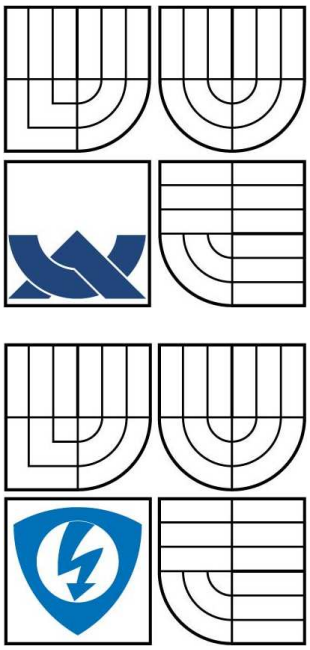


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLÓGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

SOFTWARE PRE DETEKCIU ZRANITEĽNÝCH POČÍTAČOV V SIETI

SOFTWARE FOR COMPUTER SECURITY VULNERABILITY DETECTION

BAKALÁRSKA PRÁCA

BACHELOR'S THESIS

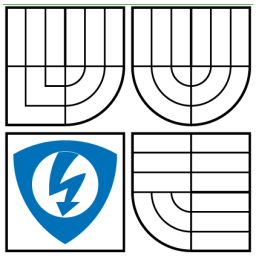
AUTOR PRÁCE — Radko KRKOŠ

AUTHOR

VEDÚCI PRÁCE — Ing. Michal POLÍVKA

SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Radko Krkoš	***ID:***	78626
Ročník:	3	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

Software pro hledání zranitelných počítačů v síti

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte možnosti detekce zranitelných počítačů v síti. Navrhněte a popište metody detekce zranitelných počítačů. Realizujte program vyhledávající zranitelné počítače v lokální síti v předem definované časové periodě, případně na vyžádání obsluhy s využitím postupů navržených v teoretické části práce. Programovací jazyk zvolte C/C++ (IDE volte podle vlastního uvážení).

DOPORUČENÁ LITERATURA:

[1] SCAMBRAY, Joel, MCCLURE, Stuart, KURTZ, George. Hacking bez tajemství. Praha: Computer Press, 2001. 592 s. ISBN 80-7226-549-0.
[2] ENDORF, Carl, SCHULTZ, Eugene, MELLANDER, Jim. Hacking - Detekce a prevence počítačového útoku. Praha: Grada, 2005. 356 s. ISBN 80-247-1035-8.
[3] DOSTÁLEK, Libor, KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. 3. vyd. Praha: Computer Press, 2002. 542 s. ISBN 80-7226-675-6.
[4] SHINDER, Debra Littlejohn. Počítačové sítě. [s.l.]: Softpress, 2003. 752 s. ISBN 8086497550

Termín zadání:	9.2.2009	***Termín odevzdání:***	2.6.2009

Vedoucí práce: Ing. Michal Polívka

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Abstrakt

Práca sa zaoberá bezpečnosťou počítačových systémov a sieťovej infraštruktúry. Opisuje problematiku bezpečnostných dier a zraniteľností a diskutuje možnosti ochrany pred počítačovými útokmi. Popísané sú najčastejšie bezpečnostné diery a chyby v nastaveniach bežných klientských a serverových aplikácií. Uvedené sú jednoduché rady a postupy ako nastaviť prostredie s ohľadom na bezpečnosť a minimalizovať riziko zlyhania či narušenia systému kvôli bezpečnostným chybám a dieram. Práca sa tiež venuje spôsobom hľadania bezpečnostných dier v programoch a zariadeniach zverených správcovi, inventarizácií zariadení, možných zraniteľností a možnostiam prevencie. Popisuje ako vykonať vlastný bezpečnostný audit a ako pracovať s jeho výsledkami. Navrhuje z pohľadu bezpečnosti postupy a metódy pre správu siete, servera, či intranetu počas ich životného cyklu. Taktiež je analyzovaný existujúci dostupný software a sú zhodnotené vlastnosti a možnosti jednotlivých programov s ohľadom na problematiku. Programy sú vyberané tak, aby správcovi zjednodušili prácu v niektorej alebo viacerých oblastiach správy bezpečnosti. Jedná sa predovšetkým o analýzu stavu siete, detekciu chýb a zraniteľností v počítačoch, sieťovej infraštruktúre a vo webových aplikáciách, ale aj o aplikácie na sledovanie zmien v sieti. Práca tiež navrhuje požiadavky na aplikáciu pre bezpečnostný audit a správu zabezpečenia a diskutuje vhodné vlastnosti tejto aplikácie z hľadiska funkcionality a pridanej hodnoty. Vytvorená sada skriptov zjednodušuje prácu správcovi siete tým, že mu umožňuje automatizovať časté a časovo náročné úlohy a poskytuje informácie v kompaktnej a jednoduchej forme, čo robí prácu pohodlnejšou a skracuje reakčný čas na nápravu krízových stavov ako objavenie novej bezpečnostnej diery alebo narušenie bezpečnosti.

Kľúčové slová:

bezpečnosť, počítač, skenovanie, bezpečnostná diera, detekcia, aplikácia, software

Abstract

This thesis concerns about computer system and network infrastructure security. It describes the topic of security holes and vulnerabilities and discusses possibilities for computer attack defense. Common security holes and client and server systems configuration errors are described. There are stated simple rules and advices for configuring network environment according to security and minimalization of failure or violation risk due to security holes and vulnerabilities. Thesis adresses approaches of security hole detection in programs and devices entrusted to administrator, device and security holes inventory control and prevention possibilities. Thesis describes how to execute self security audit and how to process its results. It suggests techniques and methods to administer network, server, or intranet during its lifespan according to security. It also analyses existing available software and evaluates its features and resources with regard to security topic. Software is chosen to simplify work for administrator in some or more parts of security management like network condition analysis, error and vulnerability detection in computer systems, network infrastructure, web applications or applications for network alternation detection. Thesis recommends requirements for security audit application and discusses eligible features in regard of functionality and added value. Created set of scripts simplifies administrator's work by automating common and time consuming tasks and delivers information in compact and simple form, what makes the work more comfortable and shortens the reaction time to crises such as discovery of new security hole or security breach.

Keywords:

security, computer, scanning, vulnerability, detection, application, software

Citácia práce

KRKOŠ, R. Software pre detekciu zranitelných počítačov v sieti. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 54 s. Vedoucí bakalářské práce Ing. Michal Polívka.

Prehlásenie

Prehlasujem, že svoju bakalársku prácu na tému „Software pre detekciu zraniteľných počítačov v sieti“ som vypracoval samostatne pod vedením vedúceho bakalárskej práce s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname použitej literatúry na konci práce.

Ako autor uvedeného bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Zb. platného v Českej republike, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia § 152 trestného zákona č. 140/1961 Zb. platného v Českej republike.

V Brne dňa 01.06.2009 .....

podpis autora

Obsah

Zoznam obrázkov

Úvod

Práca sa zaoberá problematikou bezpečnosti počítačových systémov v počítačových sieťach. Cieľom je naštudovať túto problematiku a navrhnúť spôsob ochrany pred počítačovými útokmi vypracovaním metodiky zisťovania bezpečnostných dier a slabých zneužiteľných miest v počítačových systémoch s prihliadnutím na bezpečnosť systémov v počítačovej sieti. Konečným produktom by mal byť program na detekciu a inventarizáciu bezpečnostných dier a slabín pre uľahčenie práce správcovi lokálnej počítačovej siete, intranetu či serveru.

V prvej časti sú rozpracované možnosti detekcie zraniteľných miest a bezpečnostných dier pomocou scanovania stavu IP portov počítačových systémov a analýzou odozvy. Uvedených je niekoľko techník, ktoré sú popísané v rozsahu umožňujúcom o nich získať základné znalosti potrebné pre prácu s programom.

V druhej časti sú načrtnuté najčastejšie bezpečnostné diery v počítačových systémoch a to aj z hľadiska serverov aj pracovných staníc užívateľov. Zaoberá sa systémami, ktoré sú prítomné vo väčšine počítačových sietí, najmä vo firemných, ale aj vo väčších domácich (SOHO). V prípade pracovných staníc sa zameriava na najčastejšie využívané programy, ktoré sú používané bežnými užívateľmi. Uvádzaný je aj spôsob detekcie a základné rady pre ochranu pred bezpečnostnými dierami.

V tretej časti je uvedený opis základných vlastností niektorých bežne dostupných programov, ktoré administrátor siete, intranetu alebo serveru môže použiť pre zjednodušenie práce. Ide o programy a aplikácie umožňujúce analýzu stavu siete, detekciu zraniteľností v počítačoch a sieťovej infraštruktúre, detekciu chýb a zraniteľností vo webových aplikáciách, ale aj na sledovanie zmien v sieti. Opis je zameraný na vymenovanie vlastností a možností jednotlivých programov a ich zhodnotenie. Takisto je uvedené pod akou licenciou je daný program vydaný za akých podmienok je ho možné používať.

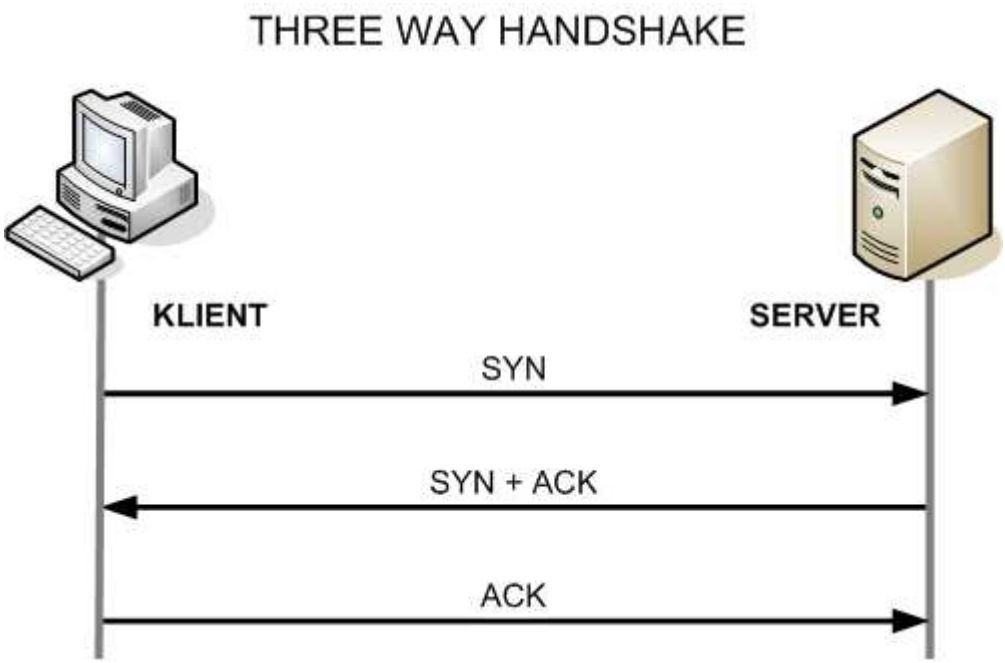
Vo štvrtej časti sa práca zaoberá tvorbou samotného programu na detekciu zraniteľných počítačov. Je tu uvedený základný zoznam požiadaviek, ktoré by mal program spĺňať či už kvôli jednoduchosti a komfortu ovládania alebo užitočnosti vo svojom cieľovom použití.

1. Scanovacie techniky

Scanovanie ako metóda na hľadanie využiteľných komunikačných kanálov sa používa už roky. Cieľom je vyskúšať toľko počítačov, koľko sa len dá a zaznamenať si tie užitočné. Počas rokov sa vyvinulo mnoho techník na zisťovanie používaných protokolov a služieb a stavu jednotlivých portov. Každá má svoje výhody a nevýhody.

1.1. TCP connect() scan

Najjednoduchšia a najzákladnejšia forma scanovania. Používa bežné SOCKET connect() volanie operačného systému na vytvorenie spojenia s každým žiadaným portom na cieľovom počítači. Ak na danom porte cieľ počúva, úspešne prebehne trojcestný handshake, connect() ohlási úspešné spojenie a scanner spojenie ihneď zatvorí. Inak je port nedostupný a connect() ohlási chybu. Priebeh trojcestného handshake-u je zobrazený na Obr. 1.1. Veľká výhoda tejto metódy je že na jej použitie nie sú potrebné administrátorské práva. Akýkoľvek používateľ môže zavolať funkciu connect().



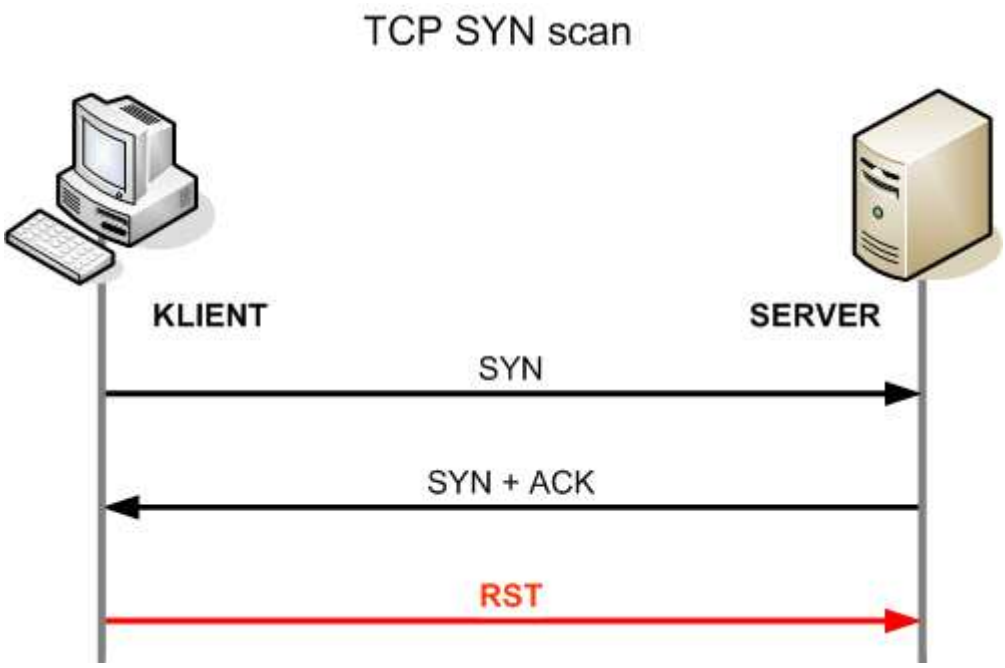
Obr. 1.1: Priebeh trojcestného handshake-u použitého pri TCP connect() scanovaní

Ďalšou výhodou je rýchlosť. Oddelené volanie connect() pre každý port za sebou by trvalo veľmi dlho, je ale možné použiť paralelné SOCKET-y. Použitie neblokujúcich I/O

dovoluje nastaviť krátky časový limit a skenovať všetky porty naraz. Obvykle je to najrýchlejší spôsob, avšak je ľahko detekovateľný a filtrovateľný. Cieľ zaznamená množstvo pripojení a chybových hlásení služieb, ktoré nadviazali spojenie a to bolo následne hneď ukončené. Niektoré archaické a zle navrhnuté služby môžu zhavarovať keď je spojenie ihneď po nadviazaní ukončené bez poslania dát.

1.2. TCP SYN scan

Pri tejto metóde si namiesto použitia sieťových funkcií operačného systému scanner vytvorí IP pakety sám a prípadné odpovede si sám odchytáva a spracováva. Táto metóda sa taktiež nazýva „half-open scan“, pretože pri nej nie je vytvorené celé spojenie. Pošle sa SYN paket ako pri naozajstnom spojení a čaká sa na odpoveď. V prípade SYN/ACK ide o počúvajúci port, RST znamená uzavretý port. Ak príde SYN/ACK, ihneď sa naspäť odosiela RST na zrušenie spojenia pred dokončením trojcestného handshake-u. Priebeh TCP SYN scanovania je naznačený na Obr. 1.2.



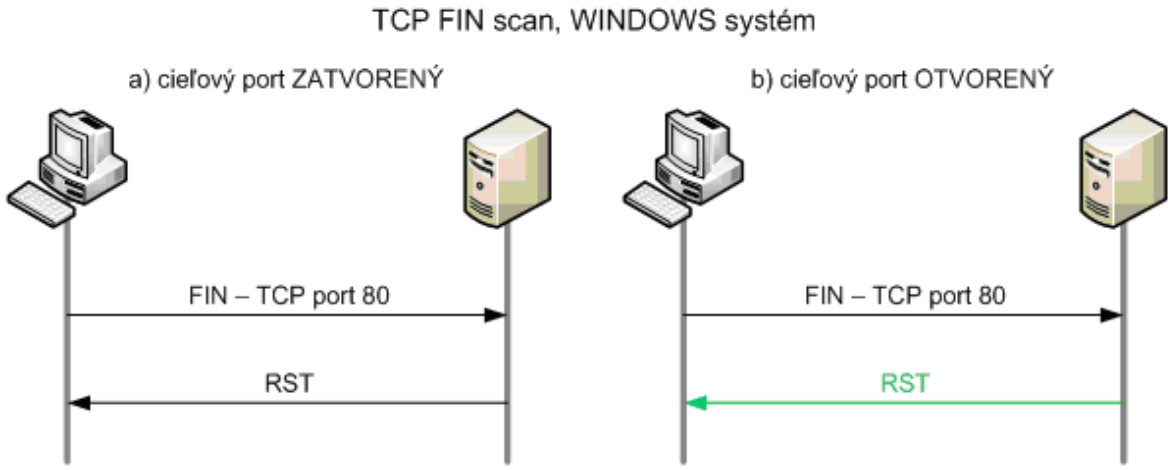
Obr. 1.2: Priebeh TCP SYN scan-u

Najväčšia výhoda je že tento spôsob býva zachytený menším počtom cieľov, pretože vlastná služba vôbec nedostane správu o realizovanom spojení. Avšak niektoré implementácie TCP/IP sú citlivé na zrušenie spojenia počas handshake-u a takýto postup

spôsobí narušenie ich funkcie. Taktiež na vytváranie vlastných IP paketov sú potrebné administrátorské práva. Pri takomto vytváraní IP paketov je však možná úplná kontrola odosielaného obsahu a timeout-ov, čo umožňuje získať detailnejšie informácie. Všeobecne je táto metóda kvôli svojim výhodám najpoužívanejšia.

1.3. TCP FIN scan

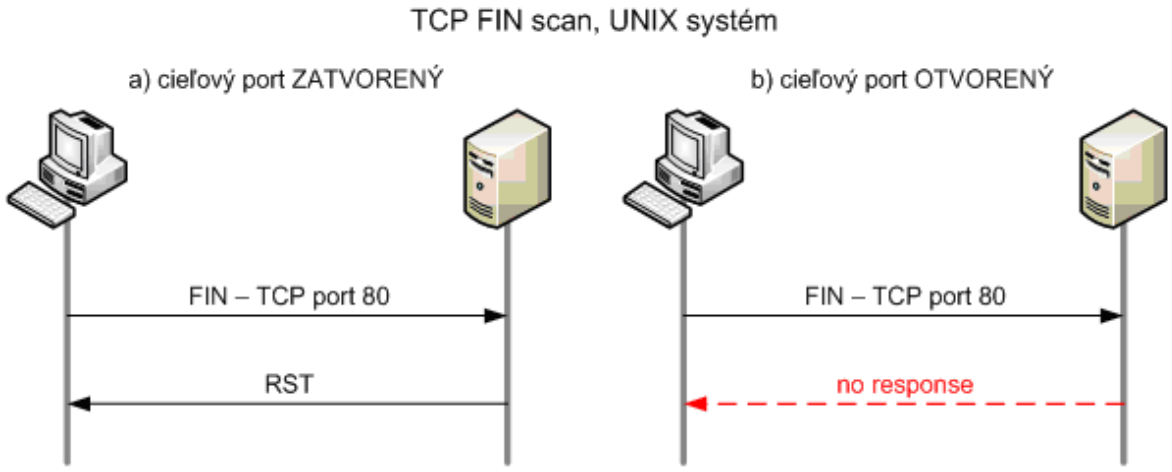
Niekedy ani SYN scan nie je dostatočne nedetekovateľný. Niektoré firewally a packetové filtre sledujú SYN pakety na zakázané porty a dokonca existujú programy ako synlogger, ktoré slúžia na detekciu SYN scanovania. FIN pakety ale môžu prejsť týmito sytémami nedetekované. Táto technika bola prvý krát popísaná Urielom Maimonom v magazíne Phrack 49, článok 15 (preto sa niekedy zvykne nazývať aj Maimon scan). Podstata je v tom, že zatvorené porty zvyknú odpovedať na FIN pakety vhodným RST. Otvorené porty zas FIN pakety ignorujú. Prakticky je toto žiadúce chovanie systému TCP. Avšak niektoré systémy, najmä od firmy Microsoft, toto správanie nedodržiavajú. Odpovedajú RST paketom bez ohľadu na stav portu a sú tak odolné voči tejto technike. Na väčšine ostatných systémov ale TCP FIN scan funguje bez problémov.



Obr. 1.3: Priebeh TCP FIN scanovania systému MS Windows; a) cieľový port zatvorený; b) cieľový port otvorený – nesprávna odpoveď RST paketom;

Prakticky je potom táto technika dobre použiteľná na rozlíšenie UNIX a MS systémov. Keď je o nejakom porte známe, že je otvorený, napríklad z predchádzajúceho scanovania, a na zaslaný FIN paket odpovie RST paketom, tak ide o MS systém (Obr. 1.3).

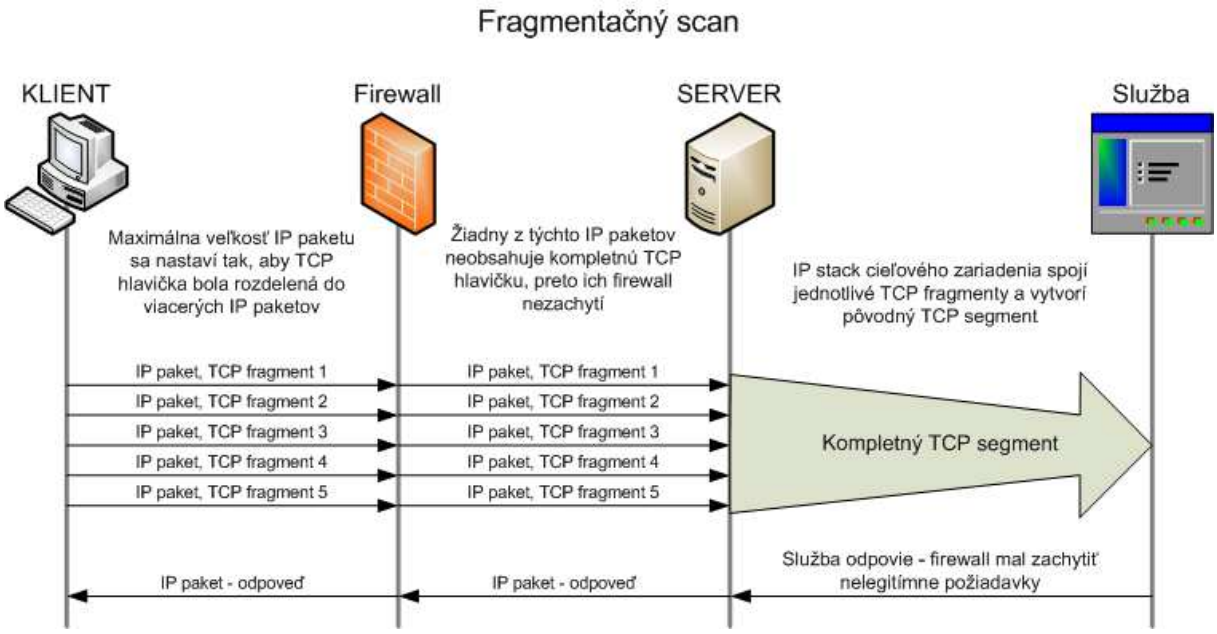
Ak odpoveď nedorazí, jedná sa o systém UNIX (Obr. 1.4). Keďže paket môže byť odoslaný, ale k cieľu kvôli nespoľahlivosti IP protokolu doraziť nemusí, tak scanovanie treba niekoľkokrát opakovať aby sa vylúčili nesprávne výsledky.



Obr. 1.4: Priebeh TCP FIN scanovania systému UNIX; a) cieľový port zatvorený; b) cieľový port otvorený – správna reakcia;

1.4. Fragmentačný scan

Táto technika v skutočnosti nie je novým druhom scanovania, ale len modifikáciou TCP SYN a TCP FIN scanovania. Namiesto vyslania jedného scanovacieho paketu je tento rozdelený na niekoľko menších IP paketov. TCP hlavička sa rozdelí do niekoľkých paketov, čo výrazne zťaží prácu paketovým filtrom a podobným systémom. V cieľovom systéme IP stack jednotlivé TCP fragmenty spojí a transportná vrstva dostane takú istú informáciu ako keby bol prijatý jeden IP paket s kompletným TCP segmentom. Princíp fragmentačného scanovania je naznačený na Obr. 1.5. Niektoré programy, či dokonca aj nekvalitne navrhnuté implementácie IP stacku majú ale problémy spracovať TCP segmenty s nekompletnou TCP hlavičkou. Takisto tieto pakety neprejdú cez detekčné systémy, ktoré spracovávajú IP fragmenty ako celok. Avšak degradáciu výkonu, ktorú takéto spracovanie spôsobuje si nemôže dovoliť mnoho systémov. Táto metóda sa kvôli svojej náročnosti a obvykle zanedbateľnému zlepšeniu dosiahnutých výsledkov používa málo.



Obr. 1.5: Princíp fragmentačného scanovania

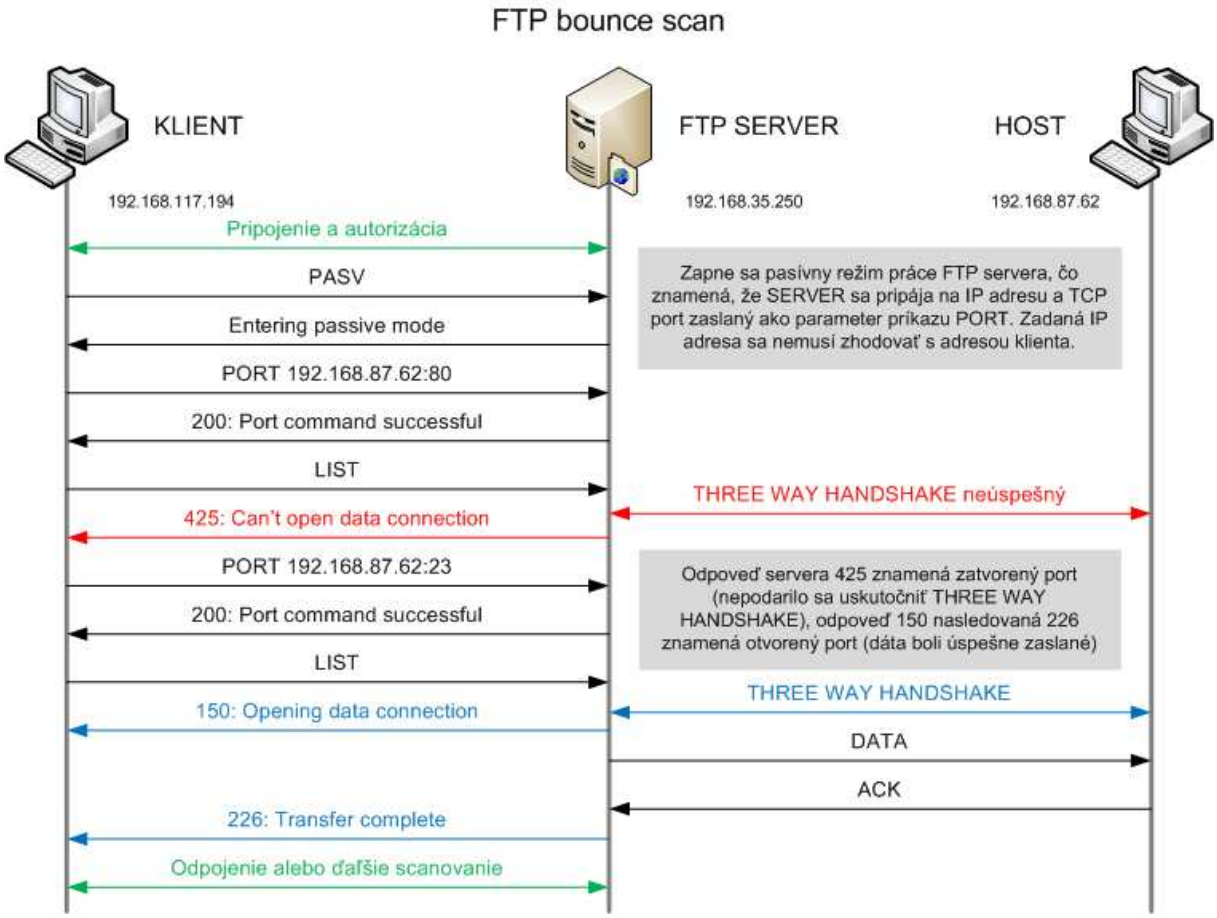
1.5. TCP reverse ident scan

Protokol ident [1] umožňuje zverejnenie užívateľského mena vlastníka akéhokoľvek procesu s aktívnym TCP spojením dokonca aj keď tento proces spojenie sám neinicioval. Je teda možné napríklad pripojiť sa na HTTP server a následne použiť ident na zistenie, či HTTP démon beží s administrátorskými oprávneniami. Je ale nutné kompletné TCP spojenie.

1.6. FTP bounce scan

Zaujímavá vlastnosť protokolu FTP [2] je podpora proxy pripojení. V praxi to znamená, že je možné pripojiť sa na FTP server a potom si vyžiadať zaslanie súboru kamkoľvek. Predpokladá sa použitie FTP klienta ale podľa špecifikácie je možné zaslať dáta aj inému serveru. Samozrejme toto fugovalo v dobách keď bolo RFC len písané (1985), napríklad na posielanie takmer nevystopovateľných e-mailov a diskusných príspevkov. Aj dnes je ale táto technika použiteľná na scanovanie TCP portov z proxy FTP servera. Je možné sa napojiť na FTP server za firewallom a z vnútra siete scanovať porty, ktoré sú pri prístupe z vonka väčšinou blokované. Navyše ak je na serveri aspoň jeden adresár umožňujúci aj čítanie aj zápis, je možné posielať dáta na porty detekované ako otvorené.

Príklad takéhoto útoku je, že sa použije FTP príkaz PORT na deklaráciu, že klientský dátový proces počúva na cieľovom počítači na scanovanom porte. Potom sa použije napríklad príkaz LIST na výpis obsahu adresára. Ak je daný port na cieľovom počítači otvorený, prenos bude úspešný a FTP server oznámi stav 150 (File status okay, about to open data connection -otvorené dátové spojenie) a následne 226 (Closing data connection, requested file action successful - súborová operácia úspešná). V opačnom prípade server oznámi stav 425 (Can't open data connection - dátové spojenie nemohlo byť otvorené). Tak sa pošle nový PORT príkaz s ďalším portom na scanovanie. Veľkou výhodou tejto metódy je že je ťažko stopovateľná a môže obchádzať firewally. Najväčšou nevýhodou je zase veľmi nízka rýchlosť (timeouty FTP servera) a to, že niektoré nové FTP servery neumožňujú proxy pripojenia. Princíp FTP bounce scanovania je zobrazený na Obr. 1.6.



Obr. 1.6: Princíp FTP bounce scanovania

1.7. Idle scan

Táto špeciálna technika sa používa ako účinný spôsob na zakrytie skutočného scanovateľa. Pri takomto scanovaní je použitý iný počítač v sieti, ktorý mmusí byť zaspnutý a musí mať aspoň jeden otvorený TCP port. Táto technika využíva znalosti o fungovaní TCP potvrdzovacieho okna a sekvenčných čísiel paketov, ich správanie v bežných a špeciálnych prípadoch a rozdiely implementácie na jednotlivých operačných systémoch.

Scanovanie touto technikou prebieha podobne ako bežné scanovanie, avšak útočník namiesto svojej IP adresy do zasielaných paketov vloží podvrhnutú IP adresu „zombie“ zariadenia. Scanovaný cieľ odpovedá na túto IP adresu a preto ďalej komunikuje s „zombie“ počítačom. Tento však žiadnu odpoveď neočakáva (keďže spojenie v ktorom scanovaný cieľ pokračuje neinicioval) a takisto nie je schopný zaslať tieto pakety útočníkovi. Je preto nutné doručenie alebo nedoručenie paketov „zombie“ zariadeniu detekovať iným spôsobom. Najvhodnejšia metóda na toto je overovanie zmien v sekvenčných číslach zasielaných paketov.

Scanovaný cieľ si teda vymieňa pakety iba s týmto tzv. „zombie“ počítačom, z ktorého je scanovacím počítačom vďaka sekvenčným číslam IP paketov zisťované, či bol nejaký paket „zombie“ počítačom prijatý. V prípadných logoch scanovaného cieľa sa tak objaví len komunikácia s „zombie“ počítačom a vyzerá to tak, akoby scanovanie viedol on sám. Bez dosť podrobne nastaveného logovania komunikácie na aj scanovanom aj „zombie“ zariadení a bez značnej spolupráce administrátorov týchto dvoch systémov nie je možné odhaliť naozajstného vykonávateľa scanu. Táto metóda je používaná najmä v prípade útoku na infraštruktúru, je ju ale možné použiť v prípade analýzy siete a to na mapovanie práv „zombie“ zariadenia voči scanovanému cieľu, čo umožňuje odhaliť skryté a nepríjemné bezpečnostné diery. Tieto sú vo väčšine prípadov ľahko napraviteľné, ale administrátor siete o nich musí vedieť skôr ako útočník.

1.8. UDP scanovacie techniky

Tieto metódy sa odlišujú v použitom protokole, namiesto TCP používajú UDP. Zatiaľ čo tento protokol je jednoduchší, jeho scanovanie je značne obtiažnejšie. Keďže ide o nespojovo orientovaný a nespoľahlivý protokol, otvorené porty nemusia zaslať potvrdenie spojenia a dokonca ani zatvorené porty nemusia poslať chybovú správu. Takisto UDP pakety nemusia doraziť na miesto určenia a vysielajúca služba sa o tom nedozvie.

UDP scanovanie nie je také populárne a obľúbené ako TCP, avšak v niektorých prípadoch má veľké výhody. Dobrým príkladom je diera v rcpbind v Solarise. Služba rcpbind používa náhodný nedokumentovaný UDP port nad 32770. Takže aj keď je TCP port 111 blokovaný firewallom, je možné prehladať horných vyše 30000 portov, na čo výborne poslúži niektorá z UDP scanovacích techník.

1.8.1. UDP ICMP port unreachable scan

Našťastie neodpovedanie pri pokuse o pripojenie na zatvorený UDP port nie je pravidlom a väčšina systémov pošle chybu ICMP port unreachable, ak je daný port zatvorený. Takto je možné vylučovacou metódou zistiť, ktoré porty sú otvorené. Samozrejme ani UDP ani ICMP správy nemusia doraziť. Takže ak neprijde odpoveď je potrebné spojenie opakovať ešte niekoľko krát aby bolo zabránené nesprávnej detekcii. Táto metóda je pomalá kvôli limitovaniu zaslaných ICMP chybových hlásení (RFC 1812 sekcia 4.3.2.8) [3]. Ak je zasielanie ICMP chybových správ úplne zakázané, tak sa všetky porty budú javiť ako otvorené. Ďalšia nevýhoda je, že na čítanie ICMP chybových správ sú potrebné administrátorské práva.

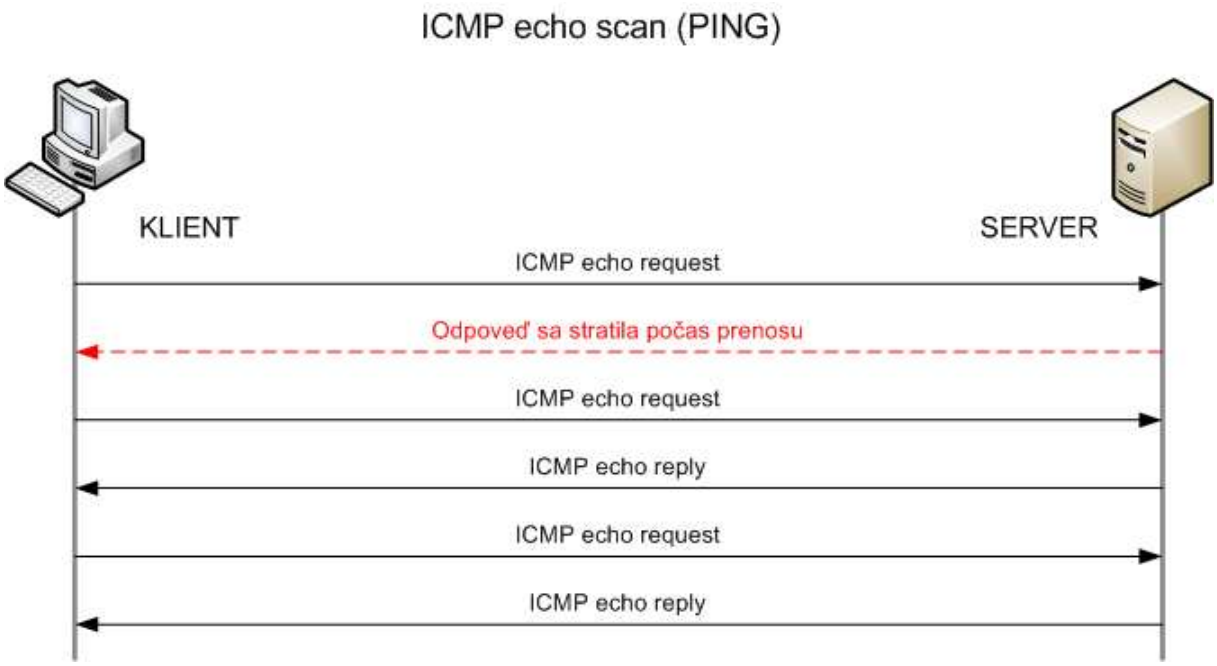
1.8.2. UDP recvfrom() a write() scan

Aj keď na prácu s ICMP socketmi sú potrebné administrátorské práva, mnoho UNIX systémov umožňuje toto obísť a zisťovať stav ICMP socketov nepriamo. Konkrétne je použiteľné to, že druhé volanie funkcie write() pre prístup k zatvorenému portu skončí okamžite chybou. Takisto volanie recvfrom() pri neblokujúcich UDP operáciách vráti chybu 13 (skús znova) ak ICMP chyba nebola prijatá a chybu 111 (spojenie odmietnuté) ak už bola. Toto je obdoba predchádzajúcej metódy a je použiteľná aj pre užívateľov bez administrátorských práv.

1.9. ICMP echo scan

Táto metóda vlastne nie je scan, keďže ICMP nepracuje s portami. Umožňuje ale rýchlo získať zoznam fungujúcich počítačov v sieti. Tento postup používa aj program „ping“. Nevýhodou je, že niektoré systémy na „ICMP echo request“ pakety neodpovedajú. Kvôli tomu, že IP protokol je nespoľahlivý, je v prípade neprijatia odpovede nutné požiadavky opakovať, aby sa vylúčila strata odpovede počas prenosu. Výhodou je, že tento

postup sa dá využiť na monitorovanie aktivity zariadení. Ak cieľ v minulosti odpovedal na ICMP echo request, tak je možné rýchlo a ľahko určiť, či je práve spustený. Priebeh ICMP echo scanovania je zobrazený na Obr. 1.7.



Obr. 1.7: Priebeh ICMP echo scanovania

1.10. Version detection – heuristická analýza služieb

Uvedené scanovacie techniky zisťujú len zoznam otvorených portov. Konkrétnu službu je možné odhadnúť napríklad z tabuliek dobre známych portov, keďže bežné služby majú svoje porty na ktorých bežia. Tento postup však nezaručuje správny výsledok, pretože teoreticky môže ktorákoľvek služba bežať na ľubovolnom porte. Pre zaistenie správnosti je potrebné podľa prijatých, prípadne aj odoslaných dát zistiť aká služba na porte beží. Takýto postup je nesmierne náročný, pretože je vlastne potrebné vytvoriť univerzálneho klienta, ktorý bude navyše vybavený databázou kde bude podľa realizovanej komunikácie hľadať o akú konkrétnu službu sa jedná.

2. Bežné bezpečnostné diery a obrana proti nim

2.1. Zero Day Vulnerability

O zero day vulnerability hovoríme vtedy, keď bola objavená chyba v kóde programu a exploit na zneužitie tejto chyby existuje skôr ako oprava alebo patch. Keď je raz fungujúci exploit vypustený, užívatelia dotknutých systémov budú ohrození kým sa objaví záplata, alebo užívateľ sám nepodnikne preventívne kroky. Bezpečnostná diera zero day vulnerability môže byť nájdená a zneužitá v akomkoľvek operačnom systéme alebo aplikácií. Aj keď počet týchto bezpečnostných dier z roka na rok klesá, stále sa zvyknú objaviť a je im nutné venovať pozornosť. Zneužitie zero day vulnerability je veľmi obávaným pre mnohých administrátorov.

Metodika zníženia dopadu zero day útoku:

- znemožniť všetkú nepovolenú prevádzku cez firewally a iné zariadenia chrániace vnútornú sieť
- oddeliť vnútorné servery od serverov poskytujúcich služby pre verejnosť
- vypnúť nepotrebné služby na serveroch a na užívateľských staniciach odstrániť programy, ktoré nie sú potrebné k vykonávaniu práce
- nastaviť užívateľom minimálne potrebné oprávnenia a prístup potrebné pre ich prácu
- zabrániť alebo aspoň obmedziť používanie aktívneho kódu (ActiveX, JavaScript, VBScript) v prehliadačoch
- zaučiť užívateľov o neotváraní neznámych a neoverených súborových príloh
- vypnúť možnosť otvárať HTTP linky priamo z e-mailu a stahovanie obrázkov z webu e-mailami
- zabezpečiť si spôsob získavania informácií o nových zero day bezpečnostných dier
- zabezpečiť nasadenie opráv a zlepšení ihneď ako sa stanú prístupnými
- v prípade používania Microsoft Active Directory je potrebné čo najlepšie využívať skupinovú politiku na správu prístupových práv
- nespoliehať sa na antivírusové programy, pretože zero day útoky sú často nedetekovateľné kým nie sú zaradené do databázy a heuristická kontrola ich nenájde
- použiť dodatočné systémy ochrany proti pretečeniu zásobníka kde to je len možné
- riadiť sa radami výrobcu programu na zamedzenie alebo zmiernenie dopadu zero day útoku kým nie je uvoľnená záplata

2.2. Bezpečnostné diery v serverových sytémoch

2.2.1. Webové aplikácie

Webové aplikácie ako systémy správy obsahu (CMS), wiki, portály, bulletin board-y alebo diskusné fóra sú používané aj malými aj veľkými organizáciami. Množstvo organizácií taktiež vyvýja a používa vlastné webové aplikácie. Každý týždeň sú objavené stovky bezpečnostných dier, či už v komerčne dostupných alebo open source aplikáciách, a sú aktívne zneužívané. Vlastné webové riešenia sú taktiež napádané a zneužívané, aj keď diery v týchto aplikáciách nie sú verejne oznamované a nie sú udržované ich zoznamy vo verejných bezpečnostných databázach. Počet útokov najmä na veľké web-hostovacie farmy sa šplhá až k miliónu denne. Všetky typy webových aplikácií a framework-ov (PHP, .NET, J2EE, Ruby on Rails, ColdFusion...) sú ohrozené bezpečnostnými chybami od nedostatočného overovania až po logické chyby.

PHP Remote File Include – PHP je najrozšírenejší jazyk pre webové aplikácie a podporná štruktúra používaný dnes. Predvolene PHP umožnuje funkciám zabezpečujúcim súborové operácie prístup k ľubovoľným zdrojom na internete pomocou vlastnosti „allow_url_open“. Ak PHP skript umožňuje užívateľovi aby svojim vstupom ovplyvnil názvy súborov, je možné spôsobiť vloženie vzdialeného súboru.

PHP Remote File Include útoky umožňujú napríklad:

- vykonanie vzdialeného kódu
- inštaláciu rootkit-ov
- na OS Windows je možné celkové narušenie systému použitím PHP funkcií zaobaľujúcich SMB súborové operácie

SQL injektáž – injektáž, najmä SQL injektáž, je vo webových aplikáciách bežne využívaným spôsobom útoku. SQL Injektáž je umožnená miešaním užívateľsky zadávaných údajov s dynamicky vytváranými dopytmi alebo zle navrhnutými procedúrami.

SQL injektáž umožňuje napríklad:

- vytvoriť, prečítať, upraviť alebo zmazať akékoľvek dáta prístupné aplikácii
- v najhoršom prípade úplne narušiť databázový systém a systémy v jeho okolí

Cross-site Scripting (XSS) – najdeštruktívnejšia a najjednoduchšie objaviteľná bezpečnostná diera. XSS umožňuje útočníkovi narušiť štruktúru webových stránok, vložiť nepriateľský kód, viesť phishingový útok, získať kontrolu nad užívateľovým prehliadačom použitím JavaScript-ového škodlivého kódu a prinútiť užívateľov vykonať príkazy, ktoré vykonať nechceli – útok známy ako cross-site request forgery.

Cross-site request forgery (CSRF) – donúti legitímnych užívateľov vykonať príkazy bez ich vedomia a schválenia. Tomuto typu útoku je veľmi obtiažne zabrániť, pokiaľ aplikácia nie je bez možnosti použitia XSS, vrátane DOM injektáže. S nástupom Ajax technológií, a rastúceho povedomia ako vykonať XSS útoky, sa CRSF stali extrémne sofistikované, či už ako aktívne individuálne útoky, alebo v podobe automatizovaných červov.

Webové scanovacie nástroje môžu pomôcť nájsť tieto bezpečnostné diery, najmä ak sú diery dobre známe. Avšak na nájdenie všetkých potenciálne zraniteľných miest je potrebná analýza zdrojového kódu a test preniknuteľnosti aplikácie. Toto by malo byť vykonané vývojármi pred vydaním akejkoľvek dôležitej webovej aplikácie.

Je potrebné preštudovať konfiguráciu podpornej štruktúry a vhodne ju zabezpečiť. Systémoví administrátori by mali zvážiť pravidelné scanovanie bezpečnostnými scannermi, najmä ak na serveri beží mnoho úžívateľských skriptov, ako napríklad pri web hostingu. Nikto by nemal písať webové aplikácie, kým nemá aspoň základné bezpečnostné schopnosti a vedomosti pre bezpečné aplikácie.

Ochrana pred bezpečnostnými dierami vo webových aplikáciách:

- upgrade-ovať na PHP 5.2 pretože eliminuje mnoho dlhodobých bezpečnostných problémov a podporuje bezpečnejšie rozhranie pre tvorbu aplikácií
- vždy otestovať a použiť záplaty a nové verzie PHP keď sa stanú prístupnými
- ak sa používa väčšie množstvo PHP aplikácií, je vhodné pravidelné bezpečnostné scanovanie
- používať systémy na detekciu a zabránenie prieniku, prípadne systémy na zablokovanie známych útokov
- zakázať aplikácie, ktoré sú známe častým zneužívaním bezpečnostných dier a dlhou dobou na opravenie známych bezpečnostných dier

2.2.2. Služby Windows

Rodina operačných systémov Windows podporuje veľké množstvo služieb, sieťových metód a technológií. Mnoho z nich beží ako služby pomocou programu „services.exe“. Zraniteľnosti v službách ktoréimplementujú funkcie tohto operačného systému sú jedny z najčastejšie zneužívaných. Po inštalácii OS sú početné služby nastavené aby sa spúšťali bez ohľadu na to, či sú naozaj potrebné. Všetky nepotrebné služby by mali byť vypnuté, čo značne zlepší celkovú bezpečnosť. Služba sa musí prihlásiť aby mala prístup k zdrojom operačného systému a mnoho služieb nie je navrhnutých tak, aby mali svoje predvolené účty zmenené. Ak sa zmení heslo alebo celé konto, tak tieto služby nedokážu naštartovať.

V OS Windows sú tri vstavané lokálne účty pre systémové služby:

- ***Local System*** – má plný prístup k počítaču a na sieti umožňuje procesom pod ním bežiacim vystupovať ako počítač. Ak sa služba pod týmto účtom spojí s doménovým kontrolérom, má prístup k celej doméne. Niektoré služby sú predvolene nastavené aby používali tento účet a to by nemalo byť menené. Tento účet nemá heslo pre užívateľský pristup.
- ***Local Service*** – špeciálny zabudovaný účet podobný autentifikovanému užívateľskému účtu. Má umožnený rovnaký prístup k zdrojom ako členovia skupiny „Users“. Tento obmedzený prístup pomáha bezpečnosti ak jednotlivé služby sú kompromitované. Služby pod týmto účtom pristupujú k sieťovým zdrojom pod prístupovými právami „Anonymous“. Tento účet takisto nemá heslo pre užívateľský prístup.
- ***Network Service*** – takisto je podobný autentifikovanému užívateľskému účtu a má rovnaké oprávnenia k prístupu k objektom ako „Local Service“. Pre prístup k sieťovým zdrojom ale služby bežiace pod týmto účtom ale vystupujú ako počítač. Ani tento účet nemá heslo pre užívateľský prístup.

Niekoľko základných systémových služieb sprístupňuje vzdialené rozhrania klientským komponentom cez RPC (Remote Procedure Call – vzdialené volanie služieb). Najčastejšie sú vystavené cez named pipe endpoint prístupný cez protokol CIFS (Common Internet File System), dobre známe TCP a UDP porty a v niektorých prípadoch dočasné TCP a UDP porty. V minulosti bolo množstvo bezpečnostných dier v službách, ktoré mohli

byť zneužité anonymnými užívateľmi. Keď sú tieto bezoečnostné diery zneužíté, dovoľujú útočníkovi rovnaké oprávnenia na napadnutom systéme ako pôvodne služba mala.

Detekcia zraniteľnosti systému:

- použiť akýkoľvek bezpečnostný scanner na kontrolu, či má systém záplatu proti týmto bezpečnostným dieram,
- overiť prítomnosť záplaty kontrolou kľúča v registri, takisto je vhodné skontrolovať, či sú v systéme nainštalované opravené súbory,
- pre kontrolu, či je systém zraniteľný na chybu vo voliteľnej službe je potrebné zistiť či je táto služba zapnutá, čo je možné urobiť cez Service Manager Interface (Services v Administrative tools),

Ochrana pred bezpečnostnými dierami v službách Windows:

- zapnúť Windows Firewall, alebo nainštalovať iný firewall a zaistiť, že pravidlá obmedzujú prístup k systému okrem tých spojení, ktoré sú potrebné. Mnoho bezpečnostných dier sa nachádza na rozhraniach CIFS a teda zaplokovať TCP porty 139 a 445 je základ v predchádzaní vzdialeným útokom. Takisto je dobrým zvykom zablokovať prichádzajúce RPC požiadavky z internetu na port 1024 a vyšší, aby sa predišlo zneužitiu ostatných na RPC založených bezpečnostných dier. Vo firemnom prostredí môže byť Windows Firewall nastavený cez skupinovú politiku v Microsoft Active Directory,
- pre Windows verzie aspoň 2003 SP1 a vyšších použiť Security Configuration Wizard v spolupráci s Windows Firewall-om pre zníženie rizika napadnutia,
- udržovať systémy aktualizované všetkými najnovšími záplatami a Service Pack-mi. Prípadne zapnúť Automatic Updates,
- na vonkajších firewalloch siete použiť egress (iba pakety so zdrojovou adresou zvnútra siete môžu sieť opustiť) [4] a ingress (do siete nesmú z vonku vstúpiť pakety, ktoré majú zdrojovú adresu z vnútra siete) filtrovanie. Takto je možné zamedziť vnútorným aj vonkajším útokom,
- používať systémy detekcie a zabránenia prieniku na sieťovej úrovni aj na úrovni jednotlivých systémov pre zabránenie alebo aspoň zistenie útokov zneužívajúcich tieto bezpečnostné diery,

- zmenšiť riziko napadnutia vypnutím nepotrebných služieb. Hlavne v skorších systémoch boli mnohé služby zapnuté, čo zvyšovalo riziko útoku. Od Windows 2008 zvolila firma MS inú metódu, tzv. „secure by default“, čo sa prakticky prejavuje tak, že všetky nepotrebné služby a vlastnosti systému sú predvolene vypnuté a pre používanie je ich nutné doinštalovať.

2.2.3. Služby UNIX/Linux/Mac OS

Väčšina UNIX/Linux systémov obsahuje v základnej inštalácii viacero štandardných služieb. Mac OS často trpí rovnakými bezpečnostnými dierami ako UNIX, pretože je na ňom založený. Nepotrebné služby by mali byť vypnuté a všetky servery prístupné z otvorených sietí by mali byť chránené firewall-om. Služby poskytujúce vzdialené prihlásenie a vzdialenú službu nemôžu byť jednoducho zablokované firewall-om. Bezpečnostné diery typu Buffer overflow a chyby v autentifikačných funkciách často umožňujú vykonanie škodlivého kódu, niekedy aj s administrátorskými oprávneniami. Takže je nutné získavanie informácií o zraniteľnosti a rapídne inštalovanie záplat. Bezpečnostné diery Buffer overflow sú stále nachádzané.

Tieto služby, aj keď sú opravené najnovšími záplatami, môžu byť príčinou kompromitácie serveru. Útoky metódou hrubej sily na vzdialené služby ako SSH, FTP alebo telnet sú najbežnejšou formou útoku na servery prístupné z Internetu. V poslednej dobe majú aj červy a podobný malware vstavané brute-force možnosti. Systémy so slabými heslami na užívateľských účtoch sú aktívne kompromitované, často je následne použitý postup na získanie vyšších oprávnení a do systému sú nainštalované rootkit-y. Dôležité je, že metóda hrubej sily je jednoducho použiteľná aj na dobre zaplátané systémy.

Pre vzdialený prístup k systému by sa mal použiť SSH alebo iný kryptovaný protokol. Ak je použitá aktuálna a plne zaplátaná verzia SSH, je možné túto službu považovať za bezpečnú. Avšak aj tak je možné systém úspešne napadnúť hádaním hesla metódou hrubej sily. Tomuto je možné zabrániť použitím autentifikácie verejným kľúčom. Pre ostatné podobné služby je potrebný napríklad audit hesiel, aby sa overilo či sú dostatočne komplexné aby odolali metóde hrubej sily.

Minimalizáciou počtu služieb bežiacich na systéme sa tiež získa dodatočná bezpečnosť. Mnoho služieb je možné zneužiť a isté kombinácie služieb, napríklad HTTP a FTP server zdieľajúce rovnaké adresáre, veľmi jednoducho.

Štandardné inštalácie (či už od dodávateľa alebo administrátora) operačných systémov alebo sieťových aplikácií môžu obsahovať veľké množstvo nepotrebných a nepoužívaných služieb. V mnohých prípadoch neistota o následnom použití operačného systému alebo aplikácie vedie k inštalácii mnohých programov pre prípad, že budú v budúcnosti potrebné. Toto značne zjednodušuje proces inštalácie, ale zanechá množstvo nepotrebných služieb a užívateľských účtov s predvolenými, slabými alebo bežne známymi heslami. Na objavenie potencionálnych bezpečnostných dier po predvolených inštaláciach je možné efektívne použiť port scanner.

Ochrana pred bezpečnostnými dierami v UNIX/Linux/Mac OS:

- zoscanovať server port scannerom alebo bezpečnostným scannerom pre zistenie nadbytočných služieb, ktoré nie sú používané žiadnou potrebnou aplikáciou, a tie následne vypnúť;
- pravidelne inštalovať posledné záplaty od výrobcu aby sa zabránilo bezpečnostným chybám v bežiacich službách. Management záplat je kritická časť bezpečnosti systému;
- Systémy managementu záplat pomáhajú nájsť neupdatované systémy. V sieťach kde beží niekoľko serverov je dôležité skontrolovať všetky z nich, lebo aj len jeden robí zraniteľnou celú sieť;
- v kritických systémoch je dôležité použiť monitorovanie logov v reálnom čase;
- tam kde je to možné je vhodné spúšťať služby na iných portoch ako predvolených, pretože automatické scannery väčšinou hľadajú bežiace služby len na predvolených portoch;
- použiť hardware alebo software firewall a IDS/IPS na detekciu a blokovanie útokov a ochranu potrebných služieb. Ak je to možné, je výhodné obmedziť zoznam adries pre vzdialenú prácu a služby;
- nepoužívať predvolené heslá na žiadnom účte, nedovoliť slabé heslá a heslá založené na slovách zo slovníka;
- obmedziť počet neúspešných prihlásení k službám a účtom;
- minimalizovať počet účtov, ktorými sa je možné vzdialene prihlásiť, zakázať vzdialený prístup pre konto „root“;
- zakázať zdieľané kontá a nepoužívať jednoduché názvy účtov ako: test, guest, admin, sysadmin;

- zaznamenávať počet neúspešných prihlásení, veľký počet vyžaduje kontrolu systému, či nebol kompromitovaný;
- zvážiť autentifikáciu osobným certifikátom;
- obmedziť množstvo funkcií vykonávaných jedným systémom, zlá konfigurácia viacerých služieb je oveľa viac náchylná na zneužitie.

2.3. Bezpečnostné diery v klientských systémoch

2.3.1. Webové prehliadače

Microsoft Internet Explorer je najrozšírenejší webový prehliadač a je predvolene nainštalovaný na všetkých dnešných operačných systémoch Microsoft Windows. Nezaplátané a staršie verzie Internet Explorer-u obsahujú množstvo bezpečnostných dier, ktoré môžu spôsobiť napríklad porušenie pamäte alebo umožniť vykonanie vzdialeného kódu. Nejnebezpečnejšie sú tie, ktoré umožňujú vykonanie kódu bez vedomia a nezávisle na činnosti užívateľa, napríklad keď užívateľ navštívi zlomyselnú stránku alebo si prečíta takýto e-mail. Kód na zneužitie mnohých bezpečnostných chýb v Internet Explorer-i je verejne dostupný. Naviac môže byť Internet Explorer donútený aby zenužil chyby v iných základných častiach OS Windows ako napríklad HTML Help a jadro generovania grafiky. Častými sú aj chyby v ActiveX moduloch od Microsoftu ale aj mnohých iných tvorcov software-u. Tieto sú tiež zneužívané cez Internet Explorer.

Mozilla Firefox je druhý najpoužívanejší webový prehliadač. Taktiež trpí množstvom bezpečnostných chýb. Vydávané záplaty adresujú najmä verejne dobre známe chyby. Podobne ako v prípade Internet Explorer-u staré nezaplátané verzie obsahujú množstvo bezpečnostných dier, umožňujúcich porušenie pamäte a vykonanie vzdialeného kódu. Webové stránky zneužívajúce chyby v prehliadačoch často používajú niekoľko postupov a spustia vhodný postup pre zneužitie bezpečnostnej diery v prehliadači, ktorý potenciálna obeť používa.

S nástupom multimediálnych služieb vo webových prehliadačoch paralelne narastá množstvo prídavných modulov pristupujúcim k dokumentom cez MIME. Tieto moduly, ako napríklad Macromedia Flash alebo Shockwave, často poskytujú skriptovacie možnosti na klientskej strane. Väčšinou sú polotransparentne nainštalované webovou stránkou, takže úžívateľ často netuší, že na jeho systém sa inštaluje zneužiteľný program. Takéto moduly poskytujú ďalšie možnosti na zneužitie a kompromitovanie systému užívateľ ktorého navštívi zlomyselné webové stránky.

Niektoré moduly ako napríklad Adobe Reader a Apple Quicktime vykonávajú kontrolu verzií a ponúkajú update, tento je väčšinou vnímaný ako príťaž a užívateľmi ignorovaný. Taktiež je často ťažké zistiť aká verzia modulu je nainštalovaná. Napríklad

v systéme môže byť nainštalovaných niekoľko verzii kvôli spätnej kompatibilite, ale pre užívateľa je veľmi obtiažne zistiť, ktorá sa aktuálne používa.

Ochrana pred bezpečnostnými dierami vo webových prehliadačoch:

- ak sa používa OS Windows a Microsoft Internet Explorer, najľahšia cesta je nainštalovať Windows XP SP3. Zlepšená bezpečnosť a zabudovaný Windows Firewall rapídne znížia riziko napadnutia. V prípade nemožnosti použitia systému s SP3 uvažovať nad iným prehliadačom;
- updatovať na Internet Explorer 8, keďže prináša značné zlepšenie bezpečnosti oproti svojim predchodcom. Posledná verzia IE8 je distribuovaná Microsoftom cez Windows Update ako kritická záplata;
- aktualizovať systémy najnovšími záplatami a service pack-mi, ak je to možné, zapnúť „Automatické aktualizácie“;
- venovať pozornosť Microsoft Security Advisories [5], implemetáciou odporúčaných opatrení pred vydaním aktualizácie značne zníži vystavenie zero day útokom;
- použiť UAC (user access control) na Windows Vista, alebo aplikácie ako Microsoft DropMyRights na Windows XP pre zníženie pístupových práv s ktorými procesy bežia, čo zabráni vzdialenému kódu bežať pod administrátorskými právami;
- skontrolovať Browser Helper Objekty antispyware scannerom, keďže množstvo spyware sa nainštaluje ako BHO. BHO sú spúšťané po každom spustení IE a rozširujú jeho možnosti;
- použiť antivírusový program, anti-spyware a malware detektor pre zablokovanie nebezpečných HTML skriptov;
- operačné systémy Windows 98/Me/NT a staršie už nie sú výrobcom podporované a nie sú na ne vydávané aktualizácie, je teda potrebná aktualizácia na novší operačný systém;
- ako núdzové riešenie je možné použiť iný webový prehliadač, napríklad Mozilla Firefox, ktorý nepodporuje ActiveX a je voči takýmto útokom imúnny.

2.3.2. Kancelárske programy

Najpoužívanejším kancelárskym a e-mailovým balíkom je Microsoft Office. V aplikáciách MS Office bolo nájdených množstvo bezpečnostných dier pričom niekoľko z nich bolo zero day pre ktoré existoval postup zneužitia, alebo boli verejne vydané

technické detaily alebo dôkaz zneužiteľnosti pred tým ako Microsoft sprístupnil záplaty týchto bezpečnostných dier.

Postupy zneužitia bezpečnostných dier v kancelárskych programoch:

- útočník pošle špeciálne vytvorený dokument ako prílohu v elektronickej pošte, keď je príloha otvorená, zlomyselne navrhnutý obsah dokumentu zneužije bezpečnostné diery v kancelárskom programe;
- útočník zdieľa takýto dokument na webovom serveri alebo v zdieľanom adresári a naláka úžívateľa, aby ho stiahol a otvoril. Vo väčšine prípadov Internet Explorer otvorí dokument sady Office automaticky, takže k útoku často stačí len prezeranie zlomyselnej stránky alebo adresára;
- útočník používa vlastný NNTP (news) server alebo napáda RSS správy a tak pošle zlomyselné dokumenty klientom news a RSS systémov.

Ocharana pred bezpečnostnými dierami v kancelárskych programoch:

- aktualizovať systémy najnovšími záplatami a service pack-mi, ak je to možné, zapnúť „Automatické aktualizácie“;
- neotvárať prílohy z neznámym alebo nedôveryhodným pôvodom, v prípade neočakávanej prílohy z dôveryhodného zdroja taktiež postupovať opatrne;
- vypnúť v IE automatické otváranie dokumentov MS Office;
- neotvárať neznáme linky na diskusných fórach alebo z e-mailov;
- nastaviť MS Outlook (súčasť MS Office) alebo Outlook Express (Windows Mail vo Windows Vista) na zvýšenú bezpečnosť;
- použiť bezpečnostný scanner na zistenie miery rizika;
- použiť IDS/IPS a antivírusový program pre zabránenie zlomyselným serverovým odpovediam a dokumentom aby sa dostali k užívateľovi;
- použiť systémy filtrovania e-mailov a webového obsahu na úrovni siete, aby sa zabránilo zlomyselným dokumentom v prístupe k užívateľovi.

3. Aplikácie na sledovanie bezpečnosti siete

V tejto časti je uvedený opis základných vlastností niektorých bežne dostupných programov, ktoré administrátor siete, intranetu alebo serveru môže použiť pre zjednodušenie práce. Ide o programy a aplikácie umožňujúce analýzu stavu siete, detekciu zraniteľností v počítačoch a sieťovej infraštruktúre, detekciu chýb a zraniteľností vo webových aplikáciách, ale aj na sledovanie zmien v sieti. Opis je zameraný na vymenovanie vlastností a možností jednotlivých programov a ich zhodnotenie.

3.1. Wireshark

Wireshark je nástroj na zachytávanie paketov a analyzovanie sieťovej prevádzky. Pracuje na operačných systémoch rodiny Windows (NT/2000/XP/2003/Vista/2008) aj UNIX (Linux, BSD, Solaris). Je šírený pod licenciou GNU GPLv2, takže je možné stiahnuť jeho kompletné zdrojové kódy. Niektoré časti sú licencované inými licenciami kompatibilnými s GNU GPL.

Wireshark je možné použiť v sieťach typu Ethernet a štandardu pre káblové modemy. Počúvané rozhranie dokáže automaticky prepnúť do promiskuitného režimu, takže sú zaznamenané aj pakety, ktoré nie sú určené pre danú sieťovú kartu (NIC), ale predsa sa na tomto rozhraní objavia. Ak je teda použitým sieťovým prvkom rozbočovač, tak Wireshark dokáže zachytiť všetkú sieťovú prevádzku. V prípade prepínaného Ethernetu je potrebné použiť iné techniky.

Sieťovú prevádzku je možné zachytávať aj podľa dopredu nastaveného filtra z regulárnych výrazov. Je však možné zachytávať všetku sieťovú prevádzku a filter aplikovať následne. Taktiež je možné tieto filtre kombinovať. Zachytené pakety je možné uložiť pre offline analýzu. Zachytávané pakety je možné zobrazovať v reálnom čase spolu s rozlišovaním mien zariadení na transportnej (názvy well-known portov), sieťovej (reverzný DNS lookup) a MAC (identifikácia výrobcu sieťovej karty) vrstve.

Hlavné okno Wireshark-u je štandardne rozdelené na tri časti. V prvej je výpis paketov vyhovujúcich nastaveným filtrom s jednoduchým popisom. V druhej je obsah vybraného paketu rozdelený do jednitlivých položiek použitého protokolu a v tretej je kód paketu ako bol zachytený zo siete.

V oblasti analýzy Wireshark podporuje veľké množstvo protokolov a typov paketov (takmer 1000) a to od 2. až po 7. vrstvu. Je možné si aj nadefinovať vlastné pravidlá a tým

pridať iný, užívateľský protokol. K dispozícii je aj nástroj „Expert info“, ktorý zosumarizuje všetky chyby, ktoré boli počas zachytávania zaznamenané (chybné kontrolné súčty, neúplné pakety a pod.), takže vytvorí rýchly prehľad o stave danej siete.

Pokročilé analytické funkcie:

- **Protocol Hierarchy** – zobrazí štatistickú skladbu protokolov v jednotlivých zachytených paketoch a vyznačí stromovú štruktúru použitých protokolov. Toto umožňuje rýchlo a ľahko zistiť, aký typ premávky je na sieti dominantný a upraviť tak vlastnosti siete pre tento typ dát. Takisto zjednodušuje detekciu zakázaných prenosov a rôznych chýb.
- **Conversations** – tabuľkovo zobrazí konverzácie na rôznych vrstvách a za použitia rôznych protokolov, ktoré boli v sieti zachytené. Toto umožňuje rýchlo zisiť rôzne typy neželaných prenosov v sieti.
- **Endpoints** – veľmi podobné predchádzajúcej funkcii, avšak zobrazuje štatistiky týkajúce sa jednotlivých koncových bodov a nie na základe konverzácií. Umžňuje ľahko nájsť konkrétne zariadenie, ktoré je zdrojom neobvyklého množstva sieťového prenosu.
- **I/O Graphs** – graficky znázorňuje ľubovoľné informácie o zachytenej sieťovej prevádzke. Umožňuje zobrazenie až 5. priebehov v jednom grafe s plne nastaviteľnými rozmermi ôs a typom vykresleného priebehu.
- **VoIP Calls** – zobrazuje štatistické informácie ako začiatok a koniec hovoru, počet prenesených paketov, jednotlivých účastníkov a použitý komunikačný protokol a to ako o VoIP volaniach, tak aj o faxových prenosoch. Veľmi užitočné v telekomunikačných službách.
- **GSM** – obsahuje podmenu sprístupňujúce rôzne štatistické informácie o GSM prevádzke. Wireshark dokáže analyzovať komunikáciu medzi BTS stanicami, všetku komunikáciu medzi koncovým zariadením a BTS (registrácia do siete, dohadovanie použitých kanálov, hand-over a pod.) a aj režijné informácie jednotlivých komunikačných protokolov (SMS, CSD a veľmi podrobne GPRS)
- **RTP** – štatistika komunikácie realizovanej pomocou protokolu pre prenos dát v reálnom čase. Je možné vypísať všetky RTP streamy so základnými informáciami a podrobne analyzovať konkrétny zaujímavý stream. Umožňuje získať prehľad o RTP prenosoch v sieti. RTP sa používa na prenos streamovaného videa a audia (internetové rádiá a televízne vysielania) ale aj v niektorých VoIP implementáciách (napr. SIP)

- **SIP** – štatistiky zaslaných kontrolných SIP paketov podľa stupňa kritickosti. Umožňuje rýchlo zistiť chyby v SIP ústredniach alebo sieťovej infraštruktúre zabraňujúce správnemu fungovaniu SIP VoIP protokolu.
- Ďalšie jednoduché analytické funkcie, ako napíklad analýza DHCP a BOOTP správ, HTTP prenosov, multicastového vysielania, WLAN sieťovej prevádzky a jednoduché štatistiky zastúpenia jednotlivých IP adries, TCP/UDP portov, alebo priemernej veľkosti paketov a pod.

3.2. Tenable Nessus

Tenable Nessus je nástroj zo skupiny vulnerability scanner-ov, slúži na automatické hľadanie bezpečnostných chýb v počítačoch a sieťovej infraštruktúre. Jedná sa o platený produkt, avšak pre nekomerčné použitie si je možné registrovať aj bezplatnú verziu. Má širokú škálu podporovaných hostiteľských operačných systémov a to ako medzi UNIX-like, tak aj z rodiny MS Windows. Aplikácia je písaná štýlom klient-server. Klient sa spúšťa na počítači operátora a server môže bežať buď na tom istom počítači, alebo na inom počítači, bez problémov aj s iným OS.

Po spustení klientskej aplikácie sa zobrazí jednoduché okno, kde sa je možné pripojiť k Nessus Serveru. Potom je možné určiť rozsahy kontrolovaných IP adries a vybrať scanovací profil. Kliknutím na „Scan Now“ sa spustí samotné scanovanie. Priebeh scanovania je naznačený s stavovom riadku a zistené skutočnosti sú užívateľovi zobrazované v reálnom čase. Výpis je formátovaný do stromovej štruktúry systémom IP adresa a nájdené otvorené TCP a UDP porty. Po vybratí konkrétneho portu sa zobrazia informácie o službe na ňom bežiacej. Zistenia jednotlivých modulov sú rozdelené do štyroch kategórií podľa závažnosti a vplyvu na bezpečnosť systému. Každá služba je farebne označená podľa závažnosti najvyššie kategorizovanej bezpečnostnej diery, čo zjednodušuje orientáciu a upriamuje pozornosť na najdôležitejšie problémy. Podobne je podľa závažnosti najvyššie kategorizovanej bezpečnostnej diery nájdenej na danom prvku farebne označená IP adresa. V jednom čase môže klient vykonávať iba jednu kontrolu, avšak počet klientov napojených na jeden server nie je obmedzený, rovnako ako počet klientov spustených na jednej stanici.

Rozdelenie chýb podľa závažnosti v Tenable Nessus:

- **None** – modrá – modul neoznačil stav služby za nebezpečný;
- **Low** – modrá – boli zistené malé nedostatky, ktoré je možné zneužiť k získaniu málo dôležitých informácií o danej stanici, zneužitie na získanie kontroly nad strojom nie je možné;
- **Medium** – oranžová – v službe sa vyskytuje bezpečnostná diera, alebo chyba v nastaveniach, ktorá umožňuje použitie služby bez autorizácie; skutočný význam pre útočníka môže byť rôzny;
- **High** – červená – bola nájdená bezpečnostná diera, ktorá je priamo využiteľná na získanie kontroly nad cieľovým počítačom, alebo pre DoS útok; jedná sa o chyby, ktoré je nutné úrýchlene opraviť;

Tenable Nessus server pracuje s použitím plugin-ov (zásuvných modulov), pričom každý z nich je zameraný na konkrétnu bezpečnostnú dieru v jednom alebo v skupine serverov. Týchto zásuvných modulov je vyše 25000, pričom približne raz za mesiac sú vydávané nové a existujúce sú dopĺňané a zlepšované. Vďaka tomuto systému je možné nové moduly vytvárať ľahko a za kratší čas, bohužiaľ aktualizácia pozostáva z rozbalenia databázy, kontroly a prepísania starých verzií modulov novými a opätovných zabalením. Tento proces vďaka počtu modulov trvá značný čas. Toto však vzhľadom na potrebu len jedného Nessus servera nie je veľkým problémom.

Základné ovládanie programu je jednoduché, avšak okno na editáciu profilu umožňuje komplexné a rozsiahle nastavenia. Od základných nastavení, ako počet paralelných spojení s jedným sieťovým zariadením, počtu paralelne scanovaných sieťových zariadení alebo ktoré port-scannery použiť až po pokročilé typu detekcia zahltenia linky a úprava správania v takomto prípade. Takisto je možné zápnúť „Safe Scan“, čo spôsobí, že sa nepoužijú kontroly, ktoré by mohli mať nepriaznivé následky na kontrolované zariadenie. Samozrejmosťou je výber scanovacích modulov, ktoré sa majú použiť. Posledná záložka je venovaná detailným nastaveniam jednotlivých modulov.

3.3. Acunetix Web Vulnerability Scanner

Tento nástroj slúži na detekciu bezpečnostných dier vo webových serveroch a webových aplikáciách a takisto na analýzu kódu webových aplikácií. Jedná sa o komerčný

produkt, zo stránok produktu je ale možné stiahnuť demonštračnú verziu, ktorá však umožňuje iba kontroly na Cross-site scripting (XSS).

Užívateľské rozhranie je organizované do stromovej štruktúry, pričom s najdôležitejšími úlohami pomôžu sprievodcovia. Stromová štruktúra je rozsiahla a program umožňuje veľké množstvo nastavení, ale tieto sú logicky usporiadané a po chvíli práce je ovládanie jednoduché a intuitívne.

Po spusteni program najprv kontaktuje server s aktualizáciami a ich prípadný výskyt oznámi užívateľovi. Aktualizácie sú v dvoch variantách. Prvá možnosť je tzv. Patch aktualizácia, ktorá opravuje bezpečnostné a iné chyby v programe. Jedná sa o opravenú verziu postihnutého súboru. Tento je možné priamo z prostredia programu stiahnuť a program ho sám aktualizuje. Druhou možnosťou je tzv. build aktualizácia, čo znamená vydanie novej verzie programu. Tieto aktualizáacie prinášajú nové funkcie a samozrejme opravujú chyby z minulých verzií. Jedná sa o inštalačný balík, ktorý je možné priamo v prostredí programu stiahnuť zo stránky výrobcu a spustiť inštaláciu.

Najjednoduchší spôsob ako začať s programom pracovať, je použitie sprievodcu „New Scan“. V prvom kroku sprievodcu sa nastavuje lokalita, ktorá má byť scanovaná. Je možné vybrať konkrétnu URL adresu, súbor so zoznamom niekoľkých URL, či nechať program aby vuhľadal http a HTTPS servery v určenom rozsahu IP adries. Tak isto je možné nechať scanovať dopredu uloženú verziu web stránky, ktorú získa modul nazývaný „Site Crawler“. Táto možnosť je praktická v prípade, že chceme testovanie vykonať opakovane s pozmenenými parametrami a nechceme webový server nadmerne zaťažovať, alebo ak chceme z nejakého dôvodu testovať stav stránky v istej dobe v minulosti.

V druhom kroku program kontaktuje zadané URL adresy (alebo prescanuje zoznam IP adries) a zistí základné informácie o webových serveroch, operačnom systéme a prípadných CGI moduloch a iných rozšíreniach webového servera. Podľa zistených informácií program automaticky navrhne zameranie scanovania, užívateľ ale tieto nastavenia môže upraviť.

V treťom kroku sa určujú nastavenia pre „Site Crawler“. „Site Crawler“ slúži na automatické vytvorenie obrazu stránky. Vyhľadá všetky súbory a adresáre na webovom serveri, na ktoré sú odkazy v iných, už nájdených súboroch. Postupuje sa od koreňového súboru (index.html a pod.) a vytvorí sa zoznam súborov na ktoré sú v tomto súbore odkazy. V ďalších krokoch sú rovnakým spôsobom spracované aj tieto novonájdené súbory. Program umožňuje nastavenie detailov tejto časti, pričom predvolené nastavenia umožňujú najdetailnejšie scanovanie a preto ich vo väčšine prípadov nie je potrebné meniť.

Vo štvtom kroku sa vyberajú možnosti pre vlastné hľadanie bezpečnostných chýb na stránke. Je možné vybrať niektorý z predpripravených profilov, pričom každý z nich aktivuje vhodnú špecifickú skupinu testov. Predvolený profil „default“ použije všetky testy a je teda rozumnou voľbou pre všeobecný test. Ďaľej je možné vybrať scanovací mód, ktorý určuje správanie scanera pre túto úlohu a tiež sa tu nastavujú doplňujúce možnosti ako použitie port scannera pre nájdenie otvorených portov, či zmenu parametrov HTTP hlavičky pre emuláciu prístupu z konkrétneho webového prehliadača.

Piaty krok slúži na nastavenie HTTP autentifikácie. Acunetix WVS podporuje Basic a NTLM autentifikáciu, ale umožňuje aj nahratie a použitie vlastnej prihlasovacej sekvencie. Zaznamenanie vlastnej sekvencie je uskutočnené v zjednodušenom webovom prehliadači a použitý spôsob umožňuje použitie širokého spektra autentifikačných metód. Posledný, šiesty krok sumarizuje nastavenia a umožňuje spustenie scanovania.

Po spustení sa začne s analýzou štruktúry a následne hľadaním bezpečnostných dier. Nájdené bezpečnostné diery a zraniteľnosti sú zobrazované v reálnom čase a podľa závažnosti sú rozdelené do štyroch kategórií. Najnižšia kategória, klasifikovaná ako informácie obsahuje upozornenia na povolenú funkciu „autocomplete“, ktorá ukladá vyplňované informácie na klientskom systéme pre neskoršie opakované použitie. Takto sú ukladané aj citlivé informácie ako heslá alebo iné osobné informácie a je ich možné z počítača extrahovať. Do kategórie informácií tiež patria upozornenia na nefunkčné linky nájdené počas scanovania. Ďaľšia kategória je klasifikovaná ako nízky stupeň ohrozenia a patria sem najmä upozornenia na zasielanie hesiel v čistom textovom formáte (bez kryptovania). Takto zasielané heslá je možné cestou odchytiť a neskôr použiť. Tretia kategória sú stredne závažné zraniteľnosti a patria sem problémy ako používanie šifrovacích a autentifikačných protokolov slabým alebo žiadnym kryptovaním a podobne. Do najvyššej skupiny označenej vysoké bezpečnostné riziko patia závažné bezpečnostné diery ako používanie starých verzií programov (webový server, skriptovací zásuvný modul, databázový server, atď.), autentifikačných a kryptovacích protokolov so známymi chybami a existujúcou opravou, alebo SQL injection a XSS bezpečnostné diery.

Každá nájdená zraniteľnosť je doplnená jednoduchým popisom, dopadom na bezpečnosť stránky a určením postihnutej časti. Takisto sú tu informácie ako túto zraniteľnosť opraviť alebo odstrániť a niekoľko HTTP linkov na stránky venujúce sa danej problematike. K väčšine zraniteľností je možné rozbaliť rubriku s podrobnejšími

informáciami, ktoré stačia na vytvorenie predstavy ako možno túto chybu zneužiť a umožnia zhodnotiť jej dôležitosť a prípadne postup pre jej odstránenie.

Program ďaľej obsahuje množstvo nástrojov, ktoré umožňujú analýzu a testovanie webevých stránok a aplikácií. Okrem spomínaných nástrojov „Site Crawler“ a vyhľadávania HTTP serverov v zadanom rozsahu IP adries sú to nástroje na vyhľadávanie DNS poddomén, SQL injektáž, editor a generátor HTTP požiadaviek, porovnávanie výsledkov vaicerých scanov a nástroj na testovanie autentifikácie. Tento umožňuje skúšanie mien a hesiel slovníkovou metódou. Podporuje autentifikácie Basic, NTLM a HTML Form, ktorá umožňuje vybrať formulárové prvky zodpovedajúce menu a heslu. Je nutné vybrať prvok pre meno aj pre heslo a preto bohužiaľ nie je možné použiť tento modul v prípade, že webová stránka používa autentifikáciu len za pomoci hesla.

Vlajkovou technológiou nástroja Acunetix WVS je „AcuSensor“. Táto technológia umožňuje identifikovať väčšie množstvo zraniteľností ako bežné webové scanery a popri tom generuje výrazne nižšie množstvo falošných identifikácií, pričom dokáže tiež určiť presné miesto v kóde, kde sa zraniteľnosť nachádza. Podstatou tohto systému je kombinácia scanovania metódou čiernej skrinky a umiestnenia software-ových senzorov do zdrojového kódu webovej aplikácie. Touto kombináciou sa anulujú nedostatky oboch spôsobov použitých samostatne. Scanovanie metódou čiernej skrinky nedokáže určiť ako sa aplikácia pod povrchom správa a analýza zdrojového kódu zase neumožňuje spoľahlivo predpovedať reakciu aplikácie na prípadný útok. Senzor existuje pre ASP.NET aj pre PHP skriptovacie jazyky a inštalácia a nastavenie je rýchle a dobre dokumentované v nápovede programu. Táto technológia je zameraná skôr na vývojárov, administrátorovi väčšinou postačí bežné scanovanie, ktoré navyše nevyžaduje inštaláciu software a zmeny nastavení webového servera.

3.4. Nmap

Nmap je open-source nástoj na prieskum siete a bezpečnostný audit. Je dostupný pre OS rodiny Un*x a Windows. Vyvinul ho známy hacker Gordon „Fyodor“ Lyon v roku 1997 a pôvodne začínal ako port scanner ktorý spájal najzaujímavejšie vlastnosti vtedy dostupných port scannerov a pridával niekoľko nových nápadov. Práca na tomto programe odvtedy stále napreduje, so zaujímavého nápadu uskutočneného jedným programátorom sa stal veľký projekt s veľkou užívateľskou základňou a počtom prispievateľov. Nmap je najznámejším a najpoužívanejším nástrojom v oblasti počítačovej bezpečnosti. Od

počiatkov sa jeho použitie výrazne rozšírilo z jednoduchého port scannera pridaním skriptovacieho enginu a v dnešnej dobe je použiteľný aj napríklad na detekciu infekcie počítačovým červom Conficker. Nmap má integrované všetky známe spôsoby scanovania portov. Jeho zaujímavou vlastnosťou je heuristická analýza služieb, čo znamená, že nájdeným otvoreným portom môže priradiť bežiacu službu okrem tabuľky dobre známych portov aj pomocou zasielania požiadavkov a analýzy odpovedí. Výhoda je tá, že k službe dokáže určiť aj typ a verziu servera. Toto je priamo užitočné pri analýze bezpečnosti siete, lebo nmap takto umožňuje jednoducho katalogizovať bežiace programy a ich verzie, a tie je potom porovnať s rôznymi databázami bezpečnostných dier. Veľmi zaujímavá možnosť, ktorá je jednou z najčastejšie využívaných, je možnosť detekcie operačného systému. Na toto Nmap používa rozsiahlu databázu pravidiel správania jednotlivých operačných systémov a z nich následne počíta pravdepodobnosť že ide o konkrétny OS. Toto je taktiež doplnené databázou odtlačkov jednotlivých OS, ktorá je tak rozsiahla, že v absolútnej väčšine prípadov dokáže Nmap určiť operačný systém scanovaného počítača so 100% pravdepodobnosťou.

Skriptovací engine Nmap-u boll pridaný v priebehu roka 2008. Je založený na skriptovacom jazyku Lua a umožňuje užívateľom Nmap-u jednoducho vytvárať vlastné detekčné a scanovacie moduly aj bez znalosti programovania a v kratšom čase. Scriptovací engine spôsobil, že funkcionalita programu sa začala veľmi rýchlo rozširovať. Záber možností Nmap-u je tak široký, že sa jedná o naozaj univeryálny nástroj. Skriptovací engine umožnil aj dovtedy v Nmap-e nevidené priame testovanie prieniku ako doplnok k offline prehľadávaniu databáz bezpečnostných dier. Samozrejme Nmap zatiaľ nemá toľko testovacích modulov ako napríklad Metasploit Framework, avšak ich počet sa rýchlo zvyšuje.

Nmap je nástrojom ovládaným pomocou parametrov zadaných v príkazovom riadku. Počas jeho histórie bolo pre Nmap vyvinutých niekoľko grafických užívateľských prostredí, ktoré zjednodušovali ovládanie a nevyžadovali od užívateľa aby si zapamätal značné mnoťstvo prepínačov. Niektoré dokonca pridávali vlastné zaujímavé funkcie ako napríklad grafické zobrazenie topológie scanovanej siete. Tieto GUI boli vytvorené inými autormi a boli distribuované samostatne. Väčšina z nich sa neujala a nedosiahli veľké počty užívateľov, niektoré prestali byť vyvíjané krátko po začiatku. Kvalitatívne najvyššie bol zo všetkých prostredí Zenmap GUI. Tento sa v roku 2008 stal natoľko používaným a prepracovaným, že bol zahrnutý do oficiálnych distibučných balíčkov Nmap-u.

Samozrejme pre skriptovanie nemá grafické užívateľské prostredie veľký význam a pokročilí uživatelia ho pravdepodobne tiež používať nebudú, ale Zenmap GUI existuje a zjednodušuje prácu a proces učenia ľudom začínajúcim v oblkasti počítačovej bezpečnosti.

Nmap má množstvo parametrov príkazového riadku, ktorými je možné ovplyvniť takmer všetky aspekty jeho správania. Nasleduje popis najdôležitejších z nich:

Špecifikácia cieľa:

- [špecifikácia cieľa]: dokáže spracovať DNS či NetBIOS záznamy, IP adresy a rozsahy IP adries v mnohých formátoch zápisu (príklad: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254)
- -iL <inputfilename>: načíta zoznam cieľov zo súboru
- -iR <num_hosts>: vyberie num_hosts náhodných cieľov
- --exclude <host1[,host2][,host3],...>: vynechá zadané ciele alebo rozsahy cieľov
- --excludefile <exclude_file>: načíta zoznam cieľov pre vynechanie zo súboru

Overovanie cieľov:

- -sL: List Scan – len vypíše ciele, ktoré by boli scanované pri aktuálnom nastavení parametrov
- -sP: Ping Scan – iba zistí, či sú zadané ciele dostupné použitím ICMP echo scanovania (ping)
- -PN: Nebude overovať dostupnosť cieľov, ďalšie kroky sú vykonávané akoby boli všetky špecifikované ciele dostupné.
- -PS/PA/PU[portlist]: TCP SYN/ACK alebo UDP overovanie konkrétnych portov
- -PE/PP/PM: ICMP echo, timestamp a netmask request overovanie cieľov
- -PO[protocol list]: IP Protocol Ping
- -n/-R: Nikdy nevykonávať DNS rozlišovanie/Vždy rozlišovať [default: niekedy]
- --dns-servers <serv1[,serv2],...>: Špecifikácia vlastných DNS serverov, ktoré sa majú použiť pre scanovanie
- --system-dns: Použiť DNS rozlišovacie schopnosti operačného systému
- --traceroute: Trasovanie cesty k cieľom
- Scanovacie techniky:
- -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scanovanie

- -sU: UDP scan
- -sN/sF/sX: TCP Null, FIN, a Xmas scanovanie
- --scanflags <flags>: Vlastné nastavenie príznakov v TCP hlavičke
- -sI <zombie host[:probeport]>: Idle scan
- -sO: IP protokol scan
- -b <FTP relay host>: FTP bounce scan

Špecifikácia portov a poradia scanovania:

- -p <port ranges>: Iba určené porty (príklad: -p22; -p1-65535;
  -pU:53,111,137,T:21-25,80,
- 139,8080)
- -F: rýchly režim – scanuje menšie množstvo portov ako štandardne
- -r: Scanovanie portov v numerickom poradí, bez náhodného usporiadania
- --top-ports <number>: Bude scanovať <number> najčastejšie otvorených portov
- --port-ratio <ratio>: Bude scanovať porty bežnejšie ako <ratio> (rozsah: 0 - 1)

Detekcia služieb:

- -sV: Skúmať porty pre detekciu služieb a ich verzií
- --version-intensity <level>: Intenzita scanovania od 0 (málo testov) po 9 (vyskúša všetky testy)
- --version-light: Iba najpravdepodobnejšie testy (intenzita 2)
- --version-all: Vyskúša všetky testy (intenzita 9)
- --version-trace: Zobrazí detailné informácie o priebehu detekcie služieb (pre debugovanie)

Skriptovací engine:

- -sC: ekvivalent --script=default
- --script=<Lua scripts>: <Lua scripts> je čiarkou oddelený zoznam skriptov, adresárov, súborov so skriptami alebo kategórií skriptov
- --script-args=<n1=v1,[n2=v2,...]>: umožňuje zadať skriptom parametre
- --script-trace: Zobrazí všetky odoslané a prijaté dáta
- --script-updatedb: Aktualizuje databázu skriptov

Detekcia OS:

- -O: zapne detekciu OS
- --osscan-limit: Obmedzí detekciu OS na iba sľubné cieľe (spĺňajúce vhodné podmienky)
- --osscan-guess: Agresívnejšie hádanie pri OS detekcii

Časovanie a výkon:

- Všetky parametre majúce možnosť <time> sú v milisekundách, ak nie je špecifikované 's' (sekundy), 'm' (minúty), alebo 'h' (hodiny)
- -T<0-5>: Nastavuje časovací vzor (vyššie číslo znamená vyššiu rýchlosť)
- --min-hostgroup/max-hostgroup <size>: počet paralelne scanovaných cieľov
- --min-parallelism/max-parallelism <time>: paralelizácia požiadaviek na jeden cieľ
- --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: určuje čas čakania na odpoveď.
- --max-retries <tries>: určuje maximálny počet opakovaní zasielania testovacích paketov.
- --host-timeout <time>: určuje timeout pre konkrétny cieľ
- --scan-delay/--max-scan-delay <time>: časový rozostup medzi testami
- --min-rate <number>: zasiela minimálne <number> testovacích paketov za sekundu
- --max-rate <number>: zasiela maximálne <number> testovacích paektov za sekundu

Únik pred firewallmi a spoofing:

- -f; --mtu <val>: fragmentuje pakety (prípadne podľa zadanej veľkosti MTU)
- -D <decoy1,decoy2[,ME],...>: maskuje scanovanie návnadami
- -S <IP_Address>: predstiera falošnú zdrojovú IP adresu
- -e <iface>: použije špecifikované rozhranie na sieťovú komunikáciu
- -g/--source-port <portnum>: zasiela testovacie pakety zo zadaného portu
- --data-length <num>: pridá do zasielaných paketov náhodné dáta, prípadne v zadanej dĺžke
- --ip-options <options>: zasielané pakety budú mať v IP hlavičke nastavené zadané možnosti
- --ttl <val>: nastaví IP TTL (time to live) políčko
- --spoof-mac <mac address/prefix/vendor name>: predstiera zdrojovú MAC adresu
- --badsum: zasielané pakety budú mať vymyslený TCP/UDP kontrolný súčet

Výstup:

- -oN/-oX/-oS/-oG <file>: výstup scanovania v normálnom, XML, s|<rIpt kIddi3, a grepovateľnom formáte, v tomto poradí, do určeného súboru
- -oA <basename>: výstup v troch hlavných formátoch naraz
- -v: zvýši stupeň podrobnosti výpisu (dá sa použiť opakovane pre zvýšenie efektu)
- -d[level]: nastaví stupeň debugovania (možné až do 9)
- --reason: zobrazí dôvod, prečo je danému protu priradený daný stav
- --open: zobrazí iba otvorené (alebo pravdepodobne otvorené) porty
- --packet-trace: zobrazí všetky odoslané a prijaté pakety
- --iflist: vypíše rozhrania a routovaciu tabuľku hostiteľského OS (pre debugovanie)
- --log-errors: zapíše chyby a varovania do textového súboru
- --append-output: pridá nový výpis na koniec súborov namiesto ich prepísania
- --resume <filename>: obnoví prerušené scanovanie
- --stylesheet <path/URL>: XSL stylesheet pre prevod XML výstupu na HTML
- --webxml: referenčný stylesheet z Nmap.Org pre ľahšie portovateľné XML
- --no-stylesheet: zabráni priradeniu XSL stylesheet-u k XML výstupu

Ostatné:

- -6: Zapína IPv6 scanovanie
- -A: Zapína detekciu OS a služieb, skriptovací engine a trasovanie cesty
- --datadir <dirname>: Nastavuje umiestnenie dátových súborov Nmap-u
- --send-eth/--send-ip: Zasielanie pomocou priamych Ethernet rámcov alebo IP paketov
- --privileged: predpokladá, že užívateľ má všetky oprávnenia
- --unprivileged: predpokladá, že užívateľ nemá práva na priamu manipuláciu so socketmi
- -V: Vypíše číslo verzie programu
- -h: Zobrazí pomocníka s popisom najčastejších prepínačov

4. Program na detekciu zraniteľných počítačov v sieti

4.1. Požiadavky na scanovaciu aplikáciu

4.1.1. Dynamický výpočet oneskorenia

Aby scanovanie netrvalo príliš dlho a aby sa zabránilo vypršaniu limitu pred prijatím odpovede, je potrebné stanoviť vhodný čas oneskorenia. Toto je nutné spraviť automaticky bez zásahu používateľa a takisto je potrebné počas scanovania túto dobu automaticky upravovať, pretože systémy sa chovajú inak keď sú zaplavované požiadavkami. Najjednoduchšia technika je použiť čas získaný z ICMP echo scanovania (ping-u, popísané v kapitole 1.9).

4.1.2. Opakovanie prenosu

Vyslanie všetkých skúšobných paketov a následné čakanie na odpovede môže viesť k nepresnostiam ak boli pakety cestou zahodené. Priestor pre výraznúchybu je napríklad pri UDP alebo FIN scanovaní, kde sa hľadajú porty ktoré neodpovedajú. Preto je potrebné použiť niekoľko opakovaní skúšobných paketov pre porty, od ktorých nebola prijatá odpoveď.

4.1.3. Paralelizácia scanovania

Lineárne (sériové) scanovanie portov je možné použiť maximálne pre protokol TCP a len v rýchlej lokálnej sieti. V iných prípadoch je výkon takéhoto riešenia veľmi nízky. Preto je vhodné vo všetkých prípadoch použiť neblokujúce I/O operácie a scanovanie paralelizovať. Nárast výkonu je veľmi výrazný pre pomalšie siete, pri veľkom počte paralelných spojení na rýchlej sieti ale môže dôjsť dokonca k degradácii výkonu. Takisto ak je scanovaných niekoľko počítačov naraz, je vhodné scanovanie paralelizovať. Toto je značne priaznivé riešenie, keďže na rýchlych sieťach nedochádza k zníženiu výkonu ako v prípade scanovania jedného počítača.

4.1.4. Detekcia neexistujúcich systémov

Scanovanie všetkých portov neexistujúceho systému zaberie veľa času a neprinesie žiadne výsledky. Preto je vhodné kvôli zrýchleniu detekovať neexistujúce systémy a tie

následne nescanovať. Vhodnou možnosťou je detekcia odpovede na ICMP echo, známy ping. Niektoré systémy však na ICMP echo neodpovedajú a preto je nutné implementovať aj inú metódu, napríklad pomocou scanovania niektorých často používaných portov. Toto však tiež nezaručuje úplnú presnosť. Preto je vhodné implementovať aj možnosť detekciu neexistujúcich systémov vypnúť.

4.1.5. Flexibilná špecifikácia rozsahu

Podobne ako v predchádzajúcom prípade, nie vždy je potrebné scanovať celú sieť, resp. všetky počítače v zozname. Pre tieto prípady je vhodné umožniť scanovanie vybraného systému či skupiny systémov. Prípadne je žiadúce vytvoriť možnosť výberu skupiny podľa definovateľných kritérií.

4.1.6. Flexibilná špecifikácia portu

Niekedy je potrebné rýchlo zoscanovať len niekoľko dôležitých portov a z časových dôvodov je scanovanie všetkých 65535 portov nevhodné. Takisto nie je vhodné scanovať len určitý počet dolných portov. Preto je vhodné vytvoriť zoznam najdôležitejších portov, ktorý by sa v podobných situáciách použil.

4.1.7. Automatická detekcia vlastnej IP adresy

Najjednoduchším spôsobom je detekcia pomocou ICMP echo, keď sa použije adresa na ktorej sa odpoveď na ICMP echo request prijme. Taktiež je možné pokúsiť sa o detekciu primárneho sieťového rozhrania systému a použiť jeho IP adresu.

4.2. Užívateľské rozhranie

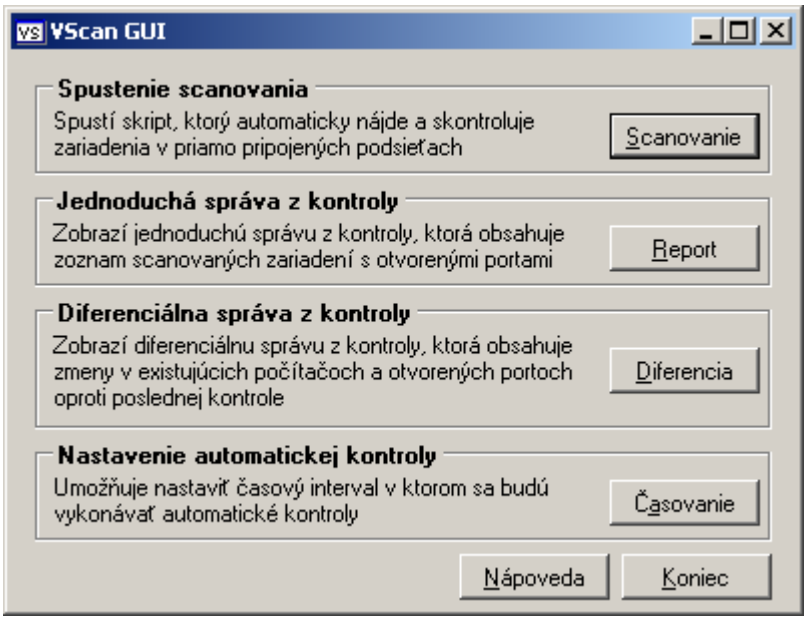
Užívateľské rozhranie programu musí byť navrhnuté tak, aby umožňovalo cieľovej skupine užívateľov jednoduchú a pohodlnú prácu. Keďže program je určný pre administrátorov, musí umožňovať automatizáciu pomocou skriptovania. Pokročilý užívateľ bude pravdepodobne používať najmä veriantu pre príkazový riadok, ale pre jednoduché a prívetivé ovládanie pre začínajúcich užívateľov je ale vhodné implementovať aj grafické užívateľské rozhranie.

4.2.1. Ovládanie z príkazového riadku

Keďže sa program skladá zo skupiny skriptov, je ovládanie z príkazového riadku a začlenenie do vlastných skriptov jednoduhé a rýchle. Jednotlivé skripty majú niekoľko parametrov. Ich zoznam a jednoduchý popis je možné získať spustením skriptu bez parametrov. Podrobnejší popis parametrov je v nápovede programu. Niektoré zo skriptov sú využívané len interne a nie sú určené na použitie užívateľom. Tieto skripty na túto skutočnosť pri svojom spustení upozorňujú.

4.2.2. Grafické užívateľské rozhranie

Grafické užívateľské rozhranie (zobrazené na Obr. 4.1) má uplatnenie v prípade, že administrátor chce rýchlo vykonať rozdielový bezpečnostný audit. GUI umožňuje jednoducho spustiť hľadanie nových počítačov a nových otvorených portov na existujúcej mape siete. Takisto umožňuje zobraziť výsledky scanovania v jednoduchej a prehľadnej forme. S pomocou grafického užívateľského rozhrania si môže administrátor udržiavať prehľad o dianí a zmenách v sieti. Grafické uživateľské rozhranie umožňuje takisto naplánovanie časovaných kontrol, ktoré sa následne automaticky vykonávajú bez vonkajšieho zásahu. Administrátor potom môže kontrolovať automaticky generované správy z kontrol a príslušne reagovať na vzniknuté situácie.



Obr. 4.1: Grafické užívateľské rozhranie – VScan GUI

4.3. Inštalácia

Inštalácia programu je vykonaná inštalačným skriptom. Počas inštalácie sa okrem programových súborov a podporných programov používaných v skriptoch (nmap, nessus, grep, diff) nainštalujú aj doplňujúce služby potrebné pre beh programu (WinPcap, runtime balíčky). Na systémoch Windows s podporou UAC sú vyžadované doplňujúce kroky, aby bola zachovaná úplná funkcionalita programu.

4.4. Textové filtre

Vytvorené skripty sú založené na filtrovaní výstupu rôznych programov. Filter je program, ktorý číta dáta zo štandardného vstupu (stdin), tieto dáta istým spôsobom spravováva a následne zapisuje na štandardný výstup (stdout). Spôsob spracovania závisí na povahe filtra. Práca so štandardným vstupom a výstupom je dôležitá kvôli zreťazeniu a presmerovaniu, ako sa textové filtre najčastejšie využívajú. Niektoré filtre ale čítajú vstup zo súborov a výstup môže byť takisto riešený týmto spôsobom. Okrem bežných filtrov bolo nutné vytvoriť niekoľko vlastných filtrov s požadovanými vlastnosťami.

Dôležité a často používané textové filtre:

- **grep.exe** – program na hľadanie vzoriek definovaných regulárnymi výrazmi v súboroch. Je to veľmi mocný nástroj umožňujúci veľmi pokročilé konštrukcie a tým komplexné nastavenia hľadania. Program má množstvo prepínačov, ktoré upravujú jeho činnosť;
- **find.exe** – program s podobným použitím a účelom ako „grep.exe“. Je to zjednodušená varianta „grep.exe“ pre OS MS Windows, ktorá neobsahuje také možnosti a celkovo je jej funkcionalita zjednodušená. Program find.exe nepodporuje vyhľadávanie podľa regulárnych výrazov, ale len podľa jednoduchých reťazcov znakov;
- **diff.exe** – program na porovnávanie súborov. Vstup číta z dvoch súborov a na štandardný výstup zapíše rozdielne riadky v jednotlivých súboroch. Existuje aj verzia „diff3.exe“, ktorá porovnáva údaje z troch súborov a „diff.exe“ umožňuje aj binárne porovnávanie súborov. Tento filter umožňuje rôzne formátovanie výstupu a na systémoch Unix, Linux a BSD sa v spojení s programom „patch.exe“ často používa na vytvorenie záplat pre zdrojové kódy programu a teda diferenciálne aktualizácie (program „patch.exe“ slúži na aplikovanie zmien z diferenčného súboru vygenerovaného pomocou „diff.exe“);

Vlastné filtre:

- **liner.exe** – textový filter, ktorý načíta dáta zo štandardného vstupu a zapíše ich na štandardný výstup tak, že každý záznam je v novom riadku. Táto funkcionalita umožňuje nahradenie programu „grep.exe“ jednoduchším programom „find.exe“;
- **one2many.exe** – číta dáta zo štandardného vstupu a jendotlivé záznamy zapíše každý zvlášť do vlastného súboru. Tieto sú číslované od nuly po 9999. Tento filter bol nutný v skorších verziách programu pre umožnenie paralelného scanovania rôznych zariadení, neskôr bol nahradený iným postupom, avšak stále má svoje uplatnenie.
- **htmlpack.exe** – filter pre upravenie HTML formátovania výstupu pre správy z kontroly. Odstraňuje nadbytočné znaky a tak robí výsledný kód čitateľnejší a zároveň zmenšuje veľkosť výsledného súboru;

4.5. Vytvorené skripty a programy

Vytvorené sada skripty pozostáva z niekoľkých oddelených dávkových súborov, ktoré sú ale navzájom prepojené. Každý z týchto skriptov má niekoľko prepínačov, ktoré umožňujú pozmeniť jeho správanie.

4.5.1. vscangui.exe

Použitie: vscangui.exe

- Grafické užívateľské prostredie programu. Umožňuje jednoduchú obsluhu skriptov. Jeho možnosti siahajú na spustenie automatickej kontroly, zobrazenie jednoduchej alebo diferenciálnej správy z kontroly a nastavenie automatického opakovania kontroly v určených časových intervaloch;
- Nepokrýva celú funkcionalitu vytvorených skriptov;

4.5.2. !vscan.cmd

Použitie: !vscan.cmd

- Hlavný skript vytvorenej sady. Slúži na automatickú detekciu všetkých požadovaných parametrov pre scanovanie priamo pripojených podsietí a automatické vytvorenie správ z kontroly.
- Má niekoľko parametrov, ale tieto nie sú určené na užívateľské použitie, len na volanie z programu „vscangui.exe“

4.5.3. getip.cmd

Použitie: getip.cmd výstupný_súbor

- Slúži na automatickú detekciu IP adries všetkých sieťových adaptérov (aj virtuálnych) v systéme;
- Parameter „výstupný_súbor“ je meno súboru (umožňuje úplnú špecifikáciu cesty) kam majú byť zistené IP adresy zapísané;

4.5.4. report.cmd

Použitie: len pri volaní z iného skriptu

- Rozhranie pre generovanie správ z kontrol, pre vlastné generovanie správ volá konkrétne skripty z podadresára „repgen“;
- Nie je určený na volanie užívateľom, je volaný automaticky z ostatných skriptov keď je potrebné vygenerovať správu z kontroly;
- Má niekoľko paramtervo určujúcich výstupný súbor, jazyk a stupeň zahrnutých detailov generovanej správy;

4.5.5. scanone.cmd

Použitie: scanone.cmd id_zariadenia id_výstupu

- Skript určený pre paralelné scanovanie detekovaných zariadení v priamo pripojených podsieťach;
- Parameter „id_výstupu“ určuje označenie súborov s výstupmi jednotlivých kontrol (určuje meno súboru, jednotlivé výstupy sú odlíšenie príponou);
- Parameter „id_zariadenia“ určuje vnútornú reprezentáciu zariadenia, ktoré má byť scanované touto inštanciou skriptu;
- Použitie tohto skriptu je náročné a zložité a preto je vhodnejšie použiť skript „scanip.cmd“, ktorý má jednoduchšie ovládanie a poskytuje všetky možnosti tohto skriptu;

4.5.6. scanip.cmd

Použitie: scanip.cmd IP_adresa

- Skript pre manuálne scanovanie jedného zariadenia identifikovaného IP adresou;
- Tento skript je užívateľským rozhraním pre „scanone.cmd“;

- Parameter „IP_adresa“ určuje IP adresu zariadenia určeného pre kontrolu, táto IP adresa nemusí patriť do niektorej z priamo pripojených podsietí, ale cesta k podsieti, kam patrí musí byť uvedená v routovacej tabuľke OS scanovateľa;

4.5.7. setup.cmd

Použitie: automaticky volaný pri inštalácii

- Tento skript je použitý pre nastavenie prostredia a inštaláciu podporných programov;
- Nie je určený pre volanie užívateľom, je volaný automaticky pri inštalácii;

4.5.8. timescan.cmd

Použitie: timescan.cmd

- Tento interaktívny skript slúži na nastavenie automatickej kontroly zariadení v priamo pripojených podsieťach;
- Má niekoľko parametrov, ktoré sú používané ostatnými skriptami a nie sú určené na použitie užívateľom;

4.5.9. uninstal.cmd

Použitie: uninstal.cmd [help]

- Použitie pre odinštalovanie programu a všetkých podporných balíčkov;
- Má niekoľko ďaľších parametrov, pre ich zoznam a vysvetlenie je potrebné použiť parameter „help“;

4.5.10. update.cmd

Použitie: update.cmd [help]

- Určený pre automatickú aktualizáciu programu na najnovšiu verziu z aktualizačného servera;
- Vyžaduje pripojenie k sieti Internet (z pochopiteľných dôvodov);
- Je možné urobiť aj manuálnu aktualizáciu preinštalovaním balíčkom s novšou verziou;
- Má niekoľko ďaľších parametrov, pre ich zoznam a vysvetlenie je potrebné použiť parameter „help“;

Záver

Práca mala za úlohu pripraviť teoretické pozadie pre vývoj aplikácie určenej na detekciu zraniteľných počítačov v sieti. Bola vykonaná analýza niekoľkých metód detekcie, najmä scanovaním portov a spracovávaním prijatých odpovedí. Takisto bola analyzovaná situácia v bezpečnosti počítačových systémov a rozpracovaný spôsob detekcie najčastejších bezpečnostných dier. Pre ochranu pred týmito bezpečnostnými dierami je zakaždým uvedených niekoľko jednoduchých rád, ktoré umožnia zabrániť počítačovému útoku alebo aspoň zmierniť jeho následky. Taktiež sú tu popísané bežné programy používané pri hodnotení počítačovej bezpečnosti a pri každom sú zhodnotené vhodné vlastnosti. Programy sú vyberané tak, aby správcovi zjednodušili prácu v niektorej alebo viacerých oblastiach správy bezpečnosti. Jedná sa predovšetkým o analýzu stavu siete, detekciu chýb a zraniteľností v počítačoch, sieťovej infraštruktúre a vo webových aplikáciách, ale aj o aplikácie na sledovanie zmien v sieti. V poslednej časti sú stanovené isté základné požiadavky na pripravovanú aplikáciu pre bezpečnostný audit a správu zabezpečenia a diskutuje vhodné vlastnosti tejto aplikácie z hľadiska funkcionality a pridanej hodnoty, aby mohla čo najlepším spôsobom slúžiť svojmu účelu.

Ďalej je navrhnutý a vypracovaný vlastný program, ktorý zistené poznatky v praxi uplatní, čo je konečným cieľom práce. Ako najvýhodnejšia možnosť bola vybraté zostavenie sady skriptov nad bežne dostupnými nástrojmi pre kontrolu zabezpečenia, konkrétne programu „Nmap“. V oblasti vytvorenia databázy pre detekciu bezpečnostných dier a zraniteľností je ešte potrebné vykonať nejakú prácu, pretože táto databáza musí mať značný rozsah aby dokázala plniť svoj účel. K skriptom je vypracované grafické užívateľské rozhranie, ktoré umožňuje zjednodušenie práce v niektorých často používaných úkonoch. Grafické užívateľské rozhranie nepokrýva všetky možnosti vytvorených skriptov, pretože ťažisko programu je v dávkovom spracovaní a automatizácii. Skripty sú stavané tak, aby maximálne zjednodušovali pridávanie a rozširovanie funkcionality, bez väčších ťažkostí je program možné rýchlo rozširovať.

V práci by bolo možné pokračovať najmä vybudovaním rozsiahlej databázy pre detekciu bezpečnostných dier a zraniteľností. Takisto by bolo vhodné rozšíriť grafické užívateľské rozhranie, aby pokrývalo širšie spektrum možností skriptov. Samozrejme tu je aj pridávanie novej a rozširovanie existujúcej funkcionality skriptov.

Použitá literatúra

[1] JOHNS, M. St.. *Request for Comments 1413 : Identification Protocol* [online]. February 1993. Dostupný z WWW: http://www.ietf.org/rfc/rfc1413.txt

[2] POSTEL, J., REYNOLDS, J.. *Request for Comments 959 : FILE TRANSFER PROTOCOL (FTP)* [online]. October 1985. Dostupný z WWW: http://www.ietf.org/rfc/rfc959.txt

[3] BAKER, F.. *Request for Comments 1812 : Requirements for IP Version 4 Routers* [online]. June 1995. Dostupný z WWW: http://www.ietf.org/rfc/rfc1812.txt

[4] NORTHCUTT, Stephen, et al. *Inside Network Perimeter Security*. 2nd edition. [s.l.] : [s.n.], 2005. 768 s.

[5] Microsoft Corporation. *Microsoft Security Advisories* [online]. December 19, 2007. Dostupný z WWW: http://www.microsoft.com/technet/security/advisory/default.mspx

Zoznam použitých skratiek

ACK	acknowledgment
ASP	active server pages (skriptovací systém pre webové stránky a aplikácie od MS)
BHO	browser helper object
BOOTP	bootstrap protocol
BSD	Berkeley software distribution (varianta OS UNIX)
BTS	base transciever station (vysielač v GSM sieťach)
CIFS	common internet file system
CMS	content management system
CSD	circuit switched data (prepínanie okruhov)
CSRF	cross-site request forgery
DHCP	dynamic host configuration protocol (protokol na automatickú IP konfiguráciu zariadení)
DNS	domain name system
DOM	document object model
DoS	denial of service (útok na webovú službu alebo sieťovú infraštruktúru spôsobujúci nedostupnosť služby)
FIN	finish
FTP	file transport protocol
GPL	general public license
GPRS	general packet radio service (paketový prenos v GSM sieťach)
GSM	global system for mobile communications
GNU	GNU's not Unix
HTTP	hyper text transport protocol
I/O	input/output
ICMP	internet control message protocol
IDS	intrusion detection system
IE	Internet Explorer (Microsoft Internet Explorer)
IP	internet protocol
IPS	intrusion prevention system
J2EE	Java 2 Platform, Enterprise Edition
LAN	local area network (lokálna sieť)

MAC	medium access control
MIME	multipurpose internet mail extensions
MS	Microsoft
.NET	knižnica funkcií a virtuálny stroj pre ich vykonávanie od MS, určená na zjednodušenie a unifikovanie programovania aplikácií pre OS Windows
NIC	network interface card (sieťová karta)
NNTP	network news transfer protocol
NT	new technology (Windows NT)
NTLM	NT lan manager
OS	operating system
PHP	professional home pages
RFC	request for comments
RPC	remote procedure call
RSS	resource description framework site summary
RST	reset
RTP	realtime transport protocol
SIP	session initiation protocol (protokol pre kontrolu spojenia vo VoIP)
SMS	short message service
SOHO	small office home office (malá firemná alebo domáca kancelária)
SP	service pack
SQL	structured query language (jazyk pre dotazy v relačných databázach)
SSH	secure shell
SYN	synchronize
TCP	transport control protocol
UAC	user access control
UDP	user datagram protocol
VoIP	voice over IP (prenos hlasu v paketových IP sieťach)
WLAN	wireless LAN (bezdrôtová lokálna sieť)
WVS	web vulnerability scanner (Acunetix Web Vulnerability Scanner)
XSS	cross-site scripting

Obsah priloženého CD

- elektronická verzia práce vo formáte .PDF
- elektronická verzia licenčnej zmluvy
- vypracované programy a skripty
- nápoveda k programu
- zdrojové kódy k vypracovaným programom
- inštalačné balíčky podporných programov