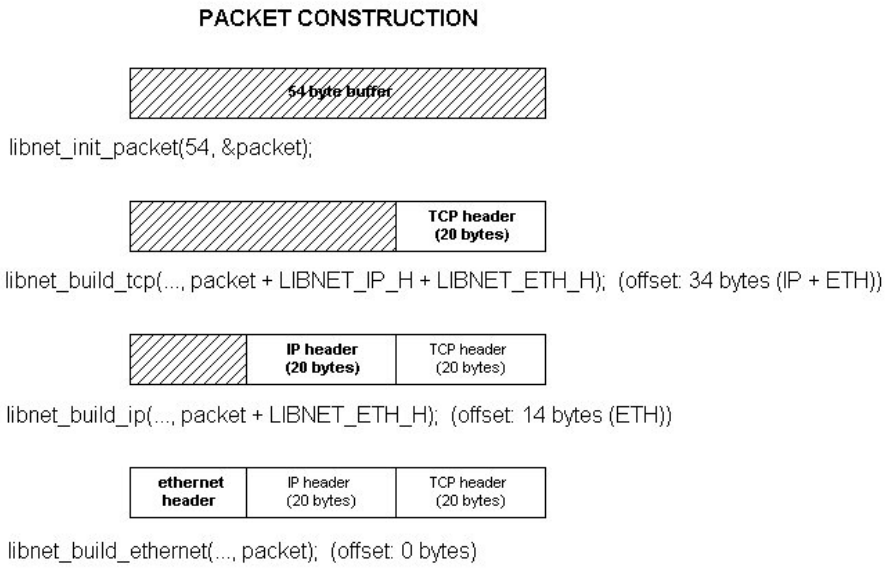
Jako první krok je nutné provést inicializaci síťového rozhraní, kterým chceme pakety posílat. Můžeme použít dva způsoby inicializace. Pokud inicializujeme rozhraní jako *raw socket interface*, máme kontrolu při vytváření paketů po síťovou vrstvu. Hlavička linkové vrstvy bude doplněna kernelem. Při inicializaci rozhraní jako *link-layer interface* máme kontrolu i nad linkovou vrstvou a musíme vytvořit i ethernetovou hlavičku paketu.

Alokace paměti

Při alokaci paměti musíme předem vědět, jak velký paket chceme poslat. Do velikosti je nutné započítat všechny hlavičky protokolů a velikost případných dat. Záleží také, jak je inicializováno síťové rozhraní. Pro TCP paket a 30 bytů dat musíme alokovat při použití **raw interface** 70 bytů (IP header + TCP header + data) a při použití **link-layer interface** 84 bytů (ethernet header + IP header + TCP header + data). Bezpečným řešením může být alokace maximální velikosti paketu s ethernet hlavičkou.

Konstrukce paketu

Konstrukci paketu jednoduše vysvětluje obrázek 6.1. Provede se alokace paměti pro paket. Prvních 14 bytů zabere hlavička ethernetu, následujících 20 bytů hlavička protokolu IP a posledních 20 bytů je určeno pro protokol TCP.



Obrázek 6.1: Vytváření paketů pomocí Libnet

Výpočet kontrolních součtů a poslání paketu

Dalším krokem je výpočet kontrolních součtů a poslání paketu. Pokud se používá raw socket interface, je nutné spočítat kontrolní součty u TCP. Kontrolní součty pro IP jsou vypočítány kernelem. Pro link-layer interface je **nutné** vypočítat i kontrolní součet pro IP. Posledním krokem je poslání paketu do sítě.

6.1.2 Libpcap

Pcap je rozhraní pro zachytávání síťové komunikace. Shrnutí vlastností:

- Pcap je možno používat na systémech Linux a Unix (libpcap) nebo Windows (WinPcap).
- Pomocí Pcap je možné zachytávat síťovou komunikaci procházející zvoleným síťovým rozhraním.
- Je implementován pro použití v jazycích C, C++. Ostatní jazyky mohou používat wrapper (python-libpcap pro python, jpcap pro Javu atp.)

Použití Pcap pro zachytávání síťové komunikace

Použití pcap pro zachycení a analýzu síťové komunikace můžeme popsat v následujících krocích:

1. Inicializace pcap
2. Nastavení filtrů
3. Zachycení a analýza komunikace

Inicializace pcap

Jako první krok je nutné inicializovat pcap a zvolit, na kterém síťovém rozhraní má naslouchat. Rozhraní může být více.

Nastavení filtrů

Pomocí filtrů můžeme omezit zachytávání pouze námi definované komunikace (např. pouze pakety z počítače 192.168.0.1 přicházející na port 23). Tyto filtry je nutné před hlavním odposlechem "zkompilovat" a nastavit pro zvolené rozhraní.

Příklad filtru:

```
ip.addr==192.168.0.1 && tcp.port==23
```

Zachycení a analýza komunikace

Po nastavení požadovaných filtrů následuje vlastní odposlech. Pcap naslouchá po požadovanou dobu, nebo do odchycení zvoleného množství paketů na definovaném rozhraní, a pro každý paket vyhovující filtru je volána programátorem definovaná funkce pro jeho zpracování.

6.1.3 PHP

PHP[1] je skriptovací programovací jazyk, používaný hlavně k tvorbě dynamických webových stránek a aplikací. Je nezávislý na platformě. Obsahuje velké množství knihoven pro práci s grafikou, zpracování textu a umožňuje přístup k většině databázových serverů (např. MySQL, Oracle, PostgreSQL a další).

Knihovna Jpgraph

Pomocí této objektově-orientované knihovny napsané kompletně v PHP můžeme za pomocí PHP skriptů generovat různé typy grafů.

6.1.4 MySQL

MySQL je databázový systém, který používá jazyk SQL. Díky tomu, že je volně šířitelný (je k dispozici pod GPL i pod komerční licencí), multiplatformní, výkonný a má dobrou podporu v PHP, je jedním z nejpoužívanějších současných databázových systémů pro webové aplikace. Od počátku bylo optimalizován na rychlost. Proto některé z pokročilejších funkcí jazyka SQL (pohledy, triggery, vnořené SELECTy) byly přidány až v posledních verzích.

[1] PHP: Hypertext Preprocessor

6.2 Knihovna funkcí

V jazyce C jsem naprogramoval knihovnu funkcí, které se mohou používat pro práci s pakety a jejich analýzu. Popis jednotlivých funkcí je možné najít v souboru `headers.h`, kde jsou jednotlivé funkce definovány.
Funkce jsou následující:

- Funkce pro práci s knihovnou Libnet a pro vytváření paketů. Implementace v `create_packet.c`

  `init_libnet()` – Tato funkce inicializuje zvolené rozhraní a knihovnu libnet. Je vyžadovaná na začátku programu pro inicializaci knihovny libnet na zvoleném rozhraní. Při použití volá funkci `libnet_initz` knihovny `libnet.h`.

  `prepare_packet()` – Pomocí této funkce se vytvoří paket jak je popsáno v 6.1.1. Z knihovny `libnet.h` jsou volány funkce `libnet_build_tcp()` a `libnet_build_ipv4()`.

  `send_packet()` – Pošle paket na zvolené rozhraní. Používá funkci `libnet_write()` pro odeslání paketu a standardní funkci `gettimeofday()` z `sys/time.h` pro zjištění doby odeslání paketu.

- Funkce pro práci s knihovnou libpcap a pro zachytávání a analýzu paketů. Implementace v `capture_packet.c`

  `init_pcap()` – Inicializuje rozhraní a knihovnu pcap. Podobně jako u knihovny Libnet je i u knihovny pcap vyžadováno volání této funkce na začátku, pro inicializaci knihovny pcap na zvoleném rozhraní. Z `pcap.h` volá funkci pro inicializaci `pcap_open_live()`

  `prepare_filter()` – Nastaví na zvoleném rozhraní požadovaný filtr. Z `pcap.h` volá funkce `pcap_compile()` pro zkompilování zvoleného filtru a `pcap_setfilter` pro nastavení filtru na zvolené rozhraní.

  `parse_packet()` – Tato funkce je volána při zpracování paketu, který vyhovuje nastavenému filtru. Používá makra `IP_HL()`, `TH_OFF()` pro zjištění velikosti hlaviček protokolů IP a TCP. Kontroluje nastavené flagy u protokolu TCP (SYN, ACK atp.) pro zjištění, zda je daný port otevřen, filtrován nebo ve stavu closed. Volá také funkci `gettimeofday()` z `sys/time.h` pro zjištění doby přijatého paketu.

- Funkce pro práci s časem. Implementace v `print.c`

  `print_ts()` – Vypíše čas např. poslání nebo příjmu paketu.

  `print_response()` – Vypíše odezvu dané služby. Používá makro `TIMERSUB()` pro odečtení dvou časových hodnot.

- Ostatní funkce. Jejich implementace je v `utils.c`

  – `is_alive()` – Funkce zjišťuje, zda je požadovaný port otevřený, blokovaný, nebo filtrovaný. K tomuto účelu používá funkce `pcap_setnonblock()` pro odblokování handleru, který vrací funkce `init_pcap()`. Pokud je nastaven non–blocking mód, funkce `pcap_dispatch()`, která slouží k zachycení paketu, vrací ihned 0, místo toho, aby handler blokovala a čekala, až paket dorazí. Pokud žádný paket nedorazí, port bude pravděpodobně filtrovaný a je poslán další paket pro potvrzení.

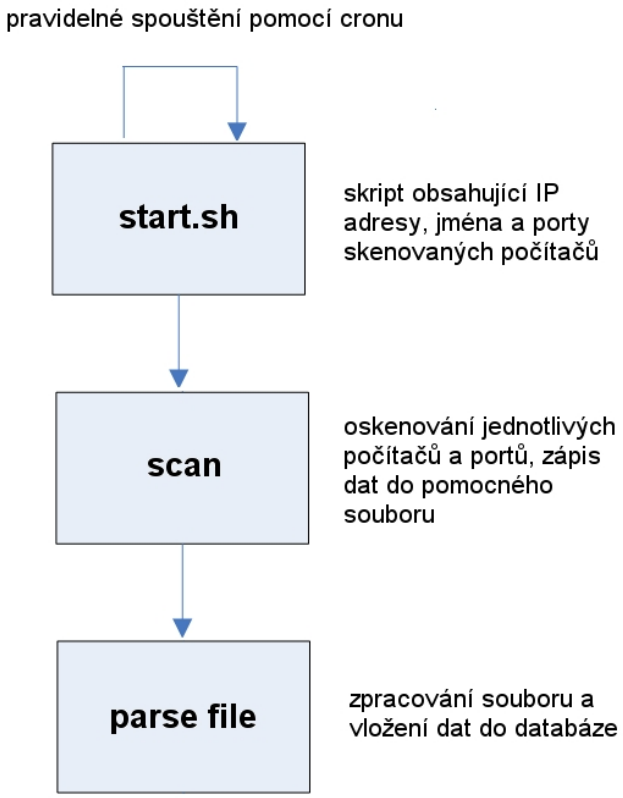
V opačném případě se zanalyzují nastavené flagy v TCP hlavičce paketu a port se prohlásí za otevřený (příznaky SYN+ACK), nebo zavřený (příznak RST). Po zjištění stavu portu funkce `is_alive()` opět nastaví schránku na blokující.

- `find_service()` – Zjistí požadovanou službu, která na portu běží. K tomuto využije kompletního spojení (three-way-handshake) pomocí socketů.
- `write_info()` – zapisuje získané data do souboru

Dále bylo nutné definovat struktury pro IP a TCP hlavičky, aby bylo možné snadněji manipulovat s přijatými daty. Tyto struktury jsou také definovány v `headers.h` a jsou navrhnuty tak, aby mohly být použity i v jiných projektech. Pro práci s IP hlavičkou jsou definovaná makra `IP_HL(ip)` a `IP_V(ip)`, která zjistí velikost hlavičky nebo verzi protokolu. Pro práci s časem je definováno makro `TIMERSUB(a, b, result)`, pomocí kterého lze bezpečně odečítat 2 časové hodnoty definované ve struktuře `timeval`.

6.3 Demonstrace funkčnosti

Pro demonstraci funkčnosti funkcí z navrhnuté knihovny jsem implementoval program, který zjišťuje odezvy portů a generuje grafy zatížení jednotlivých služeb. Pořadí jednotlivých činností je zobrazeno na obrázku 6.2. Grafické zobrazení dat pomocí grafů je prováděno v separátním procesu.



Obrázek 6.2: Posloupnost a jednotlivé kroky programu

6.3.1 Inicializace a spuštění

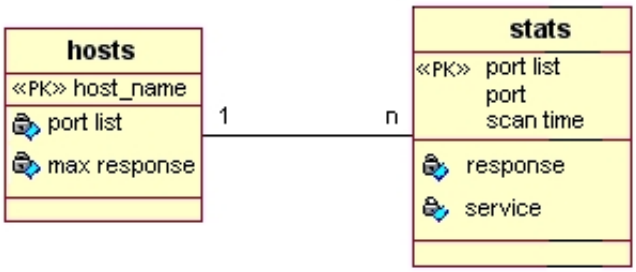
Protože skenování jednoho počítače by bylo nedostatečné a zapisovat více počítačů na příkazovou řádku by bylo neefektivní, je seznam počítačů a jejich portů zapsán ve skriptu `start.sh`, který je pravidelně co minutu spouštěn cronem. Přidávání počítačů nebo upravování jejich skenovaných portů je možné ruční editací skriptu, nebo přes grafické rozhraní viz. A. Program, který je zavolán přes skript, si zjistí z argumentů příkazového řádku IP adresu nebo hostname počítače a porty, které má skenovat. Následuje skenování a zjištění dostupnosti a odezvy daného portu.

6.3.2 Zaznamenání dat

Program zapíše data do pomocného souboru, který je následně zpracován PHP skriptem a data jsou uložena v MySQL databázi.

Datové struktury a formát souborů

Použité datové struktury pro databázi MySQL jsou zobrazeny v následujícím ER diagramu.



Obrázek 6.3: ER-Diagram

Při oskenování jednotlivých počítačů a portů jsou výsledky zapsány do pomocného souboru, ve formátu CSV, který je zpracován PHP skriptem viz 6.2. Data jsou ukládána v posloupnosti port, služba, doba skenování v UNIX formátu, odpověď portu.

Příklad:
Název souboru: kazi.fit.vutbr.cz
Obsah souboru:

```
22,SSH-2.0-OpenSSH_4.2p1 FreeBSD-20060930,1178790912.848808,0.000463
21,FTP,1178790912.921678,0.000480
80,no service,1178790912.922299,0
```

Tato data jsou zpracována PHP skriptem a vložena do databáze MySQL. Z tohoto příkladu vidíme, že na portu 80 neběží žádná služba, což může být z důvodu výpadku dané služby nebo v tomto případě tím, že server danou službu neposkytuje.

Upozornění administrátora

Pokud při posledních 5 měřeních přesáhla odezva portu nastavenou hodnotu, pro daný port odešle se e–mail administrátorovi serveru.

6.3.3 Zobrazování dat

Pro interakci s uživatelem je možné zvolit libovolný grafický webový prohlížeč. Data jsou zobrazena za zvolené časové období, jako spojnicový graf. K vykreslení grafu je použita knihovna jpgraph 6.1.3. Příklady jsou uvedeny v příloze.

6.3.4 Nastavení

Nastavení je možno provádět ruční editací skriptů nebo pomocí webového rozhraní, kde je tato možnost implementována.

6.4 Experimentální výsledky

Testovat jsem zkoušel studentský server **eva.fit.vutbr.cz** a zaměstnanecký server **kazi.fit.vutbr.cz**. Konfigurace serverů jsou následující.

- **kazi.fit.vutbr.cz**

  Hardware

  SuperMicro 6025B-TR+V, MB X7DBE+, 2xIntel Xeon 5160 (Core2 3GHz-4MB), 4 GB RAM, 150 GB HDD + 750 GB RAID-5 3Ware 9550SX

  Software

  FreeBSD 6.2, sendmail, NFS, Apache, Samba, IMAP4

- **eva.fit.vutbr.cz**

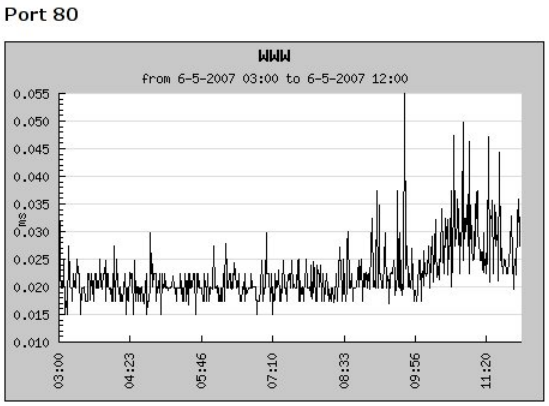
  Hardware

  SuperMicro 6025B-TR+V, MB X7DBE+, 2xIntel Xeon 5160 (Core2 3GHz/4MB), 4 GB RAM, 150 GB HDD + 750 GB RAID-5 3Ware 9550SX

  Software

  FreeBSD 6.2, sendmail, NFS, Apache, Samba, IMAP4

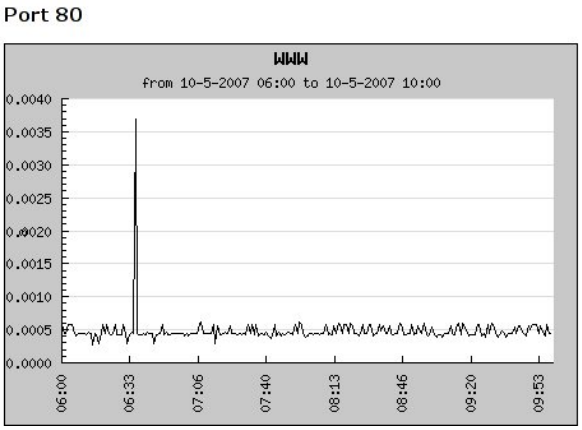
Monitorování portu 80 u serveru **eva.fit.vutbr.cz**



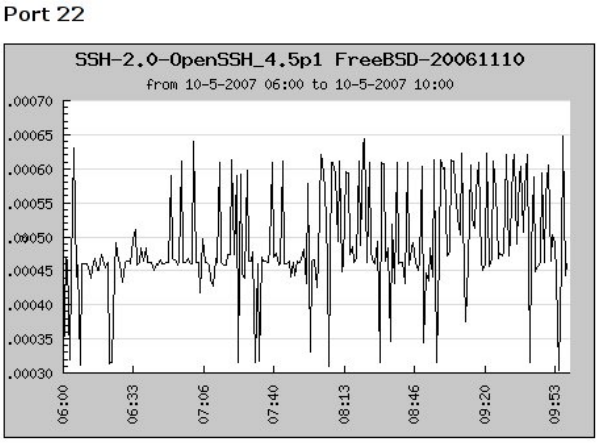
Obrázek 6.4: graf vytíženosti portu 80

V obrázku můžeme vidět zvětšení doby odezvy po 10 hodině. Příčinou může být větší vytížení sítě nebo více požadavků na daný server a službu. Studenti se například vzbudili a jdou si zkontrolovat své výsledky ve wisu.

Při rychlejší lince může graf vypadat např. jako na obrázku 6.5, kde můžeme pozorovat chvíli trvající výkyv služby, zatímco služba ssh 6.6 je tímto výkyvem nedotčena. Z toho můžeme usuzovat, že se nejedná o výkyv v síti, ale o zvětšené množství požadavků na tuto službu atp.



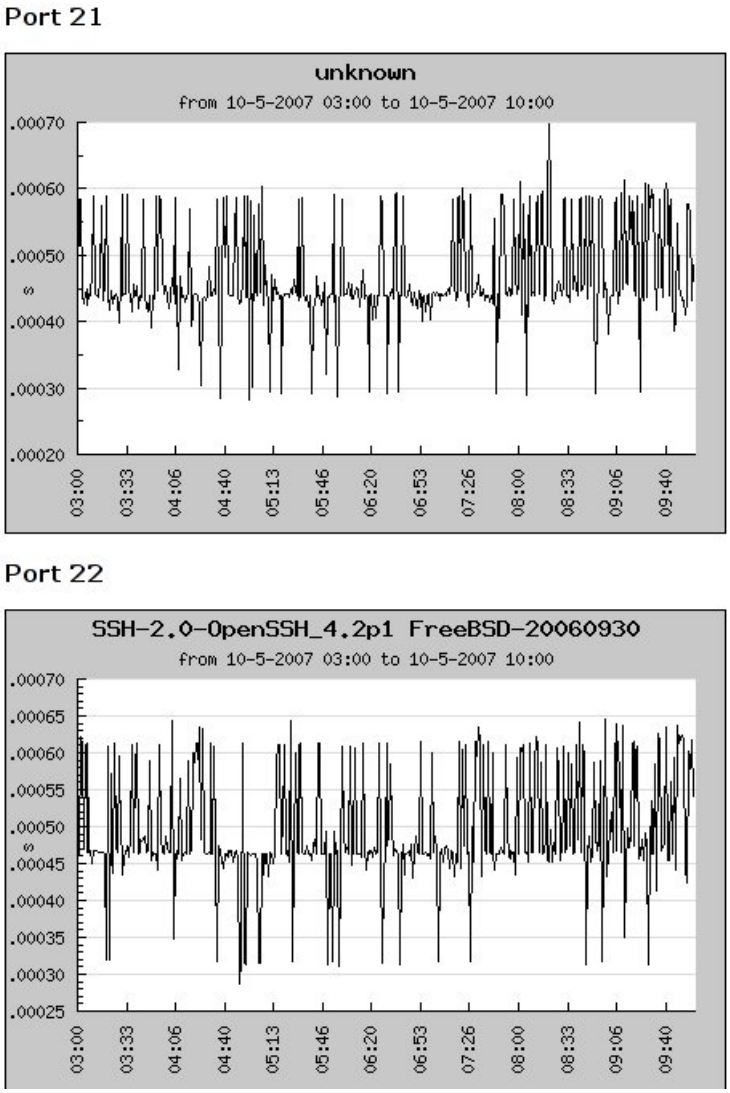
Obrázek 6.5: graf vytíženosti portu 80



Obrázek 6.6: graf vytíženosti portu 22

Monitorování portu 22 a 21 u serveru **kazi.fit.vutbr.cz**
Při monitorování zaměstnaneckého serveru můžeme na obrázku 6.7 vidět, že tyto porty jsou víceméně rovnoměrně zatíženy. Odchylky jsou v rozmezí desetitisícin sekundy, což jsou spíše výkyvy v síti, než změna zatížení dané služby.

Přesto lze z přiloženého grafu podle velikosti odezvy odhadnout, že FTP služba bude využívána méně, než služba SSH. Případně je pro službu FTP vyhrazena větší část procesoru, paměti nebo kapacity linky.



Obrázek 6.7: graf vytíženosti portu 22

Kapitola 7

Závěr

V této práci jsem se seznámil detailně s protokoly používanými v sítích LAN. Další část práce popisuje útoky, které mohou být realizovány v této síti, a jak se jim bránit. Prostudoval jsem také různé techniky skenování portů. Teoretický základ mi poskytla publikace [2].

V implementační části jsem se prostudoval knihovny Libnet a Pcap, které se používají k návrhu síťových aplikací. V dnešní době je pomocí těchto knihoven implementován např. wireshark, tcpdump, dsniff, nmap, snort a mnohé další. Pomocí těchto knihoven a jazyka C jsem navrhl a implementoval sadu funkcí, které mohou být použity v síťové aplikaci.

Implementovaná knihovna může sloužit k monitorování zabezpečení počítačů. Dokáže sledovat otevřené porty, monitoruje dostupnost služeb a hlídáním prodlev odpovědí může předcházet některým útokům, např. DoS.

Při implementaci mi pomohly články [1] a [6]. Problém ukládání dat jsem vyřešil použitím databáze MySQL. Data jsou zobrazována pomocí dynamických stránek implementovaných v PHP. V PHP jsem také navrhl sadu skriptů, které ukládají data ze souborů do databáze. Grafy jsem implementoval pomocí PHP knihovny Jpgraph.

Literatura

[1] Carstens, T.: Programming with pcap. [online], duben 2007, [cit. 2007-04-08].
URL `http://www.tcpdump.org/pcap.htm`

[2] McNab, C.: *Network Security Assessment*. O'Reilly, první vydání, March 2004, ISBN 0-596-00611-X, 396 s.

[3] Patch, R.: An Introduction to Network Programming Using the NetBIOS Interface. *Microsoft Systems Journal*, ročník 7, č. 2, březen 1992: s. 61–75, ISSN 0889-9932.

[4] Postel, J.: Internet Control Message Protocol. RFC 792 (Standard), Září 1981, updated by RFCs 950, 4884.
URL `http://www.ietf.org/rfc/rfc792.txt`

[5] Postel, J.: Transmission Control Protocol. RFC 793 (Standard), Září 1981, updated by RFC 3168.
URL `http://www.ietf.org/rfc/rfc793.txt`

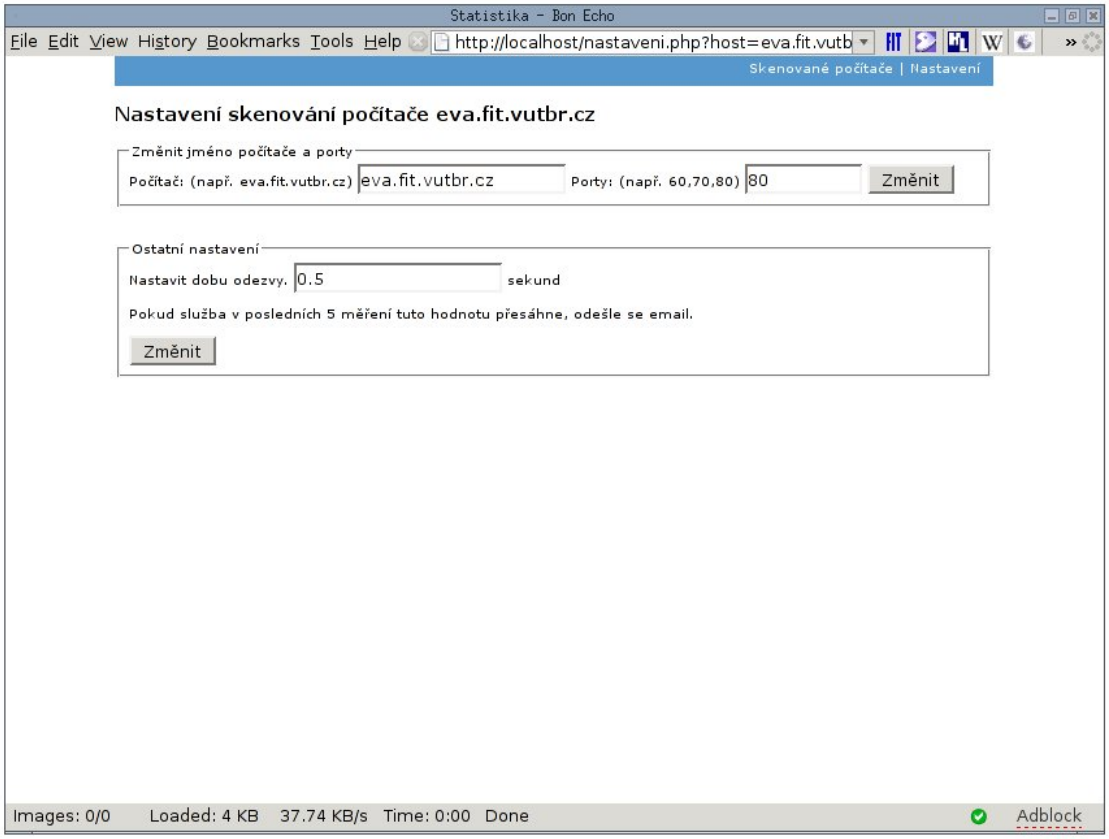
[6] Schiffman, M. D.: Libnet 101 Part 1: The Primer. [online], duben 2007, [rev. 2000-06-19], [cit. 2007-04-08].
URL `http://www.tcpdump.org/pcap.htm`

[7] Wikipedia: OSI model — Wikipedia, The Free Encyclopedia. 2007, [rev. 2007-05-08], [cit. 2007-05-08].
URL `http://en.wikipedia.org/w/index.php?title=OSI_model&oldid=129121916`

Seznam příloh

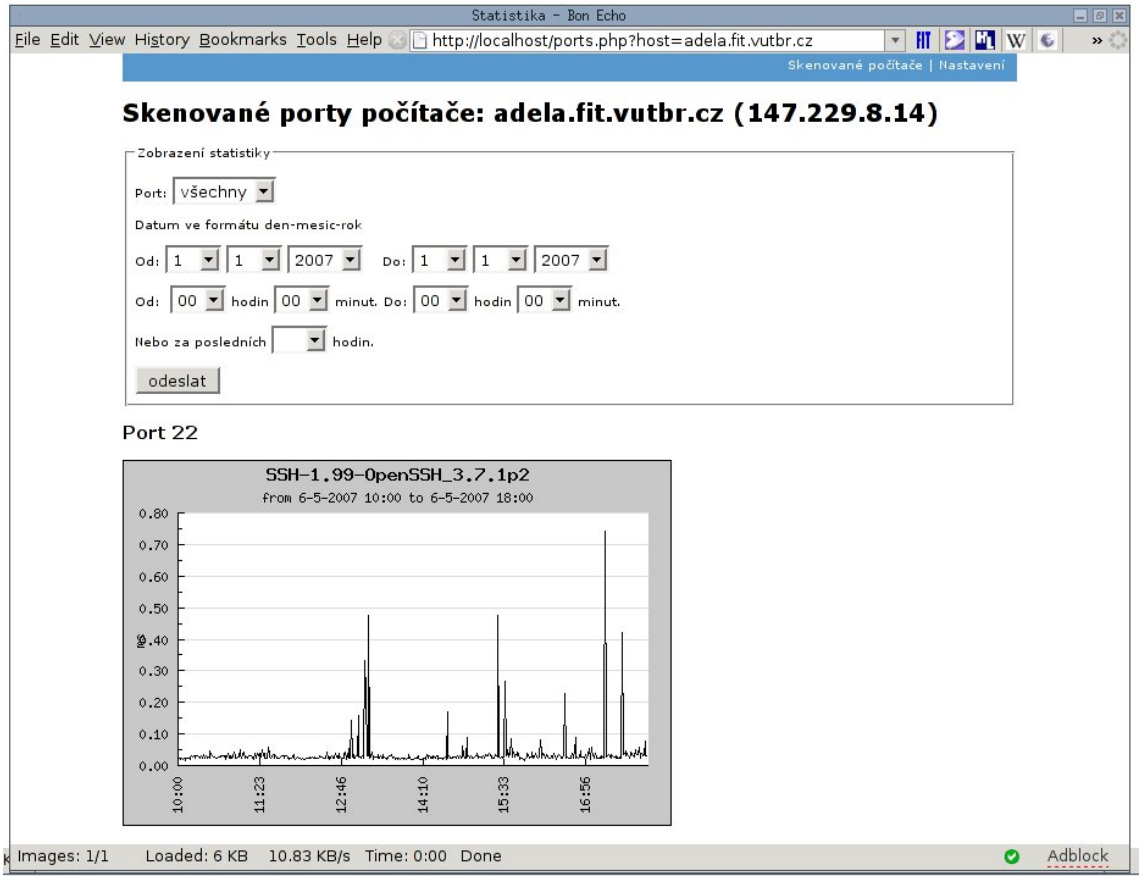
Příloha A

Webové rozhraní



Obrázek A.1: Upřesnění nastavení

Webové rozhraní – upřesnění nastavení počítače
Pomocí tohoto rozhraní můžeme upřesnit porty, které chceme skenovat, upravit jméno skenovaného počítače nebo jeho IP adresu. Také lze pomocí tohoto rozhraní jednoduše nastavit maximalní odezvu portu. Pokud v posledních 5 měření odezva portu přesáhne tuto hodnotu, odešle se e-mail správci sítě.



Obrázek A.2: Zobrazení grafů

Zobrazení statistiky za zvolené období
Pomocí tohoto formuláře si může správce sítě jednoduše zobrazit statistiku skenovaných portů. Může si zvolit vypsání jak všech skenovaných portů, tak pouze zvolených. Výběr za které časové období je neomezený. Pokud databáze za zvolené období obsahuje data (minimálně 2 hodnoty) zobrazí se graf. Jinak se vypíše chybové hlášení, že v požadovaném časovém intervalu není dostatek dat.