VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁSTROJ PRO ANALÝZU ZABEZPEČENÍ BEZDRÁTOVÝCH SÍTÍ

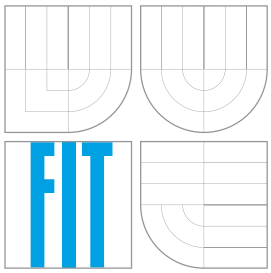
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB ŠENOVSKÝ

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁSTROJ PRO ANALÝZU ZABEZPEČENÍ BEZDRÁTOVÝCH SÍTÍ

TOOL FOR THE ANALYSIS OF NETWORK WIRELESS SECURITY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAKUB ŠENOVSKÝ
AUTHOR

VEDOUCÍ PRÁCE — Ing. LUKÁŠ ARON
SUPERVISOR

BRNO 2014

Abstrakt

Tato práce se zabývá bezpečností bezdrátových sítí a počítačovými útoky. Obsahuje popis různých zabezpečení sítí, nejznámějších útoků a nástrojů, kterými jsou útoky proveditelné. Samotná praktická část se zabývá vytvořením aplikací provádějící útoky SYN Flooding, Man in the Middle a IP Spoofing. Součástí praktické práce je i prezentace dosažených výsledků. Cílová platforma je operační systém Linux.

Abstract

This thesis deals with security of wireless networks and computer attacks. You can find description of various securities networks, most common attacks and tools for conducting these attacks. The goal of practical part is to create tools conducting SYN Flooding, Man in the Middle and IP Spoofing attack. The presentation of results is also included. The application is intended for Linux operating system.

Klíčová slova

Bezdrátové sítě, zabezpečení, útoky, testování, nástroje.

Keywords

Wireless network, security, attacks, testing, tools.

Citace

Jakub Šenovský: Nástroj pro analýzu zabezpečení
bezdrátových sítí, bakalářská práce, Brno, FIT VUT v Brně, 2014

Nástroj pro analýzu zabezpečení bezdrátových sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Lukáše Arona. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jakub Šenovský
19. května 2014

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu Ing. Lukášovi Aronovi za odborné vedení a rady, které mi při řešení práce poskytl. Dále bych chtěl poděkovat rodině a přítelkyni za podporu při studiu.

Obsah

Kapitola 1

Úvod

Bezdrátové sítě se v dnešní době staly nedílnou součástí našeho života. Člověk si je navykl používat na nejrůznějších místech jako jsou kavárny, vlaky či autobusy a jejich přítomnost v nich považuje za samozřejmost. Jejich velký rozmach, zejména v posledních letech, je zapříčiněn nízkými pořizovacími náklady, jejich mobilitou a hlavně také kvůli usnadnění práce. V minulosti se bezdrátové sítě využívaly výhradně pro hlasovou komunikaci. Dnes nabízí mnoho možností využití jako je komunikace, přenos dat či propojení nejrůznějších zařízení.

Jednou z jejich důležitou oblastí je jejich bezpečnost. Z počátku přetrvávala ze strany uživatelů nedůvěra v jejich zabezpečení. Jakmile se začaly objevovat lepší a modernější bezpečností prvky, důvěra v nich rostla a byly pořizovány stále větším množstvím uživatelů do zařízení či budov.

Špatné zabezpečení bylo v historii příčinou mnoha velkých finančních škod. Jakmile útočník pronikl do sítě přes špatné nastavení bezpečnostních prvků, nic mu nebránilo, aby odcizil nejrůznější soukromé dokumenty či přístupové údaje. Proto je na zabezpečení kladen tak velký důraz a jsou na něj vynakládány nemalé finanční prostředky.

Tato práce se v kapitole 2 zabývá jednotlivými druhy zabezpečení bezdrátových sítí. Jsou zde představeny a popsány principy, které využívají k ochraně, útoky, které jsou na ně známé a zda již bylo dané zabezpečení prolomeno. V kapitole 3 jsou představeny známé útoky prováděné v bezdrátových sítích typu IP Spoofing, Man in the Middle, SYN Flooding a další. Charakterizovány jsou zde chyby, které tyto útoky využívají a také možnosti, jak se proti nim bránit. Nástroje, které je umožňují provést, jsou popsány a porovnány v kapitole 4. Praktická část práce se zabývá vytvořením vlastní aplikace provádějící výše zmíněné útoky. V kapitole 5 je nastíněn návrh, implementace, testování vytvořených aplikací a v neposlední řadě jsou zde dosažené výsledky zhodnoceny.

Kapitola 2

Zabezpečení bezdrátových sítí

V dnešní době je mnoho možností, jak zabezpečit bezdrátové sítě. Vývoj bezpečnostních prvků jde neustále dopředu a stále vznikají nové technologie. Taktéž dochází k objevování chyb již dříve představených bezpečnostních prvků, které dokážou zabezpečení sítě ochromit a umožnit proniknutí do sítě neoprávněným uživatelům.

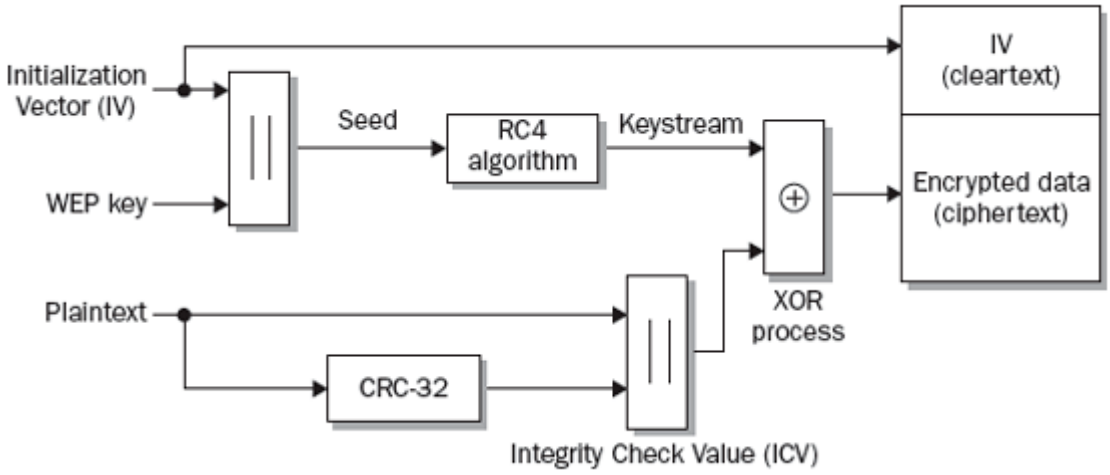
2.1 Wired Equivalent Privacy

Bezpečnostní protokol Wired Equivalent Privacy (WEP) je součástí standardu 802.11 jako volitelný doplněk. Hlavním požadavkem při jeho vytvoření bylo poskytnutí srovnatelného zabezpečení jako u drátových sítí. Proto byly stanoveny hlavní tři cíle činnosti protokolu. Prvním cílem bylo zajištění důvěrnosti osobních údajů, kdy jsou data před odesláním zašifrovaná. Druhým cílem bylo řízení přístupu k síti, kdy se uživatel autentizuje vůči přístupovému bodu statickým sdíleným WEP klíčem. Posledním cílem bylo zajištění integrity přenášených dat, kdy je před jejich zašifrováním spočítán integritní součet (Integrity Check Value) sloužící pro zamezení s jejich manipulací během přenosu. [6]

Šifrování

Šifrování dat je prováděno na datové (L2) vrstvě pomocí symetrické RC4 šifry. Tato šifra může být buď 64 nebo 128 bitová. Skládá se ze statického inicializačního vektoru (IV) dlouhého 24 bitů a sdíleného uživatelského klíče dlouhého 40 nebo 104 bitů. Inicializační vektor je složen z prostého, 24 bitového, textu, který je pravidelně měněn a tudíž výsledná šifra je pro každý paket jedinečná. Kratší, 40 bitový, uživatelský klíč tvoří 10 znaků v hexadecimálním tvaru nebo 5 znaků v ASCII tvaru, delší klíč tvoří 26 znaku v hexadecimálním tvaru nebo 13 ASCII znaků.

Při šifrování je z přenášených dat vypočítán kontrolní součet (Cyclic Redundancy Check). Je přiřazen na konec přenášených dat a s jejich spojením vzniká Integrity Check Value (ICV). Jak už bylo výše zmíněno, ICV slouží k detekci chyb během přenosu a zamezení neoprávněné manipulaci s daty. Dále je IV spojen se statickým WEP klíčem a zašifrován pomocí RC4 šifry. Poslední fází šifrování je spojení výstupu z RC4 šifry s ICV pomocí logické operace XOR. Celý proces šifrování je zobrazen na obrázku 2.1. [7]



Obrázek 2.1: Průběh šifrování dat u zabezpečení WEP[1]

Bezpečnostní chyby

Protokol WEP obsahuje mnoho bezpečnostních chyb. Jednou z chyb je jednostranná autentizace, kdy zařízení uživatele nerozeznává, s jakým přístupovým bodem komunikuje. Zároveň se autentizuje pouze zařízení, nikoliv uživatel. Bezpečnostní trhlinu představuje taktéž autentizace sdíleným klíčem, kdy je ověřovací relace odesílana otevřeným způsobem a útočník může tuto relaci odchytit a následně z ní odvodit tajný klíč. Taktéž není přesně specifikováno, jakým způsobem by se měla měnit hodnota IV, proto v jeden okamžik možnosti IV dojdou a právě v tuto dobu je možné klíč odhalit. Toto zabezpečení nepodporuje dynamickou změnu klíče, proto je jeho distribuce prováděna manuálně, což představuje problém ve velkých bezdrátových sítích.

V roce 2001 byl zveřejněn postup na jeho prolomení, kdy útočník odposlouchává komunikaci v síti a následně je z odchycených paketů vypočítán šifrovací klíč.

Nástroje, jimiž lze prolomit toto zabezpečení, je Aircrack-ng představený v kapitole níže, WEPCrack nebo Autocrack. [21]

2.2 Wi-Fi Protected Access

Wi-Fi Protected Access (WPA) vychází z normy IEEE 802.1i. Jeho vytvoření bylo reakcí na odhalení chyb v bezpečnostním protokolu WEP. Nutnou podmínkou bylo zajistit kompatibilitu s hardwarovým zařízením podporující zabezpečení WEP s RC4 šifrou, proto bylo nutné umožnit změnu zabezpečení na WPA pouze softwarovou aktualizací stávajících zařízení. Protokol řeší problémy s autentizací zařízení, distribuci tajného klíče v sítích s mnoha uživateli a případnou obnovovu klíče. Tento způsob zabezpečení byl představen v roce 2002 ještě v rozpracovaném stavu, jelikož byly bezdrátové sítě díky prolomení WEP de facto nezabezpečené. Současně se začalo pracovat na jeho nástavbě označené jako WPA2.

[1] http://groups.csail.mit.edu/mac/classes/6.805/student-papers/spring02-papers/paranoia.htm

Šifrování

WPA využívá oproti WEP silnější šifrování pomocí Temporal Key Integrity Protocol (TKIP). Pro každý paket je vygenerován nový klíč, který je vytvořen ze základního klíče, MAC adresy příjemce a čísla paketu, které slouží pro vytvoření odlišných šifrovacích klíčů. Číslování paketů je využíváno k ochraně proti útoku opakováním (Replay attack). Pro zajištění integrity dat a hlavičky rámce je použit mechanismus Message Integrity Check (MIC), vytvořený algoritmem označeným jako Michael, který je následně umístěný mezi datovou částí a ICV. [9]

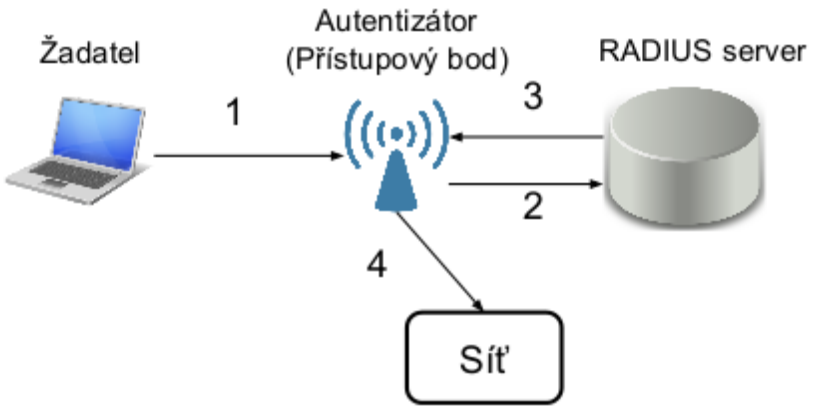
2.3 IEEE 802.11i

Tento bezpečnostní dodatek standardu IEEE 802.11, také označovaný jako WPA2, si kladl za cíl nahradit již prolomené předchozí zabezpečení WEP a WPA. K jeho schválení došlo 24. června 2004. Oproti předešlým metodám poskytuje lepší autentizace i šifrování jak zařízení, tak i dat. Jedním z nejvýznamnějších doplňků je nový protokol CCMP se silnou metodou šifrování AES. Šifrování TKIP zde bylo ponecháno kvůli možnému sloučení se zabezpečením WPA. Dalším významným rozšiřujícím doplňkem je předběžná autentizace, která umožňuje rychlou a bezpečnou komunikaci mezi přístupovými body.

Autentizace zařízení

Pro bezpečnou autentizaci a autorizaci zařízení je zde implementován protokol IEEE 802.1X. Skládá se ze tří důležitých částí. První částí je žadatel, neboli zařízení mající snahu se připojit k síti, další důležitou částí je přístupový bod plnící roli autentizátora a posledním prvkem je autorizační server označovaný jako RADIUS.

Při navázání spojení mezi žadatelem a přístupovým bodem jsou akceptovány pouze autentizační rámce přenášené protokolem EAP. Ostatní datový provoz je blokován. Žadatel odešle autentizační informace, které jsou autentizátorem přeposlány autorizačnímu serveru. Následně dochází k ověření žadatele a odeslání výsledku zpět k autentizátoru. Ten na základě výsledku uživatelovi datový provoz odblokuje či nikoliv. Tento popsaný princip je zobrazen na obrázku 2.3.1.



Obrázek 2.3.1: Průběh autorizace zařízení protokolem IEEE 802.1X

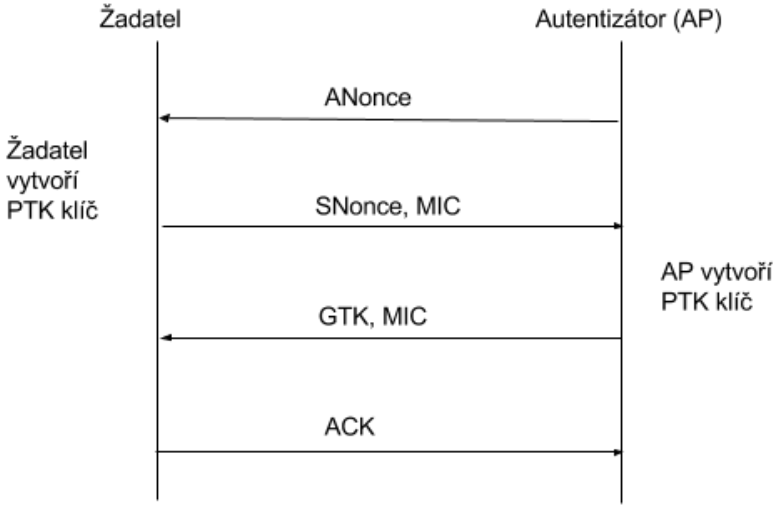
Jak už bylo zmíněno výše, volitelným doplňkem normy 802.11i je předběžná autentizace, která klientovi umožňuje autentizovat se u jiného přístupového bodu, aniž by byl v jeho dosahu. [13]

Šifrování

Bezpečné šifrování je zajištěno novým protokolem CCMP založeným na procesu AES. CCMP protokol zajišťuje ochranu přenášeným datům i hlavičkám. Tato bloková šifra používá 128 bitové klíče, které jsou, oproti zabezpečení WEP, dynamicky generované. Generování nových šifrovacích klíčů znemožňuje útočníkovi získát tento klíč, jelikož není schopen odchytnout větší množství paketů šifrovaných stejným klíčem. Pro zajištění integrity a pravosti zpráv je využívaný mechanismus MIC, pro detekci chyb slouží FCS součet. Protokol CCMP obsahuje taktéž prostředky pro zamezení útoku opakováním (Replay attack).

Distribuce klíčů

V normě IEEE 802.11i je distribuce klíčů zajištěna protokolem EAP definující dvě nové metody, a to *four-way handshake* (čtyřcestné ověřování) a *group key handshake* (skupinové ověřování). Čtyřcestné ověřování je odvozeno od počtu paketů, které si mezi sebou vyměňuje žadatel s přístupovým bodem při autorizaci. Tento princip ověřování je zobrazen na obrázku 2.3.2.



Obrázek 2.3.2: Průběh distribuce klíčů čtyřcestným ověřováním

První zpráva, odeslaná autentizátorem žadatelovi, obsahuje náhodné vygenerované číslo označené jako *ANonce*. Současně je žadatelovi, z informací obsažených ve zprávě, umožněn vypočítat párový dočasný klíč PTK. Následně žadatel vytvoří vlastní náhodné číslo *SNonce*. Autentizátorovi je odeslaná zpráva obsahující *SNonce* spolu s bezpečnostním parametrem a autentizačním součtem MIC vypočítaného pomocí potvrzovacího klíče KCK. Tento součet slouží k ověření, zda je odeslaná zpráva validní. Třetí zpráva obsahuje bezpečnostní parametr spolu se skupinovým klíčem GTK zašifrovaným pomocí šifrovacího klíče KEK. Součástí této zprávy je opět autentizační součet MIC. Čtvrtá zpráva potvrzuje, že byl párový dočasný klíč GTK doručen a nainstalován správně. [12]

Tento bezpečnostní prvek je označován jako bezpečný. Při použití slabého bezpečnostního hesla je ale prolomitelný tzv. slovníkovým útokem, kdy jsou postupně zkoušené všechny znakové kombinace. Porovnání jednotlivých vlastností již dříve představených bezpečnostních standardů WEP, WPA a WPA2 je zobrazeno v tabulce 2.1.

	WEP	**WPA**	**WPA2**
Šifra	RC4	RC4	RC4
Velikost klíče	40 bitů	128 bitů - šifrování 64 bitů - autentizace	128 bitů
Velikost IV	24 bitů	48 bitů	48 bitů
Integrita dat	CRC-32	Michael	CCM
Integrita hlaviček	–	Michael	CCM
Útok opakováním	–	IV sekvence	IV sekvence
Distribuce klíčů	–	EAP	EAP

Tabulka 2.1: Porovnání WEP, WPA a WPA2 [16]

2.4 Media Access Control filtering

Tento způsob zabezpečující přístup k síti využívá skutečnosti, že každá síťová karta zařízení má jedinečnou 48 bitovou adresu MAC, která je využívána k autorizaci. Aby se zařízení mohlo připojit do sítě, musí administrátor přidat danou adresu na seznam povolující přistup. Pokud zařízení žádá o přístup a není na seznamu, je mu přístup odepřen. Při použití tohoto způsobu zabezpečení je nutné udržovat aktuální seznam MAC adres zařízení mající povolený přístup. Problém nastává u velkých sítí s mnoha uživateli, kdy tento způsob může být velice náročný.

Media Access Control filtering neposkytuje spolehlivé zabezpečení sítě, jelikož lze softwarově měnit MAC adresu zařízení a útočník se může vydávat za uživatele s povoleným přístupem k síti.

2.5 Service Set Identifier hiding

Bezpečnostní metoda Service Set Identifier (SSID) hiding je založena na principu zakázání vysílání identifikátoru SSID všem bezdrátovým zařízením v okolí. K síti se může připojit pouze zařízení, které zná přesný název SSID. Tato metoda není bezpečná - jakmile se zařízení připojí k síti, vysílá název SSID v nezašifrovaném textu. Útočník může tento text zachytit, zjistit SSID a následně se bez větších problémů připojit k síti.

Kapitola 3

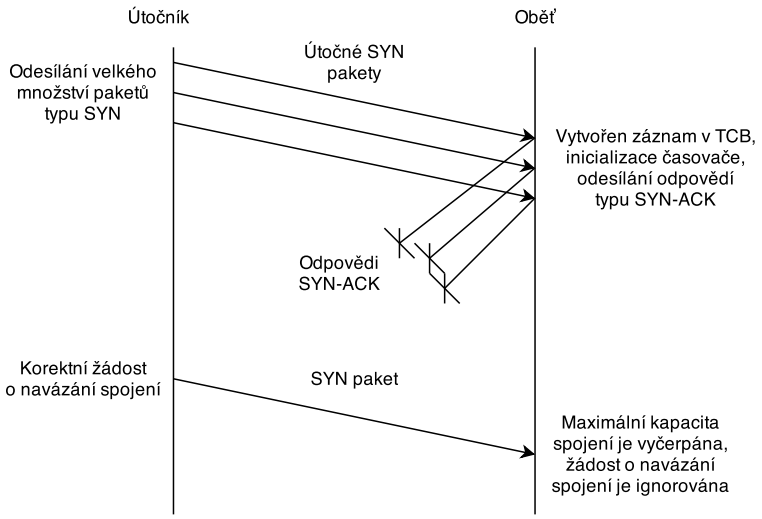
Útoky v bezdrátových sítích

V bezdrátových sítích existuje mnoho nejrůznějších typů útoků. Lze je rozdělit do následujících kategorií:

- **Pasivní útoky** - jejich principem je monitorování nezašifrované komunikace v síťovém provozu ve snaze odchytit hesla či důležité informace. Útočník nemodifikuje zachycená data, ale využívá je jako základ pro další útoky.
- **Aktivní útoky** - snaha proniknout skrz zabezpečení pomocí virů, červů či trojských koní. Útočník přímo vytváří či modifikuje data. Tento typ útoků je snáze odhalitelný oproti pasivnímu.
- **Distribuované útoky** - Tento typ útoku spočívá v rozesílání velkého množství požadavků na cílové servery ve snaze přesáhnout jejich maximální kapacitu současně zpracovaných požadavků a znemožnit jim řádný provoz. Velmi často je využíván tzv. botnet, který je vytvořen z nakažených počítačů a centrálně řízen útočníkem.
- **Phishing** - Při tomto útoku útočník vytváří falešné webové stránky či rozesílá podvodné elektronické zprávy ve snaze vylákat z uživatelů hesla k internetovým službám nebo kreditním kartám.

3.1 SYN Flooding

Tento útok lze zařadit mezi útoky typu Denial of Service (DOS) neboli odmítnutí služby. Útočník zasílá velké množství paketů typu SYN s požadavkem o navázání TCP spojení. Oběť útoku potvrdí přijetí paketu odesláním potvrzovacího paketu typu SYN+ACK. Následně operační systém oběti zabere příslušné paměťové zdroje v Transmission Control Block (TCB) pro nově vznikající spojení a označí ho jako poloviční (half-open). Tato poloviční spojení jsou přesunuta do velikostně omezené fronty, tzv. *backlog queue*, kde vyčkávají na přijetí potvrzovacího paketu ACK, který nikdy nedorazí. Taktéž se aktivuje časovač sloužící k odstranění polovičního spojení z fronty, který ovšem nikdy nevyprší, jelikož je nastavený na velkou časovou prodlevu. S rostoucím počtem útočníkových požadavků o nové spojení systém vytváří nová poloviční spojení, která jsou vkládána do fronty a dochází k jejímu úplnému obsazení. Po úplném vyčerpání kapacity fronty se může reakce na požadavek o nově příchozí spojení lišit. Prvním možných způsobem je odmítání dalších žádostí o spojení, kdy můžou být odmítnuty i korektní žádosti. Druhým způsobem je nahrazení posledního záznamu v backlog frontě nově příchozím požadavkem.



Obrázek 3.1.1: Ilustrace útoku SYN flooding

Při útoku je velmi často podvrhnutá zdrojová IP adresa útočníka k zabránění jeho odhalení. Dalším důležitým kritériem je, aby systém, mající podvrženou IP adresu, neodpovídal na odpovědi typu SYN+ACK či byl v nejlepším případě neaktivní. Pokud by byl aktivní, systém by odpověděl na nevyžádané potvrzení typu SYN+ACK paketem typu RST. Tento paket označuje zrušení spojení, tudíž by byla, po přijetí paketu RST, operačním systémem oběti uvolněna zabraná paměť v TCB bloku, odstraněn záznam v backlog frontě a následně by útok nebyl úspěšný. Pro dosažení maximálního účinku útoku je nutné sestavit seznam neaktivních IP adres a vybírat z něj pouze tyto IP adresy. [8]

Možnou ochranou proti výše popsanému útoku je možnost omezit počet nových spojení z určité zdrojové adresy, zvětšit velikost backlog fronty, zmenšit periodu čítače, odstranění nejstaršího polovičního spojení či použít technologii zvanou SYN cookies. U této technologie je po přijetí paketu SYN odeslána odpověď SYN+ACK s nově vygenerovanou číselnou sekvencí identifikující dané spojení. Jakmile je přijat od iniciátora spojení potvrzovací paket ACK, teprve pak jsou zabrány příslušné paměťové zdroje a vytvořené nové spojení. [10]

Nástroj umožňující provést tento útok je SynGUI 2.0 nebo Hping.

3.2 IP Spoofing

Internetový protokol (IP) se používá pro směrování datových paketů v síti. Součástí paketů je také IP hlavička, která v sobě zahrnuje IP adresu odesilatele, IP adresu příjemce a mnoho dalších informací. Cílem útoku je podvrhnutí adresy odesilatele cizí IP adresou a tím způsobem znemožnit identifikace skutečného odesilatele. Jelikož IP vrstva operačního systému standardně přidává IP adresu do hlavičky paketu, musí útočník tuto vrstvu obejít a vkládat podvrženou adresu přímo na úrovni síťového zařízení. Útočník odpověď na tento paket neobdrží, protože příjemce odpovídá na podvrženou IP adresu. Pokud je vložena jako adresa odesilatele adresa jiného počítače v síti, může útočník skenováním komunikace v síti odpověď odchytit a zobrazit si ji.

V praxi je velmi často tento útok využíván v kombinaci s ostatními typy útoků jako je například Ping of Dead, TearDrop nebo SYN Flooding.[1]

K ochraně proti IP Spoofing je možné využít filtrování zdrojových IP adres na routeru, kdy nejsou propuštěny adresy přicházející z vnější strany se zdrojovou adresou spadající do vnitřních sítí. [2]

Nástroje, jimiž lze provést tento útok, jsou Hping a SendIP.

3.3 Man in the Middle

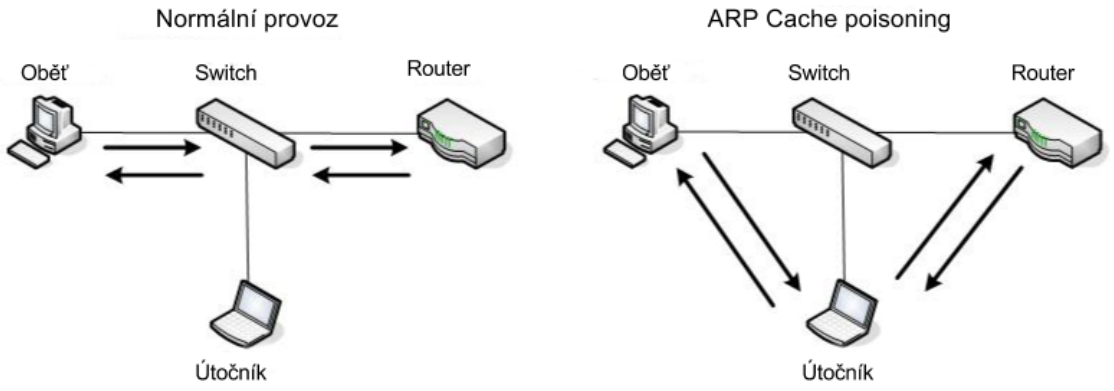
Jedná se o formu aktivního odposlechu dat v komunikaci mezi dvěma subjekty, kdy útočník mezi nimi vystupuje jako prostředník. Veškerá komunikace probíhá přes útočníka, což mu umožňuje zachytit, upravovat a přeposílat tyto data. Výhodou útočníka je, že nemusí být fyzicky přítomen na stejné síti jako cíl útoku, jelikož se dá komunikace snadno přesměrovat. Nyní zde budou představeny a popsány nejznámější verze tohoto typu útoku.

ARP Cache Poisoning

Jedná se o jeden z nejstarších typů útoku dovolující útočníkovi odposlouchávat komunikaci mezi obětmi na stejné podsíti. Jak název napovídá, je tento útok založen na Address Resolution Protocol (ARP), který je využíván pro překlad IP adres na MAC adresy a taktéž pro přímou komunikaci dvou zařízení v síti. Každé zařízení má v sobě implementovanou ARP Cache tabulku, která obsahuje záznamy o ostatních zařízeních v síti ve tvaru IP adresa - MAC adresa. Tyto záznamy můžou být staticky konfigurované nebo dynamicky se měnící. Aby bylo možné záznamy dynamicky měnit, nabízí tento protokol dva typy zpráv:

- **ARP požadavek** - slouží pro získání MAC adresy zařízení v lokální síti. Zjednoduše lze napsat, že tato zpráva obsahuje otázku ve tvaru „Kdo má tuto IP adresu? Odpověz!“.
- **ARP odpověď** - je odesílána na ARP požadavek a obsahuje MAC adresu zařízení. Po přijetí ARP odpovědi si zařízení odesílající ARP požadavek uloží obsah zprávy do ARP Cache tabulky.

Princip útoku spočívá v rozesílání falešných ARP odpovědí s cílem podvrhnout záznamy v ARP Cache tabulce zařízení. Jakmile jsou záznamy podvrhnuty, je veškerá komunikace mezi zařízeními v síti přesměrována přes útočníka, jak je možno vidět na obrázku 3.3.2.



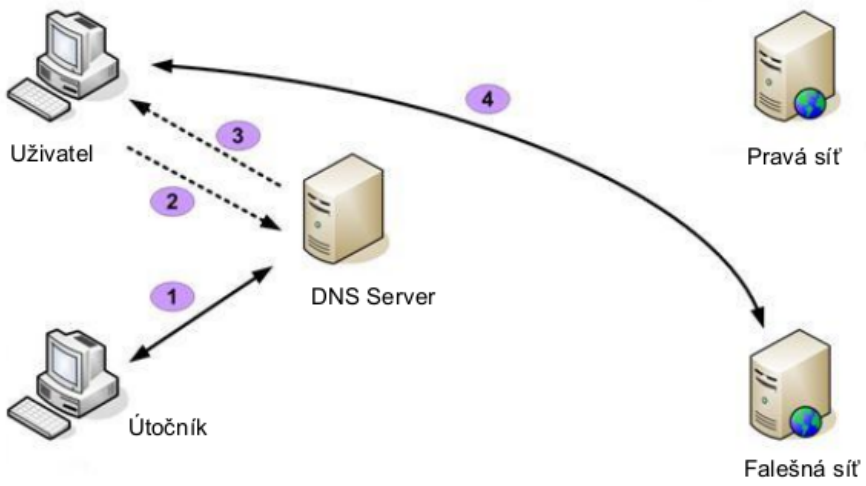
Obrázek 3.3.1: Ilustrace provozu na síti před útokem a po útoku ARP Cache Poisoning [1]

Existuje několik druhů obrany proti tomuto útoku. V první řadě se jedná o vytváření statických záznamů v ARP Cache tabulce, které nemohou být dynamicky přepsány. Další možností je certifikovat ARP odpovědi speciálním softwarem. Necertifikované odpovědi by byly blokovány nebo ignorovány. V neposlední řadě je obrana přímo v operačním systému, kdy jsou ARP odpovědi, bez předchozího odeslání požadavku, ignorovány. [11]

DNS Spoofing

Domain Name System (DNS) protokol je jeden z nejdůležitějších protokolů využívaných v internetu. Slouží pro překlad doménových jmen na IP adresy a naopak, překlad názvů počítačů na kanonická jména, určení poštovního serveru pro danou doménu nebo pro další činnosti. Systém DNS obsahuje veškeré IP adresy a jejich ekvivalenty v podobě doménových jmen. Je využíván většinou aplikačními protokoly jako jsou HTTP, FTP, UDP nebo SMTP.

Jakmile je zadáno do prohlížeče doménové jméno zařízení, resolver (program zajišťující překlad) odesílá požadavek s obsahem a unikátním identifikačním číslem na DNS server s žádostí o zjištění IP adresy zadané domény. Tento požadavek je odchycen útočníkem, identifikační číslo je vyjmuto a následně uloženo do falešně vytvořeného paketu, který je odeslán tazateli. Oběť útoku je následně připojena na podvrženou IP adresu.



Obrázek 3.3.2: Ilustrace útoku DNS Spoofing [2]

Zabezpečení, kterým lze zabránit útoku DNS Spoofing, je použití DNSSEC, který zajišťuje integritu a důvěryhodnost dat. Další možností je přesměrovat všechny DNS požadavky na lokální DNS server či blokovat požadavky z externích DNS serverů. [4]

Session Hijacking

Hlavním cílem útoku Session Hijacking je ukradnout oběti tzv. SESSION_ID (SID), které je vygenerováno při připojení oběti k požadovanému cílovému zařízeni, např. webovému serveru. SID je využívano jako autorizační heslo při relaci mezi uživatelem a serverem.

[1] http://www.windowsecurity.com/articles-tutorials/authentication_and_encryption/Understanding-Man-in-the-Middle-Attacks-ARP-Part1.html

[2] http://www.technicalinfo.net/papers/Pharming2.html

Činnost útočníka provádějící tento útok spočívá ve skenování komunikace mezi obětí a serverem. Jakmile jsou útočníkem odchyceny všechny požadované informace, nic mu nebrání připojit se k cílovému zařízení a vydávat se za oběť útoku.

Ochranou proti ukradení SID je používat bezpečné spojení HTTPS, používat dlouhé klíče obsahující náhodné znaky a číslice. Obranou na straně serveru je označit SID jako neplatné ihned po odhlášení uživatele.

3.4 Idle scan

Útok, známý také jako zombie scan, slouží ke skenování dostupných TCP portů. Tento útok využívá ke komunikaci nečinného prostředníka, tzv. zombie. Útočník odesílá oběti pakety typu SYN se zdrojovou IP adresou prostředníka a následně vyčkává odpověď na podvrženou IP adresu. Pokud prostředník obdrží odpověď typu SYN/ACK, je skenovaný port oběti otevřený. Pokud je obdržena odpověď typu RST, je port uzavřený. Tento útok využívá skutečnosti, že hodnota identifikačního pole v IP hlavičce je pro každý paket unikátní a zvyšuje se s každým odeslaným paketem o hodnotu jedna. Tyto přijaté pakety umožňují útočníkovi zkoumat měnící se hodnotu identifikačního pole v hlavičce a získat tak seznam otevřených či uzavřených portů. [18]

Mezi způsoby, jak se proti tomuto útoku bránit, je použití firewalů a přístupových filtrů.

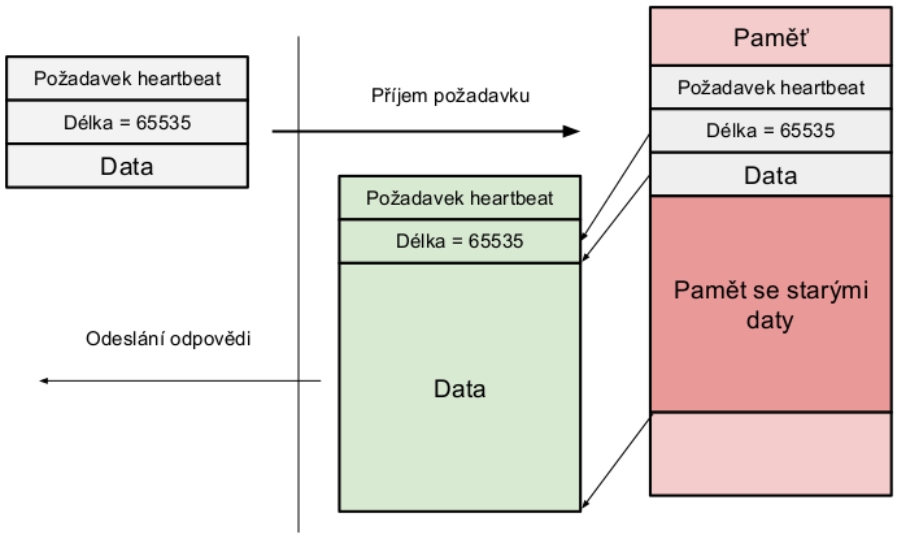
3.5 The Heartbleed Bug

The Heartbleed Bug (chyba krvácející srdce) je nedávno zveřejněná závažná bezpečnostní slabina populární šifrovací knihovny OpenSSL. Tato chyba umožňuje přístup k datům, které jsou za normálních podmínek chráněny protokolem TLS/SSL. Protokol TLS/SSL zajišťuje v internetové komunikaci bezpečnost a soukromí, a je využíván aplikacemi jako je email, různými komunikátory (IM), webovými prohlížeči a nebo virtuálními soukromými sítěmi (VPN).

Chyba využívá rozšíření protokolu TLS/DTLS nazvané *heartbeat*[3], jehož úkolem je zjišťovat, zda je protistrana při komunikace stále aktivní. Při tomto zjišťování je odeslán požadavek s polem dat, který musí protistrana odeslat beze změn zpět. Jelikož toto pole dat může být různě dlouhé, je jeho velikost určena dvoubajtovou hodnotou (maximální velikost je tedy 65535). Potíž v OpenSSL byla ta, že nedocházelo ke kontrole, zda délka dat v požadavku není delší než délka celé zprávy. Po přijetí požadavku byla zpráva uložena do paměti a následně vygenerovaná odpověď. Do odpovědi bylo překopírováno tolik dat, kolik měl deklarovanou velikost předem přijatý požadavek. Pokud byla velikost větší než skutečná data, byla do odpovědi přidána paměť následující za místem, kde se nacházel uložený požadavek. Ilustraci útoku je možné vidět na obrázku 3.5.1. V této části paměti jsou obvykle uložena data před či po zašifrování. Najdeme zde také uživatelská jména a hesla, X.509 certifikáty, emaily, důležité pracovní dokumenty, které mohl útočník snadno získat. Tato chyba se nevyskytuje pouze u serverů, ale taktéž i u zařízení uživatelů, tudíž je možné provést útok i opačným směrem, tedy ze serveru na zařízení uživatele. V praxi se ale tento opačný jev příliš nevyskytuje. [3]

Při představení této chyby bylo odhadováno, že bylo postiženo okolo 17% (500 000)

[3] https://tools.ietf.org/html/rfc6520



Obrázek 3.5.1: Ilustrace přijetí požadavku, jeho uložení a následně vytvoření odpovědi s přesahující velikostí dat požadavku

serverů certifikovaných ověřenou certifikační autoritou.[4] Michal Špaček, známý český bezpečnostní analytik, vytvořil projekt za účelem monitorovat počet zranitelných serverů na rozsahu českých IP adres a to konkrétně na portu 443 (HTTPS). Počet nezabezpečených hostů na těchto IP adresách byl i 27 dní (chyba představena 7. dubna 2014) od zveřejnění útoku celkově 3647. [20]

Opravená aktualizace knihovny OpenSSL byla vytvořena Adamem Langley a Bodem Moellerem.

[4] http://news.netcraft.com/archives/2014/04/08/half-a-million-widely-trusted-websites-vulnerable-to-heartbleed-bug.html

Kapitola 4

Nástroje pro testování zranitelnosti

Nástrojů pro testování zabezpečení systémů i analýzy sítí existuje celá řada. Lze je rozdělit na nástroje pro sledování provozu v síti, kdy jsou pakety jednotlivých protokolů pouze zachytávány a následně zobrazeny uživateli, nebo na nástroje pro vytváření a odesílání různých typů paketů. V obou případech jsou tyto aplikace využívány pro odhalení nežádoucího chování v síti.

4.1 Aircrack-ng

Jedná se o sadu nástrojů[1] sloužící k testování zabezpečení sítí. Tyto aplikace vznikly z odštěpení od původního projektu nazvaného Aircrack a v současné době jsou vyvíjené vývojáři pod vedením Thomasa d'Otreppa. Mezi důležité a často využívané utility, které v této sadě najdeme, jsou *aircrack-ng*, který poskytuje možnost prolomení zabezpečení typu WEP a WPA slovníkovým útokem. *Airdecap-ng* umožňuje pomocí získaného hesla dešifrovat odchycené soubory. K odchytávání provozu na síti a následnému exportu do pcap souboru slouží aplikace *airodump-ng*. Dále je zde obsažen nástroj *Aireplay-ng*, který je využíván pro vytváření falešných paketů.

Aircrack-ng je možné spouštět pod systémy Windows a Linux. Původní verze byly vytvořeny pouze pro platformu Linux, ale nyní lze najít verze i pro Windows, které ovšem nejsou tak funkčně obsáhlé. Poslední stabilní verze nese označení 1.1 a byla vydána v roce 2010. [5]

4.2 Wireshark

Wireshark je jedna z nejrozšířenějších aplikací pro analýzu síťového provozu. Jeho hlavním účelem je odchytávání paketů, jejich dekódování a zobrazení do snadno prezentované podoby. Tento open-source projekt je vývojáři aktivně vyvíjený a bezplatně dostupný.

Wireshark podporuje analýzu více jak 750 druhů protokolů, které mohou být získány aktivně ze sítě nebo načteny z uloženého souboru. Tato aplikace disponuje jednoduchým grafickým rozhraním, které bylo v dřívějších verzích tvořeno pomocí GTK+ widget toolkit. Současně vyvíjené verze už využívají knihovnu Qt. Další předností je možnost přepnutí

[1] http://www.aircrack-ng.org/

síťové karty do promiskuitního režimu, kdy je zaznamenávána veškerá komunikace v síti. Významnou výhodou této aplikace je její multiplatformost, kdy může být spouštěna na různých operačních systémech. Wireshark je dostupný taktéž v konzolové verzi označené pod názvem tShark.

Pro selekci jednotlivých typů paketů slouží filtry. Wireshark používá dva typy filtrů:

- **Zaznamenávající** - tento typ filtru se definuje při startu zachytávání paketů. Slouží pro specifikaci typů protokolů, které mají být odchytávány a následně zobrazeny uživatelovi.
- **Zobrazovací** - definuje se v hlavním okně programu při aktivním zachytávání či po jeho ukončení. Slouží pro selekci konkrétního typu protokolu či paketů, které jsou následně samostatně zobrazeny. Oba dva typy filtrů je možné mezi sebou kombinovat.

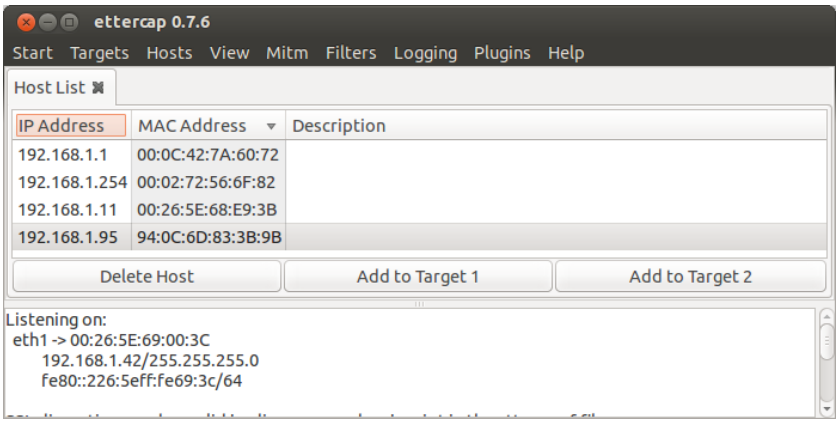
Mezi hlavní uživatele wiresharku patří správci sítí, síťoví vývojáři, programátoři, ale taktéž je využíván útočníky pro analýzu síťového provozu. [15]

4.3 Hping

Tato konzolová aplikace slouží pro analýzu TCP/IP protokolů a generování paketů. Může být spuštěna na operačních systémech Linux, Windows, OS X. Hlavním úkolem nástroje je testování firewalů, portů, sítí a poskytuje pokročilé trasování paketů. Taktéž umožňuje provést útok Idle scan popsaný v kapitole 3.4. Tato aplikace, stejně jako nástroj wireshark, je využívána správci systémů pro odhalení slabých míst sítí, ale také útočníky, kteří tyto slabé místa využívají k proniknutí do systému. [19]

4.4 Ettercap

Ettercap je open-source nástroj umožňující provádět útoky typu Man in the Middle na lokální síti. Tento multiplatformní nástroj je možno spouštět na operačních systémech jako je Linux, Mac OS X, BSD. Je vytvořen v programovacím jazyku C a distribuován pod svobodnou licencí GNU. Vývoj nástroje je stále aktivní, poslední stabilní verze byla představena v roce 2013 pod označením 0.8.0-Lacassagne.



Obrázek 4.4.1: Grafický vzhled aplikace a ukázka seznamu aktivních hostů na síti

Ettercap nabízí mnoho funkcí pro analýzu síti. Umožňuje filtrovat pakety podle IP a MAC adres, pomocí útoku ARP Cache poisoning přesměrovat veškerou komunikaci přes útočníka, sběr hesel nejrůznějších protokolů a další funkce. Předností této aplikace je možnost vyhledat ostatní útočníky v síti rozesílající podvržené pakety.

4.5 LOIC

Aplikace Low Orbit Ion Cannon (LOIC) umožňuje provést útok DoS na cílový server odesláním velkého množství paketů typu UDP nebo TCP a tím cílovou službu ochromit. Původně byla tato aplikace vytvořena firmou Praetox Technologies, v současné době je vyvíjena jako open-source projekt a distribuována pod veřejnou licencí. Je implementována v jazyku C#. Stejně jako předchozí představené aplikace je tento nástroj možné spouštět pod třemi nejpoužívanějšími operačními systémy. Tato aplikace byla inspirací pro vytvoření obdobné JavaScriptové aplikace nazvané JS LOIC poskytující stejné chování jednoduše ovladatelné pouze z webového prohlížeče.

LOIC aplikace byla využita známou skupinou počítačových pirátů sdružených pod názvem Anonymous při operaci nazvané PayBack. Cílem operace byly bankovní společnosti Visa, MasterCard a PayPal kvůli odmítnutí provádět finanční transakce zpravodajské organizaci WikiLeaks. [14]

Kapitola 5

Tvorba vlastních aplikací

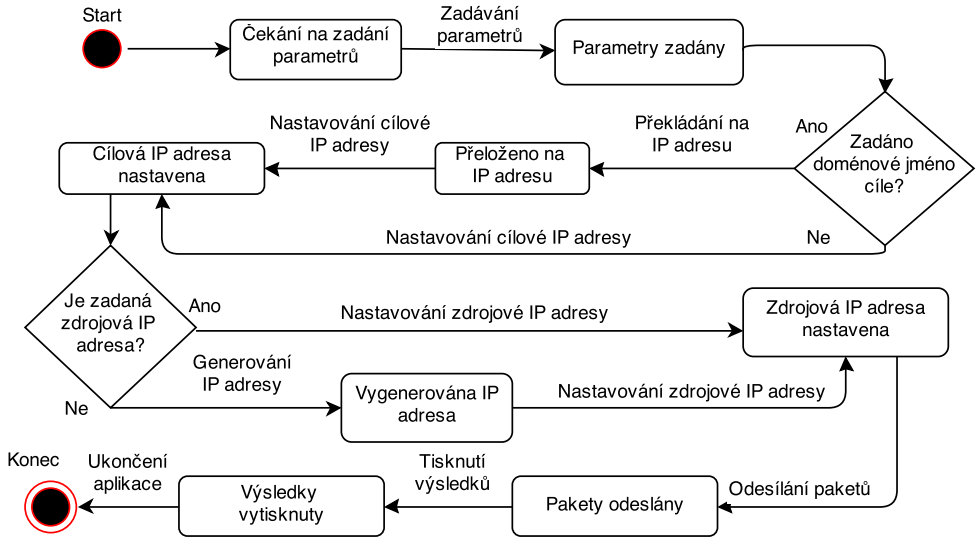
Cílem praktické části práce bylo vytvoření sady aplikací provádějící útoky typu Man in the Middle, IP Spoofing a SYN Flooding. Tyto aplikace jsou navrženy tak, aby je bylo možné spouštět jak v konzolém, tak i v grafickým režimu v operačním systému Linux. Důraz je taktéž kladen na jednoduché ovládání a grafickou přehlednost.

5.1 Návrh

Tato sekce popisuje návrhy jednotlivých aplikací. Jak je známo, důkladný návrh je základem každé schopné práce. Proto byl pro každou aplikaci vytvořen stavový diagram a také stanoveny cíle, které budou výsledné aplikace umožňovat provádět.

Aplikace pro útok typu SYN Flooding a IP Spoofing

Jak bylo popsáno v kapitole 3.1, cílem tohoto typu útoku je zahltit oběť tak, aby odmítala korektní požadavky o navázání nových spojení. Vykazování popsaného chování oběti útoku bylo stanoveno jako hlavní cíl aplikace. Stavový diagram popisující průběh aplikace je zobrazený na obrázku 5.1.1.



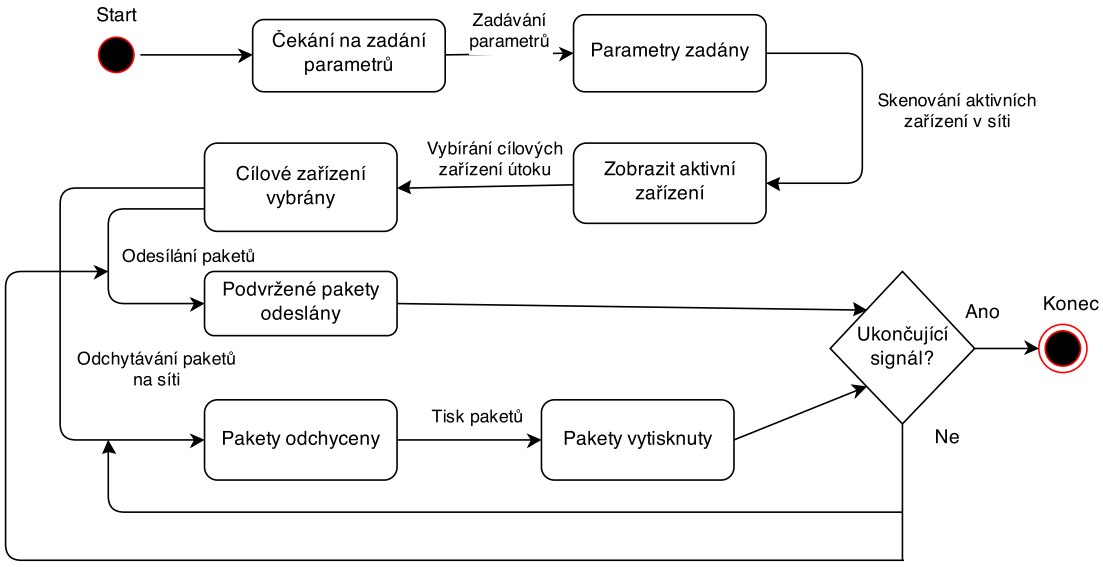
Obrázek 5.1.1: Stavový diagram aplikace provádějící SYN Flooding útok

Jak v konzolovém, tak v grafickém režimu je nutné zadat parametry pro správný průběh útoku. Mezi povinné parametry patří počet odesílaných paketů, IP adresa nebo doménové jméno oběti. Nepovinnými parametry je zdrojová IP adresa, která pokud není zadaná, je vyplněna náhodně vygenerovanou IP adresou. Taktéž nepovinným parametrem aplikace je generování náhodných zdrojových IP adres jedinečných pro každý odesílaný paket. Po správném zvolení parametrů je prováděn samotný útok. Jelikož mohou být v paketech podvržené zdrojové IP adresy, žádná odpověď nebude přijata a tudíž dochází pouze k výpisu chyb vzniklých při odesílání. Při správně provedeném útoku je konzolová aplikace ukončena, grafická aplikace může provádět libovolný počet útoků za sebou.

Aplikace pro útok typu Man in the Middle

Pro vytvoření aplikace simulující tento útok byl vybrán typ, představený v kapitole 3.3, ARP Cache Poisoning. Stanoveným cílem aplikace bylo odchytit a zobrazit komunikaci mezi dvěmi zařízeními v síti.

U konzolové aplikace je nutné zadat jako parametr programu typ rozhraní, které bude použito při útoku. U aplikace s grafickým rozhraním uživatel vybere příslušný typ ze zobrazené nabídky. V další fázi je nutné zjistit všechna aktivní zařízení na síti. K této činnosti jsou využity jednotlivé zprávy protokolu ARP. Po zpracování ARP odpovědí je uživatelovi zobrazen seznam dostupných zařízení. Následně je nutné vybrat ze seznamu dvě zařízení, mezi kterými bude odchytávána komunikace. Poté jsou vytvořeny a odesílány falešné ARP odpovědi v příslušném intervalu obětem útoku. Současně dochází ke skenování paketů, jejich zobrazení a přeposílání na jejich skutečný cíl. Stavový diagram popisující tento útok je možné vidět na obrázku 5.1.2.



Obrázek 5.1.2: Stavový diagram aplikace provádějící Man in the Middle útok

5.2 Použité technologie

Tato kapitola představuje jednotlivé technologie, které byly použity pro vytvoření výsledných aplikací. Je zde popsána jejich historie, stručná charakteristika, jejich využitelnost a aktuální dostupné verze.

Jazyk C++

Pro implementaci těchto nástrojů byl zvolen programovací jazyk C++. Tento jazyk byl vytvořen panem Bjarnem Strostrupem v Bellových laboratořích v roce 1979 pod označením *C with Classes*. Jak tento původní název napovídá, C++ vychází z jazyka C, do kterého byly přidány třídy, virtuální funkce, přetěžování operátorů, mnohonásobná dědičnost a vyjímky. Je považován za velmi výkonný programovací jazyk umožňující tři různé programovací způsoby: procedurální, objektově orientovaný a programování pomocí šablon. Taktéž je staticky typovaný, kompilovaný, multiplatformní. C++ je standardizován mezinárodní organizací pro standardizaci (ISO). Aktuálně vydaná verze je označována jako C++11, pod kterou byly tyto aplikace vytvářeny a překládány. [17]

Knihovna libcrafter

Tato linuxová knihovna[1] pro programovací jazyk C++ umožňuje jednoduše vytvářet, upravovat a odchytávat síťové pakety. Podporuje většinu běžných síťových protokolů jako jsou ARP, DHCP, IP, ICMP a další. Síťové pakety jsou tvořeny vrstvami, které jsou nad sebou vrstveny a lze k nim snadno přistupovat nebo je upravovat či mazat. Knihovna nabízí rozhraní pro vytvoření tzv. snifferu, který umožňuje skenovat a zachytávat pakety v síti. Pakety, které mají být odchyceny, mohou být tříděny pomocí filtrů mající formát typu pcap. Je zde implementovaný také TCP/IP zásobník, díky kterému je uživateli umožněno pracovat s TCP streamem. Knihovna je navržena tak, aby dovolila uživatelovi snadno vytvářet více vláknové programy, které mohou mezi sebou komunikovat. Nalezneme zde i metody pro čtení a zpracování souborů typu pcap.

Tato knihovna byla využita pouze v aplikaci provádějící útok typu Man in the Middle, verze použité knihovny libcrafter byla 0.2.

Knihovna Qt

Qt je multiplatformní framework[2] sloužící pro tvorbu programů v jazycích Python, Ruby, C++ a dalších. Obsahuje nástroje pro tvorbu uživatelských rozhraní, práci se sítí, grafikou a souborovým systémem. Tato knihovna byla vyvinuta v roce 1995 společností Trolltech, následně byla převzata společnostmi Nokia, Digia a v současné době je vyvijena otevřeně skupinou vývojářů pod označením *Qt Project*. Aplikace vytvořené za pomocí této knihovny mohou být spouštěny na desktopových platformách Windows, Linux, OS X a také na mobilních platformách Windows Phone, Symbian, Android, iOS. Aktuálně vydaná stabilní verze nese označení 5.2.1.

Veškeré grafické rozhraní aplikací bylo vytvořeno za pomocí této knihovny, použítá verze byla 5.0.2.

[1] https://code.google.com/p/libcrafter/
[2] http://qt-project.org/

5.3 Implementace

Tato kapitola obsahuje popis transformace návrhu na zdrojové kódy aplikace. Výsledné nástroje jsou tvořeny z modulů, jejichž konkrétní význam zde bude popsán. Veškerá implementace byla provedena na operačním systému Ubuntu 13.10, 64-bitové verzi. Grafický návrh byl proveden v Qt Creatoru verze 2.7.1.

Aplikace pro SYN Flooding a IP Spoofing

Hlavním jádrem této aplikace je třída *synFlood* umístěná v souboru *core.cpp* a *core.h*. Tato třída obsahuje definici mnoha metod a ty nejdůležitější zde budou představeny.

Metoda *int parseParametrs(int argc, char *argv[])* zajišťuje zpracování parametrů konzolové aplikace, ke kterému je využívána knihovna *boost*, konkrétně její část nazvaná *parse_options*. Seznam všech podporovaných parametrů aplikace, jejich význam a možné kombinace jsou zobrazeny v tabulce 5.1. Pokud jsou zadány nedefinované parametry, je zobrazena chyba na standartní chybový výstup *cerr* a program ukončen. Stejný průběh chování nastane při vložení neplatných tvarů IP adres či názvu hosta.

Parametr	**Význam a kombinace parametrů**
--help	Zobrazí nápovědu aplikace
--c hodnota	Počet odeslaných SYN paketů. Povinný parametr.
--tIP hodnota	IP adresa cíle. Nelze kombinovat s prepinacem –tHOST
--tHOST hodnota	Doménové jméno cíle. Nelze kombinovat s prepínačem –tIP
--s hodnota	Podvržená IP adresa odesilatele
--R	Generování náhodných zdrojových IP adres.

Tabulka 5.1: Seznam dostupných parametrů aplikace

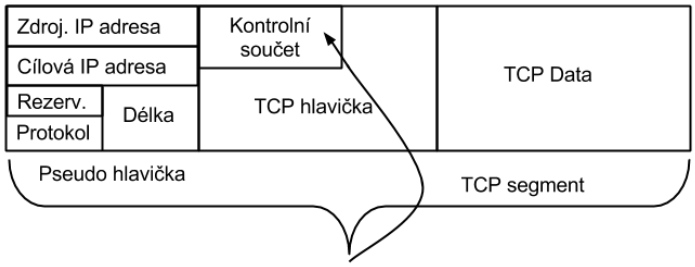
Metoda *setPacket()* slouží pro nastavení konkrétních vlastností paketů. Jelikož jsou využívány *RAW sokety*, je nutné každému paketu manuálně přidat IP a TCP hlavičky, které nejsou v tomto režimu soketů automaticky přidávány. IP hlavička paketu je vytvořena pomocí struktury *iphdr* dostupné v knihovně *netinet/ip.h*. Podle specifikace v RFC 791[3] je IP hlavička tvořena z položek, které tato datová struktura reprezentuje. Mezi nejdůležitější položky patří verze protokolu, která byla nastavena na IPv4, typem protokolu byl zvolen protokol TCP, zdrojové a cílové adresa zadány podle specifikovaných parametrů aplikace a nakonec kontrolní součet (checksum), který je popsán níže. Dále je v této metodě vytvořena TCP hlavička, k čemuž je využita struktura z knihovny *netinet/tcp.h* s názvem *tcphdr*. Mezi důležité položky této struktury patří zdrojový port, který obsahuje náhodně vygenerovanou hodnotu v intervalu 49152 - 65535, což je rozsah dynamických portů, které můžou být libovolně využívaný a nelze je zaregistrovat u organizace IANA[4]. Jako cílový port byl zvolen standartní HTTP port s číslem 80. Dále je nastaven příznak SYN a v neposlední řadě je nutné z těchto položek struktury vypočítat opět kontrolní součet.

Jak už bylo výše zmíněno, výpočet kontrolních součtů je nutný jak pro IP, tak i TCP hlavičky. Kontrolní součet je využíván při přenosu dat v síti k zajištění jejich integrity, tudíž je možné podle něj kontrolovat, zda přenos proběhl v pořádku či nikoliv. Pokud by byl tento

[3] http://www.ietf.org/rfc/rfc791.txt
[4] https://www.iana.org/

součet už při vytvoření špatně vypočítán, docházelo by, při první příležitosti, k zahození paketu. Metoda, která tento výpočet provádí, nese název *unsigned short checkSum(unsigned short *ptr, int numBytes)*. Prvním parametrem metody je ukazatel na příslušnou strukturu, druhý parametr označuje její velikost v bajtech. Před samotným výpočtem je nutné nastavit kontrolní součet na hodnotu nula. Výpočet kontrolního součtu IP hlavičky spočívá v součtu všech 16bitových binárních hodnot jednotlivých položek *iphdr* struktury. Následně je na tento součet aplikován jedničkový doplněk, který způsobí invertování jednotlivých bitů součtu.

Výpočet kontrolního součtu u TCP hlavičky se oproti IP hlavičce liší pouze v použití pomocné struktury s označením *pseudoHeader*. Tato pomocná hlavička je použita z historických důvodů, kdy ještě existoval pouze jeden protokol TCP. Po jeho rozdělení na protokoly TCP a IP bylo nutné zajistit protokolu TCP přístup k IP adresám a dalším položkám, které sloužily k ověření, zda byl paket doručen na správnou cílovou adresu či nikoliv. Struktura má velikost 12 bajtů a je využívána pro výpočet kontrolního součtu. Jsou v ní obsaženy položky pro zdrojovou a cílovou IP adresu, typ použitého protokolu, velikost TCP segmentu a jedno bajtová rezerva, jak je zobrazeno na obrázku 5.3.1. Následně probíhá výpočet stejným způsobem jako u hlavičky IP.



Obrázek 5.3.1: Struktura pseudo hlavičky a ilustrace připojení k TCP segmentu při výpočtu kontrolního součtu

Grafické rozhraní aplikace, zobrazeno na obrázku 5.3.2, je implementováno v třídě *MainWindow*, kterou lze najít v souboru *gui.cpp* a *gui.h*. Tato třída tvoří nadstavbu nad výše popsanou třídou *synFlood*. Grafické prvky jsou tvořeny tlačítky, stavovým panelem, textovými a zaškrkávajícími poli.

Zpracování parametrů probíhá současně s jejich zadáváním. Při každém vložení nového znaku IP adresy odesilatele či příjemce je vyvolána metoda *validIPAddr(QString)*, která kontroluje rozsah IP adres. Taktéž každé pole, umožňující vložení IP adresy, obsahuje validátor definovaný regulárním výrazem zajišťující její správný tvar. Pokud vložené znaky neodpovídají požadovanému tvaru, je uživatel upozorněn změnou barvy pozadí pole na červenou. Po zadání správného tvaru je pozadí změněno na barvu zelenou

Pro výběr mezi cílovou IP adresou a doménovým jménem slouží přepinače umístěné v pravé části vedle textových polí. Při změně výběru cíle je vyvolána metoda *setDest()*, která nevybrané pole zakáže. Současně dochází ke změně jeho pozadí na šedou.

Hlavními ovládacími prvky grafického rozhraní jsou tlačítka s názvem *Vymazat hodnoty* a *Odeslat pakety*. Po kliknutí na tlačítko *Vymazat hodnoty* je vyvolána metoda *reset()*, která uvede program do výchozího stavu. Signál vyvolaný tlačítkem *Odeslat pakety* je obsloužený metodou *doAttack()*, která provádí odesílání daných paketů. Veškeré informace o aktuálním

stavu aplikace, chybách, jsou zobrazeny ve stavovém řádku nacházejícím se na spodní části aplikace.

Obrázek 5.3.2: Grafický vzhled aplikace

Aplikace pro ARP Cache poisoning

Aplikace zastřešující tento útok je složena ze čtyř modulů, které mezi sebou vzájemně komunikují. V konzolové verzi aplikace je vyžadován pouze jediný parametr bez přepínače a to název rozhraní, které bude využito při útoku. Pokud by bylo zadáno neexistující rozhraní, je vypsána chyba a aplikace ukončena.

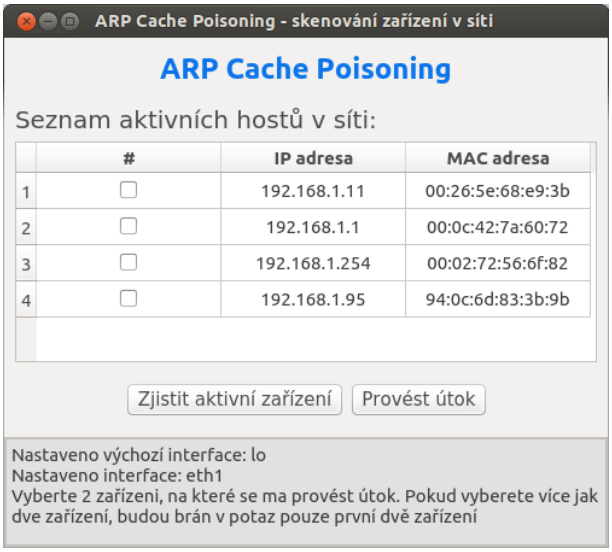
První důležitý modul této aplikace je tvořen třídou nazvanou *scanHost* umístěnou v souborech *scanHost.cpp* a *scanHost.h*. Jak už je z názvu zřejmé, náplní této třídy je získání všech aktivních zařízení v lokální síti. Řídící metoda této třídy je pojmenovaná názvem *getHost()*. V první řadě je v této metodě vytvořen tzv. sniffer, který slouží k zachytávání paketů. Díky definovanému filtru jsou odchytány pouze pakety typu ARP odpověď a ostatní typy jsou ignorovány. Následně jsou volány metody *setReqArpPacket()* a *sendSniffArp()*, které jsou vysvětleny níže.

V metodě *setReqArpPacket()* jsou vytvořeny pakety typu ARP požadavek a vyplněny útočníkovou IP a MAC adresou. Jelikož je brána v potaz pouze síť typu C, může být maximální počet vygenerovaných paketů 256. Dalším krokem je odeslání vygenerovaných paketů do sítě, což je zajištěno metodou *sendSniffArp()*. Po zachycení odpovědi na odeslané ARP požadavky je vyvolána statická metoda *getArpReply(Packet* scanPacket, void* user)* sloužící k jejich zpracování. Zpracování spočívá v uložení IP a MAC adresy do globálního modulu s názvem *glObSaveData*.

Globální instance třídy *glObSaveData* slouží především jako uložiště dat. Je zde uložen seznam aktivních zařízení na síti, IP a MAC adresy útočníka, obětí a další položky. Po správně provedeném podvržení ARP tabulek jsou zde uloženy pakety komunikace mezi obětmi útoku.

Modulem *ArpPoisoning* je prováděn samostatný útok. Metodou *setARPSpoof()* je vytvořeno nové vlákno, ve kterém jsou vygenerovány falešné ARP odpovědi a v intervalu pět vteřin odesílány na adresy obětí. Výše popsané chování vlákna je zajištěno metodou *sendingARPSpoof()*. Současně je aktivován další sniffer, který odchytává komunikaci mezi zvolenými oběťmi. Zpracování odchycených paketů komunikace je zajištěno statickou metodou *caughtPacket(Packet* scanPacket, void* user)*. Obsah paketů je vypsán přímo do terminálu, v grafickém rozhraní do příslušné komponenty. K ukončení útoku v konzolové verzi slouží zkratka ctrl+c, kdy dojde k ukončení skenování provozu a také zrušení vlákna rozesílajícího falešné ARP odpovědi.

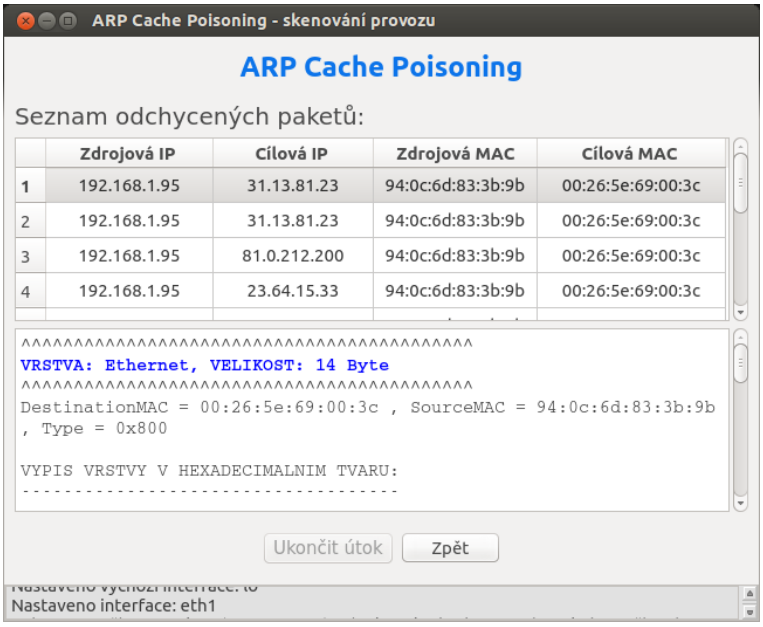
Poslední modul je tvořen třídou *MainWindow* poskytující grafickou nadstavbu nad dříve popsanými moduly. Oproti grafickým prvkům použitým v aplikaci SYN Flooding je zde využita komponenta s názvem *QTableWidget*. Do tohoto typu komponenty je následně vypsán jak seznam aktivních zařízení v síti, tak i zachycené pakety komunikace mezi oběťmi útoku. První část grafického rozhraní aplikace je možné vidět na obrázku 5.3.3.



Obrázek 5.3.3: První část grafického rozhraní aplikace

Po startu aplikace s grafickým rozhraním je zobrazeno metodou *chooseIntFace()* dialogové okno se seznamem síťových rozhraní mající přiřazenou IP adresu. Pokud uživatel žádné nevybere, je vybráno výchozí. Při stisknutí tlačítka *Zjistit aktivní zařízení* je emitovaný signál obsloužený metodou *getWriteHost()*, kde jsou volány metody již dříve popsaného modulu. Následně jsou zobrazeny výsledky do tabulky. Při aktivitě méně jak dvou zařízení na síti je vypsaná chyba do stavového řádku umístěného standardně ve spodní části aplikace. Pokud uživatel vybere více jak dvě oběti útoku, jsou nadbytečné vybrané zařízení ignorovány.

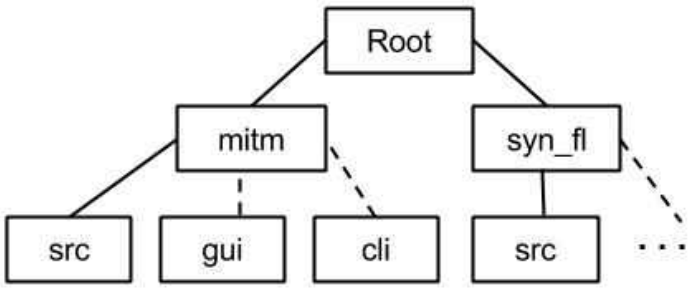
Při stisknutí tlačítka *Provést útok* je metodou *doAttack()* zobrazena druhá část grafického rozhraní a jsou volány metody již dříve představeného modulu *ArpPoisoning* k provedení cílového útoku. Tato druhá grafická část je zobrazena na obrázku 5.3.4



Obrázek 5.3.4: Druhá část grafického rozhraní aplikace

Souborový systém

Na diagramu 5.3.5 je zobrazeno uspořádání jednotlivých adresářů aplikací. Každý adresář aplikace obsahuje podadresář *src*, ve kterém jsou umístěny zdrojové kódy a taktéž soubory potřebné pro správný překlad aplikace. Po bezchybném překladu zdrojových kódů jsou vytvořeny další dva podadresáře s názvy *gui* a *cli* s binárními soubory. V složce *gui* se nachází aplikace spustitelná v grafickém režimu, v složce *cli* je možné spustit aplikaci v konzolovém režimu.



Obrázek 5.3.5: Souborový systém aplikace

5.4 Testování

V této kapitole jsou popsány navržené testy, předpokládané výsledky těchto testů a reálné výsledky získané vytvořenými aplikacemi. Testy byly zvoleny jak základní, tak i složité. Pokud byl nějaký test neúspěšný, jsou zde rozebrány příčiny, proč tomu tak bylo.

SYN Flooding a IP spoofing

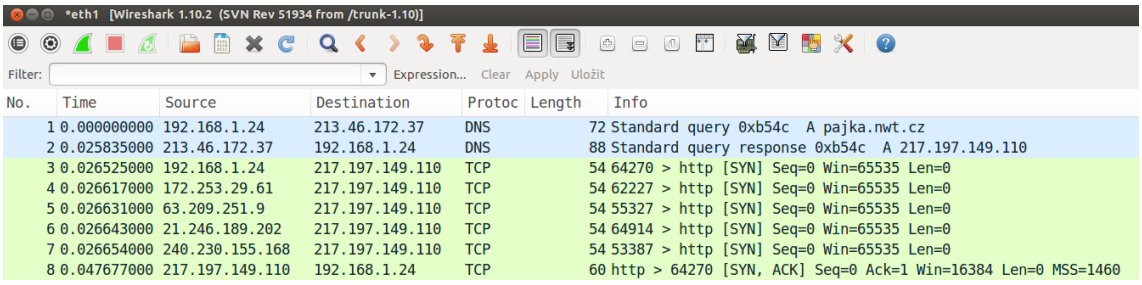
Pro testování aplikace SYN Flooding a IP Spoofing bylo nutné pronajmout či vytvořit cílový server. Proto byl zvolen server mající doménové jméno *pajka.nwt.cz* a IP adresu 217.197.149.110. Na tomto serveru byl nainstalován operační systém *OpenBSD* verze 5.3. SYN pakety a jejich odpovědi byly zobrazeny aplikací *tcpdump*. Aplikace byla spouštěna na počítači s 64bitovým operačním systémem *Ubuntu* verze 13.10.

Prvním jednoduchým testem bylo odeslání 5 paketů na IP adresu a na doménové jméno cíle. Tento jednoduchý test ukázal, zda aplikace generuje bez problémů pakety SYN, jestli jsou správně odeslány a zda jsou na ně vygenerovány odpovědi. Jak je možné vyčíst z výpisu *tcpdump* (5.1), byla na každý paket odeslaná odpověď a tudíž test proběhl v pořádku.

```
22:12:08.010428 10.0.0.1.80 > 94.113.109.123.53891:
S~3742214508:3742214508(0) ack 1 win 16384 <mss 1460> (DF)
22:12:08.010434 10.0.0.1.80 > 94.113.109.123.49933:
S~3263250581:3263250581(0) ack 1 win 16384 <mss 1460> (DF)
22:12:08.010440 10.0.0.1.80 > 94.113.109.123.55023:
S~3168874748:3168874748(0) ack 1 win 16384 <mss 1460> (DF)
22:12:08.010446 10.0.0.1.80 > 94.113.109.123.59810:
S~1627828381:1627828381(0) ack 1 win 16384 <mss 1460> (DF)
22:12:08.010469 10.0.0.1.80 > 94.113.109.123.64915:
S~178993448:178993448(0) ack 1 win 16384 <mss 1460> (DF)
```

Kód 5.1: Vygenerované odpovědi na SYN pakety zobrazeny aplikací tcpdump

Cílem druhého testu bylo zamaskovat útočníkovu skutečnou IP adresu. Opět bylo vygenerováno 5 paketů s náhodnými IP adresami včetně utočníkové aktuální. Následně byly tyto pakety odeslány, jak je zobrazeno na obrázku 5.4.1. Výsledkem testu bylo, že k oběti dorazil pouze jediný paket obsahující útočníkovu skutečnou IP adresu, na kterou mu byla odeslána odpověď. Tento výsledek naznačuje, že je na routeru (na který nebyl umožněn přístup) nastaveno filtrování, kdy nejsou propuštěny adresy s vnějším rozsahem IP adres. Zamaskovat útočníkovu adresu se nepovedlo a tudíž byl test neúspěšný.



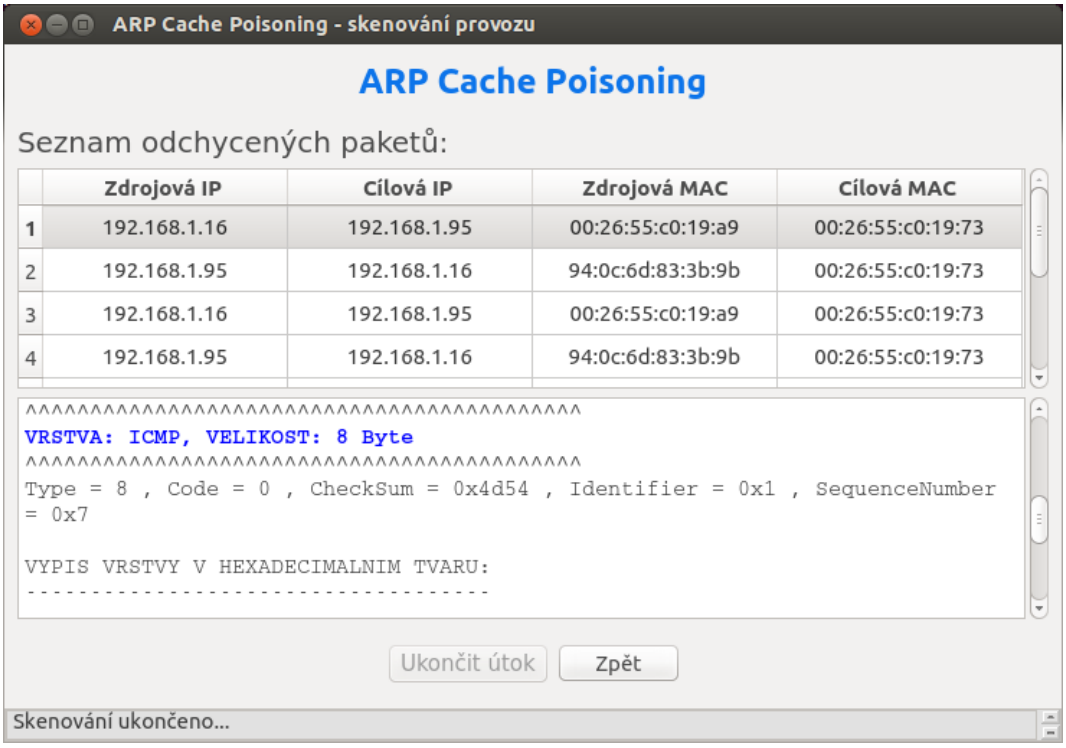
Obrázek 5.4.1: Zobrazení odeslaných paketů aplikací Wireshark

Třetím a posledním testem této aplikace bylo zahlcení cílové služby velkým množstvím paketů. Paketů bylo odesláno v řádu desítek tisíc, ale výsledkem bylo pouze nepatrné zpoždění služby. Ikdyž k úplnému zahlcení služby nedošlo, je možné výsledek označit za uspokojující.

ARP Cache Poisoning

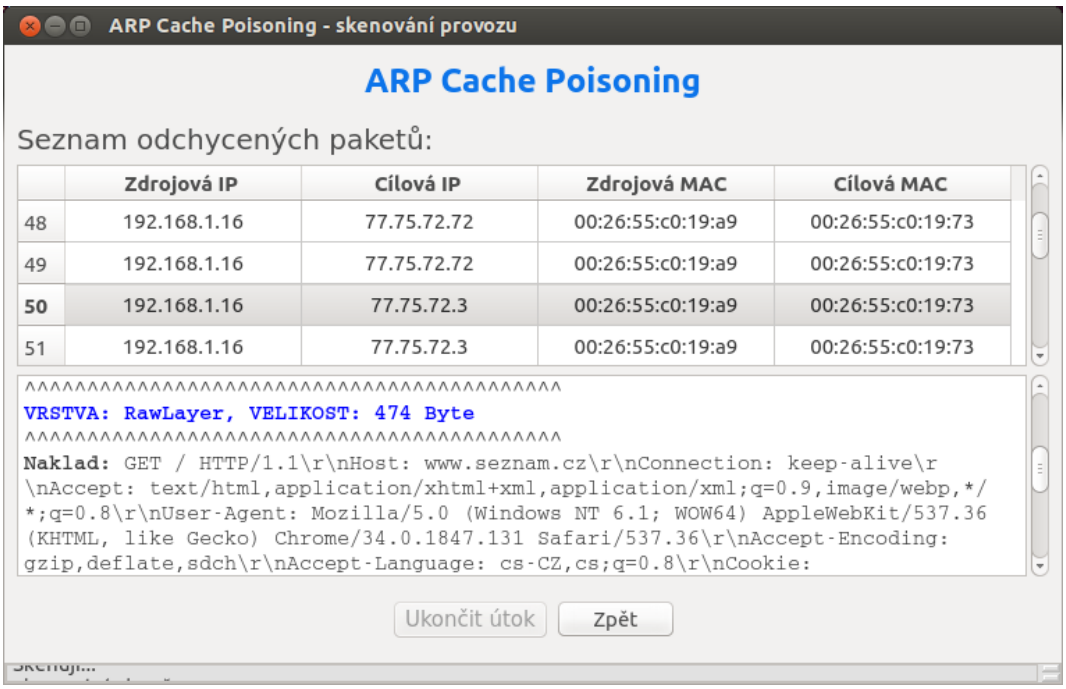
Testování druhé vytvořené aplikace bylo provedeno v drátové síti, jelikož, jak už bylo zmíněno v kapitole 2, je integrita dat v bezdrátových sítích zajištěna pomocí tajných klíčů či šifrovacích metod a tudíž by nebyl tento útok plně úspěšný. K testování byly využity 3 zařízení. Prvním zařízením byl notebook HP Probook 4510s s 64 bitovým operačním systémem Windows 7. Druhé zařízení představoval počítač s operačním systémem Windows XP, service pack 3. Tyto dva počítače byly využity jako oběti útoku. Posledním zařízením byl notebook HP Probook 4510s s operačním systémem Ubuntu verze 13.10, 64-bitové edice, na kterém byla spouštěna vytvořená aplikace. K propojení bylo uskutečněno pomocí kroucené dvojlinky a routeru TP-LINK verze TK-WR543G.

Po úspěšném propojení zařízení byl nejprve navrhnut test v podobě jednoduchého ověření správné funkcionality aplikace pomocí protokolu ICMP. V příkazovém řádku první oběti se zdrojovou IP adresou 192.168.1.16 byl zadán příkaz ping 192.168.95. Na obrázku 5.4.2 na řádcích 1 a 3 jsou zobrazeny první dva odeslané pakety. Odpovědi na tyto pakety je možné vidět na řádcích 2 a 4. V rámečku pod seznamem odchycených paketů je zobrazen obsah vrstvy paketu protokolu ICMP. Všechny pakety a odpovědi byly odchyceny a zobrazeny správně, tudíž test proběhl v pořádku.



Obrázek 5.4.2: Zobrazení odchycených paketů protokolu ICMP

Druhý test byl proveden opět se stejnými zařízeními a zapojením. Cílem útoku bylo odchycení komunikace mezi výchozí bránou routeru s IP adresou 192.168.1.1 a jedním zařízením s IP adresou 192.168.1.16. Do webového prohlížeče zařízení byl zadán požadavek na zobrazení stránky dostupné na adrese www.seznam.cz. Očekávaným výsledkem testu bylo odchycení jak požadavku, tak i odpovědi. Výsledkem testu, provedeným vytvořenou aplikací, bylo pouze odchycení odeslaného požadavku, jak lze vidět na obrázku 5.4.3. Odpovědi na tento požadavek se odchytit bohužel nepodařilo. Po důkladném prozkoumání příčiny bylo zjištěno, že rozesílané falešné ARP odpovědi byly routerem ignorovány, tudíž nebyla podvržena směrovací tabulka a odpovědi přicházely přímo k cílovému zařízení.



Obrázek 5.4.3: Zobrazení odchycených paketů HTTP požadavku

5.5 Zhodnocení výsledků

Aplikace SYN Flooding byla zdárně implementována a nachází se ve stavu, kdy umožňuje provést tento útok a zobrazit informace o jeho provedení. Jelikož byla tato slabina představená již před mnoha lety, jsou na většině zařízení implementovaný bezpečnostní prvky. Proto úspěšnost prováděného útoku vytvořenou aplikací nemusí být vždy sto procent. Aplikace byla porovnána taktéž s nástroji se stejným zaměřením. Srovnatelných výsledků bylo dosaženo mezi vytvořeným nástrojem a aplikací Hping. Tyto aplikace zahltily cílový server SYN pakety, ale omezit úplně jeho dostupnost se nepodařilo. U nástroje SynGui 2.0 nejsou vytvářeny tzv. poloviční spojení, na kterých je založena vytvořená aplikace, ale dochází zde k navázání úplného spojení. Tato aplikace byla úspěšná a došlo k úplnému zahlcení serveru a odmítání korektních požadavků. Mezi možnosti, kterými by se dalo vytvořenou aplikaci ještě více zdokonalit, by mohla být změna režimu vytváření spojení. Umožňovalo by to přepnout aplikaci na vytváření buď polovičních, nebo úplných spojení. Mezi další vylepšení by mohlo patřit vytváření seznamů s IP adresami neaktivních zařízení. Toho by

se docílilo při vygenerování nové náhodné IP adresy a následně by na ni byla odeslána zpráva protokolem ICMP. Pokud by v určitém časovém intervalu nedošlo k odpovědi, byla by IP adresa přidána na tento seznam. Ještě větší zvýšení účinnosti útoku by se dalo docílit vytvořením centrální řídící aplikace, která by ovládala vytvořené aplikace SYN Flooding. Tento nástroj by autor práce doporučil využívat správcům serverů, kteří chtějí otestovat jejich maximální zatížení a případně upravit jejich konfiguraci podle získaných výsledků. Nástroj se nedoporučuje využívat pro nelegální činnost.

Vytvoření druhé aplikace provázely komplikace v podobě špatného návrhu a následné implementace, která musela být zahozena. Novým návrhem a implementací byla aplikace dotažena do výsledné podoby představené v dřívějších kapitolách. V současném stavu vytvořený nástroj umožňuje podvrhnout ARP záznamy, odchytnout a zobrazit komunikaci. Odchycení veškeré komunikace mezi oběťmi není zaručeno, tudíž mohou být nějaké pakety přeslechnuty a nezobrazeny. Při srovnání vytvořené aplikace s nástrojem Ettercap bylo dosaženo stejných výsledků jak v prvním, tak při druhém testu. Mezi vylepšení, které by vytvořenou aplikaci rozšířily, by mohlo patřit přesměrování veškeré komunikace v síti přes útočníka. To by mohlo způsobit chaos mezi odchytnutými pakety, tudíž by bylo nutné implementovat další rozšíření v podobě filtrů, které by odchytnuté pakety třídily a zobrazovaly podle uživatelových požadavků. Tyto filtry by musely mít jednodušší syntaxi než v nástroji jako je například wireshark, aby je dokázal vytvořit a správně aplikovat i úplný začátečník v oblasti bezpečnosti. Jelikož existuje mnoho druhů útoku Man in the Middle (viz. kapitola 3.3), bylo by možné tyto útoky implementovat a následně přidat do této již vytvořené aplikace. Tím by vznikl kompaktní nástroj pro útoky typu Man in the middle, který na internetu doposud neexistuje. Zlepšení by se dalo taktéž aplikovat na grafické rozhraní aplikace. Jedná se především o modernější vzhled, lepší funkce pro zobrazovaní paketů a větší aplikační intuitivnost. Nástroj je doporučen používat správcům sítě pro testování jejich zabezpečení.

Kapitola 6

Závěr

V první částí práce se autor zabývá zabezpečením bezdrátových sítí. Provedeným průzkumem byl sestaven seznam všech nejznámějších a nejpoužívanějších typů zabezpečení, které jsou podrobně popsány v kapitole 2. Jak bylo zjištěno, mnoho z testovaných sítí využívá staré zabezpečení WEP, které neposkytuje dokonalou ochranu a taktéž je snadno prolomitelné. Některé sítě nepoužívaly dokonce zabezpečení žádné, tudíž jsou pro pomyslné útočníky snadným cílem.

V kapitole 3 se autor zaměřil na počítačové útoky, s kterými se v těchto sítích můžeme setkat. Tyto útoky jsou prováděny denodenně a využívají se k nejrůznějším účelům. Ve firemní sféře jsou použity pro ochromení konkurence, v poslední době také ve vojenské a politické sféře k odposlouchávání komunikace a odstavení důležitých zařízení cizích států. Výše zmíněné útoky byly reálně provedeny a otestovány nástroji, které jsou představeny v kapitole 4. Jak bylo z dosažených výsledků zjištěno, disponují tyto aplikace pokročilými funkcemi, které zvládne ovládat i středně pokročilý uživatel počítače.

Při implementaci a návrhu aplikací byl kladen důraz na snadné ovládání a maximální účinnost. Aby implementované útoky byly úspěšné, bylo nutné brát v potaz obranu, která tyto útoky eliminuje. Proto u návrhu aplikace pro SYN Flooding útok bylo zvoleno generování náhodných IP adres, které oběti znemožňují použití filtru na konkrétní IP adresu. U aplikace pro útok ARP Cache Poisoning je obrana v podobě statických ARP záznamů bezpečná, tudíž se autor práce zaměřil pouze na dynamicky se měnící záznamy.

V kapitole 5.4 jsou prezentovány dosažené výsledky vlastních testů provedených vytvořenými aplikacemi. Jelikož byly tyto útoky představeny před dlouhou dobou, jsou na většině udržovaných zařízeních implementovány obranné prostředky. Při útoku SYN Flooding bylo odesláno velké množství paketů SYN, ale k úplnému odstavení služby nedošlo. Odchycení paketů pří útoku Man in the middle bylo úspěšně provedeno.

Bohužel nic není ideální a tyto aplikace nejsou vyjímkou. Například aplikace nebyly otestovány na všech verzích systému Linux, tudíž může docházet k nežádoucímu chování na jednotlivých verzích. Možné rozšíření, které by chtěl autor do budoucna aplikovat, jsou zmíněny v kapitole 5.5. U aplikace SYN Flooding se jedná především o vytvoření seznamu neaktivních zařízení, u aplikace ARP Cache Poisoning o lepší zobrazování odchycených paketů a implementaci dalších typů Man in the Middle útoků.

Jednotlivé úkoly zadání se autor práce snažil popsat co nejnázorněji tak, aby byly pochopitelné i pro člověka, který se touto problematikou nezabývá. Myslí si, že se mu to podařilo a tím splnil zadání bakalářské práce.

Literatura

[1] *Scene of the Cybercrime: Computer Forensics Handbook: Computer Forensics Handbook*. Elsevier Science, 2002, ISBN 9780080480787.

[2] *Security in Distributed and Networking Systems*. Computer and network security, World Scientific Publishing Company, Incorporated, 2007, ISBN 9789812770103.

[3] The Heartbleed Bug [online]. `http://heartbleed.com/`, 2010-04-07 [cit. 2014-05-02].

[4] Allen, L.; Heriyanto, T.; Ali, S.: *Kali Linux – Assuring Security by Penetration Testing*. Packt Publishing, 2014, ISBN 9781849519496.

[5] Cardwell, K.: *Backtrack - Testing Wireless Network Security*. Community experience distilled, Packt Publishing, 2013, ISBN 9781782164074.

[6] Cole, E.: *Network Security Bible*. Bible, Wiley, 2011, ISBN 9780470570005.

[7] Coleman, D.; Westcott, D.; Harkins, B.; aj.: *CWSP Certified Wireless Security Professional Official Study Guide: Exam PW0-204*. Serious skills, Wiley, 2010, ISBN 9780470619629.

[8] Eddy, W.: TCP SYN Flooding Attacks and Common Mitigations. 8 2007.

[9] Edney, J.; Arbaugh, W.: *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Addison-Wesley, 2004, ISBN 9780321136206.

[10] Erickson, J.: *Hacking: The Art of Exploitation*. No Starch Press Series, No Starch Press, 2008, ISBN 9781593271442.

[11] Gibson, D.: *CompTIA Security+: Get Certified Get Ahead: SY0-301 Study Guide*. CreateSpace Independent Publishing Platform, 2011, ISBN 9781463762360.

[12] Halasz, D.: IEEE 802.11i and wireless security [online]. `http://www.eetimes.com/author.asp?section_id=36&doc_id=1287503`, 2008-11-01 [cit. 2014-06-06].

[13] KOLCUN, J.: *Zabezpečení Wi-Fi sítí*. 2010.

[14] Moses, A.: Australian Hacks MasterCard, Visa & PayPal Over WikiLeaks [online]. `http://www.theage.com.au/technology/security/the-aussie-who-blitzed-visa-mastercard-and-paypal-with-the-low-orbit-ion-cannon-20101209-18qr1.html`, 2010-12-09 [cit. 2014-06-06].

[15] Orebaugh, A.; Ramirez, G.; Beale, J.: *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Jay Beale's open source security series, Elsevier Science, 2006, ISBN 9780080506012.

[16] Pagani, M.: *Encyclopedia of Multimedia Technology and Networking*. Encyclopedia of Multimedia Technology and Networking, Idea Group Reference, 2005, ISBN 9781591407966.

[17] Prata, S.; Sokol, B.: *Mistrovství v C++*. Bestseller (Computer Press), Computer Press, 2007, ISBN 9788025117491.

[18] Rash, M.: *Linux Firewalls: Attack Detection and Response with Iptables, Psad, and Fwsnort*. No Starch Press Series, No Starch Press, 2007, ISBN 9781593271411.

[19] Sanfilippo, S.: Hping [online]. `http://www.hping.org/`, 2006 [cit. 2014-05-02].

[20] Špaček, M.: We Bleed: Heartbleed Bug in the Czech Republic [online]. `http://heartbleed.michalspacek.cz/`, 2010-04-15 [cit. 2014-05-02].

[21] ŠVÁB, R.: *Bezpečnost bezdrátových technologií*. Diplomová práce, Bankovní institut vysoká škola„ 2012 [cit. 2013-11-11].

Dodatek A

Obsah CD

Součástí této práce je datový nosič s následujícím obsahem:

- **sources** složka obsahující zdrojové soubory implementace
- **thesis** složka obsahující zdrojové soubory textové části
- **crafter.tar.gz** knihovna využitá při implementaci ARP Cache Poisoning
- **readme.txt** textový soubor s popisem aplikací
- **thesis.pdf** zpráva v elektronické podobě