Zadání diplomové práce (vloženo v tištěné verzi).

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Diplomová práce

Detekce zneužití VoIP ústředen

Bc. Lukáš Truxa

Vedoucí práce: Ing. Tomáš Čejka

5. ledna 2015

Poděkování

Poděkovat bych chtěl především vedoucímu diplomové práce Ing. Tomáši Čejkovi za odborné vedení, konzultace, rady a připomínky během tvorby práce. Dále děkuji sdružení CESNET za poskytnuté technické zázemí a umožnění přístupu pro testování modulu na datech z reálné sítě, což bylo při tvorbě této práce velkou výhodou. Poděkování také patří všem kolegům z projektu Liberouter, kteří se mnou spolupracovali a pomáhali při technických problémech. V neposlední řadě bych rád poděkoval celé mé rodině za podporu během mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či spracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 5. ledna 2015 .....

České vysoké učení technické v Praze
Fakulta informačních technologií

Odkaz na tuto práci

Truxa, Lukáš. *Detekce zneužití VoIP ústředen.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstract

The increasing importance and usage of voice as well multimedia communication via the Internet brings many security risks. The currently used private branch exchanges (PBX), which communicate via computer networks using the Voice over Internet Protocol (VoIP) technology, allow an interconnection of the world of Internet calls with the traditional public switched telephone networks (PSTN). In case of an insufficiently secured PBX, there is a danger of fraud. An attacker creates telephone calls to his phone numbers that are usually charged at higher rate exploiting PBX of a victim. This thesis contains the analysis of the Session Initiation Protocol (SIP) and it is focused on the development of a method for the VoIP exchange fraud detection. This thesis also contains the results of the deployment of the proposed detection module in a real computer network.

Keywords VoIP, SIP, PSTN, PBX, IP telephony, private branch exchanges, fraud detection, network security, prefix examination, localization of IP address, Nemea, TRAP, UniRec

Abstrakt

Vzrůstající význam a používání hlasové i multimediální komunikace na internetu s sebou přináší mnohá bezpečnostní rizika. Aktuálně používané telefonní ústředny (PBX) komunikující přes počítačové sítě pomocí technologie Voice over Internet Protocol (VoIP) umožňují propojení světa internetového volání s klasickými veřejnými telefonními sítěmi (PSTN). V případě, že je ústředna nedostatečně zabezpečena, hrozí nebezpečí jejího zneužití. V takovém případě je útočník schopen vytvořit telefonní hovory na svá telefonní čísla, která jsou často zpoplatněna zvýšenou sazbou. Tato práce se zabývá analýzou signalizačního protokolu Session Initiation Protocol (SIP) a detekční metodou pro odhalení pokusů o zneužití VoIP ústředen. Dále popisuje návrh a implementaci modulu schopného detekovat popsané zneužití. Součástí práce je i popis zaznamenaných útoků zachycených modulem v reálné síti.

Klíčová slova VoIP, SIP, PSTN, PBX, IP telefonie, ústředny, detekce zneužití, síťová bezpečnost, zkoušení předvolby, lokalizace IP adresy, Nemea, TRAP, UniRec

Obsah

Seznam obrázků

Seznam tabulek

Úvod

S postupným rozšiřováním technologie Voice over Internet Protocol (VoIP) vzrůstají i počty útoků, které se v počítačových sítích zaměřují výhradně na zařízení s podporou technologie VoIP. Pokud ústředna umožňuje volání do běžné telefonní sítě (PSTN) a útočníci dokáží úspěšně navázat telefonní hovory, dochází tím k finančním ztrátám provozovatele ústředny. V některých případech mohou ztráty dosáhnout až hodnoty jednotek milionů českých korun [44, 22]. Útočníci častěji využívají složitějších metod, aby útok nemohl být jednoduše rozpoznatelný. Proto se v dnešní době stává již nutností umět tyto útoky detekovat. Ve velkých sítích je nutné nasadit automatizované nástroje, které nahlašují podezřelý provoz. Správci počítačových sítí, VoIP ústředen, telefonů a dalších zařízení mohou potom nahlášené incidenty prozkoumat, vyřešit a lépe zabezpečit VoIP v jimi spravovaných sítích.

Diplomová práce se skládá ze dvou hlavních částí. Úkolem první části je seznámení s technologií VoIP. Popíšu, jak technologie funguje, jaké používá protokoly a detailně se zaměřím na protokol Session Initiation Protocol (SIP), který se používá pro signalizaci telefonních hovorů. Uvedu zasílané požadavky v protokolu SIP, jejich odpovědi a vysvětlím pojmy používané v protokolu podle RFC 3261 [40]. Následně popíšu zasílané hlavičky a vysvětlím, jak probíhá směrování v síti. Dále se zaměřím na různé druhy útoků na VoIP, SIP a principy zneužití nedostatečně zabezpečených telefonních VoIP ústředen. Uvedu, jak probíhá monitorování počítačové sítě pomocí tzv. toků a popíšu Nemea framework [16].

Ve druhé části se budu zabývat analýzou a výběrem vhodné detekční metody, která se na základě rozšířených informací o tocích z monitorovacích sond snaží odhalit zneužití telefonních ústředen na počítačové síti. Hlavním cílem práce je pak samotný návrh detekčního modulu pro systém Nemea, jeho implementace a testování v rámci reálné sítě.

Kapitola 1

VoIP telefonie a protokol SIP

1.1 Úvod do technologie VoIP

Voice over Internet Protocol (VoIP) je technologie zajišťující přenos hlasu a multimediálních dat přes počítačovou síť, respektive Internet Protocol (IP).

První komerčně dostupná technologie VoIP byla představena společností VocalTec v roce 1995. Vyžadovala u obou účastníků hovoru software nainstalovaný na běžném počítači s připojením k internetu. Díky masovému rozšiřování internetu v 90. letech se postupně stávala technologie VoIP levnější alternativou oproti běžné veřejné telefonní síti (Public Switched Telephone Network – PSTN). [43]

Dnes již existuje mnoho specializovaných zařízení pro VoIP telefonii jako IP telefony, VoIP brány pro připojení k veřejné telefonní síti či celé ústředny. Na trhu je dnes mnoho výrobců, na ukázku lze jmenovat např. Cisco, Panasonic nebo Toshiba. V dnešní době se VoIP běžně používá ve velkých organizacích, jako jsou velké firmy a vysoké školy. Postupně se tato technologie začíná využívat i v menších organizacích a domácnostech.

S rozšiřováním rychlého a kvalitního internetového připojení pro mobilní zařízení se VoIP prosazuje i v mobilních telefonech, tzv. „mobile VoIP“ (mVoIP) [33]. Textovou, hlasovou komunikaci i videokonferenci je možné přenášet přes datové připojení k internetu namísto využití konkrétních služeb a tarifů mobilního operátora. Využívání těchto služeb pak může být díky výhodným tarifům na datové přenosy levnější při srovnatelné kvalitě.

1.1.1 Používané protokoly

Obecně ve VoIP telefonii pro přenos multimediálních dat není výhodné využívat jako transportní protokol Transmission Control Protocol (TCP). Ten zajišťuje spolehlivý přenos dat, při ztrátě dat dochází k opětovnému pokusu

o doručení a ke zvyšování zpoždění. Ztráta dat během přenosu také vede k rychlému snižování přenosové rychlosti. Využívá se proto většinou nespolehlivý transportní protokol User Datagram Protocol (UDP), který je vhodný pro VoIP komunikaci. Díky tomu, že není potřeba navazování spojení (handshaking) a nevyužívá metodu potvrzování doručení paketů, může být efektivněji využita šířka pásma a sníženo zpoždění během komunikace. Krátkodobé výpadky během přenosu nemají takový vliv na kvalitu hovoru jako vyšší zpoždění přenosu.

Pro sestavení, řízení a ukončení spojení je dnes využíván především signalizační protokol Session Initiation Protocol (SIP), jeho detailní popis bude uveden v sekci 1.2. Jako starší alternativní standard je pro signalizaci používán protokol H.323. Uvedené dva protokoly pracují na principu klient-klient (peer-to-peer), existují však i protokoly pracující na principu, kdy jedno zařízení řídí komunikaci s druhým (master-slave). Jsou to protokoly Media Gateway Control Protocol (MGCP) nebo Megaco. MGCP je založeno na modelu telefonie PSTN. [28]

Přenos zvukových a obrazových dat probíhá protokolem Real-time Transport Protocol (RTP), umožňuje jak zasílání jednomu zařízení (unicast), tak skupině více zařízení (multicast). Samotné řízení datového toku včetně kontroly kvality a nastavení kodeku poskytuje protokol Real-time Transport Control Protocol (RTCP). Pro zachování důvěrnosti přenášených multimediálních dat, jejich autentizaci a ochranu před opětovným zasíláním zachycených dat (replay attack) může být využit Secure Real-time Transport Protocol (SRTP).

Pro přenos zvuku existuje několik standardizovaných kodeků, např. G.711, G.726, G.729. Liší se především potřebnou propustností a kvalitou zvuku. V závislosti na použitém algoritmu se různí vliv okolního šumu a vliv na přenos zvuku v případě krátkodobého výpadku kanálu (ztráta paketu). [28]

1.1.2 Požadavky technologie

Pro správnou funkci VoIP je požadováno stabilní síťové připojení s následujícími vlastnostmi:

- dostatečná propustnost (throughput) – rychlost stahování (download) a odesílání (upload),
- nízká ztrátovost paketů (loss rate),
- malé zpoždění (delay),
- nízké kolísání velikosti zpoždění (jitter).

Pro zajištění výše uvedených vlastností je většinou používáno v síti řízení datových toků Quality of Service (QoS), tím dochází k prioritizaci komunikace VoIP telefonie. [28]

1.2 Protokol SIP (Session Initiation Protocol)

Kapitola čerpá především z RFC 3261 [40] a dále ze zdroje [28].

Aplikační protokol Session Initiation Protocol (SIP) je signalizační textový protokol podobný HTTP či SMTP a zajišťuje sestavení, řízení a ukončení telefonního spojení. Je nezávislý na přenášených datech.

SIP používá pro adresaci jednotný identifikátor zdroje, tzv. Uniform Resource Identifier (URI), který začíná „sip:“, resp. „sips:“, pokud je vyžadována zabezpečená komunikace, která je běžně realizována pomocí Transport Layer Security (TLS) spojení. SIP URI obsahuje dostatek informací pro úspěšné navázání a samotný průběh komunikace. Identifikace je velmi podobná emailové adrese, ke které jsou připojeny parametry a hlavičky:

```
sip:[uživatel[:heslo]@]host[:port][;uri-parametry][?hlavičky]
```

Popis jednotlivých částí SIP URI:

- *uživatel* je identifikátor cílového účastníka (klienta)
- *heslo* pro daného uživatele; zasílání hesla takovýmto způsobem je však silně nedoporučováno, protože heslo je zasíláno ve formě běžného textu a je tedy viditelné pro všechny, kteří mohou zachytit tuto komunikaci
- *host* představuje cílový server (zařízení), na který má být požadavek směrován, obvykle je to plně specifikované doménové jméno (Fully Qualified Domain Name – FQDN) nebo IP adresa; využívat FQDN je doporučováno vždy, když je to možné
- *port* – číslo portu, na který má být požadavek zaslán
- *uri-parametry* mohou ovlivnit zpracování požadavku, v minulosti byly především používány:
  - *maddr* parametr, který dovoluje určit adresu serveru, na který bude požadavek zaslán; má vyšší váhu než adresa získaná z části *host*; byl používán jako jednoduchá změna směrování požadavku, používání tohoto požadavku je však zastaralé a všechny implementace by měly využívat směrovací proces pomocí *Request-URI* (bude vysvětleno později)
  - *ttl* parametr určuje Time to live (TTL) hodnotu UDP multicast paketu a musí být používán pouze v případě, že *maddr* je multicast adresa a transportní protokol je UDP

  Uri-parametry, které nejsou danou SIP ústřednou, telefonem nebo jiným SIP zařízením rozpoznány jsou jednoduše ignorovány.

- *hlavičky* mohou obsahovat další informace, ve formátu `jméno=hodnota`

Povinnou částí všech SIP URI je pouze část *host*, ostatní části jsou volitelné.

1.2.1 Požadavky

Specifikace protokolu SIP podle RFC 3261 [40] definuje 6 typů požadavků:

- REGISTER – registrace účastníka na registračním serveru
- INVITE – zahájení relace (hovoru)
- ACK – potvrzení zahájení relace
- CANCEL – přerušení během zahajování relace
- BYE – ukončení relace
- OPTIONS – žádost o informace o podporovaných možnostech protistrany

V dalších RFC jsou specifikovány i další typy požadavků, které především souvisí se zasíláním událostí, zpráv a informací o probíhající komunikaci. Více informací lze nalézt v příslušných RFC 3261, 3311, 3428, 3515, 3903, 6086, 6665. Tato práce se bude dále zaměřovat na využití základní vyjmenované sady typů požadavků.

1.2.2 Odpovědi na požadavky

Odpověď na požadavky vždy obsahuje verzi používaného SIP protokolu, včetně třímístného číselného kódu, podle kterého je určen výsledek zpracování požadavku. Typy odpovědí jsou podobné jako v protokolu HTTP:

- `1xx` : *Prozatímní odpověď* – požadavek byl přijat, proces zpracovávání pokračuje (jediný typ odpovědi, který není konečnou odpovědí)
- `2xx` : *Úspěch* – požadavek byl úspěšně přijat a proveden
- `3xx` : *Přesměrování* – nutné provedení další akce k úspěšnému dokončení požadavku
- `4xx` : *Klientská chyba* – požadavek obsahuje chybnou syntaxi nebo nemůže být proveden na tomto serveru
- `5xx` : *Chyba serveru* – server nemohl provést požadavek; odeslaný požadavek má správný formát
- `6xx` : *Obecná chyba* – požadavek nemůže být proveden na žádném serveru

Konkrétní často používané odpovědi jsou např.:

- `100 Trying` – požadavek byl přijat, probíhá zpracovávání požadavku
- `180 Ringing` – probíhá pokus o upozornění cílového uživatele (vyzvání)
- `200 OK` – požadavek byl úspěšný
- `400 Bad Request` – neplatný požadavek (chybný formát)
- `401 Unauthorized` – požadavek vyžaduje autentizaci
- `403 Forbidden` – požadavek má správný formát, je však odmítnuto jeho zpracování, požadavek by neměl být znovu zasílán
- `407 Proxy Authentication Required` – klient musí být autentizován před zasláním požadavku proxy serveru
- `500 Server Internal Error` – požadavek nemohl být zpracován, nastala neočekávaná chyba

1.2.3 Vysvětlení pojmů používaných v protokolu SIP

- *SIP transakce* dle RFC 3261 probíhá mezi klientem a serverem od prvního požadavku zaslaného klientem do finální odpovědi serveru, která není odpovědí typu 1xx.
- Během samotné SIP komunikace mohou vystupovat jednotliví klienti protokolu SIP, resp. uživatelští agenti ve dvou logických rolích:
  - *User Agent Client* (UAC) je logická část, která vytváří a odesílá požadavky na User Agent Server (UAS)
  - *User Agent Server* (UAS) je logická část, která přijímá požadavky od User Agent Client (UAC) a následně odesílá odpovědi

  Tyto role jsou platné pouze po dobu jedné transakce. Při další transakci mohou klienti vystupovat v opačné roli. Například klient zasílající INVITE požadavek se chová jako UAC, při přijímání požadavku BYE jako UAS.

  Uživatelským agentem (UA) neboli koncovým zařízením může být jakékoli hardwarové zařízení nebo software, který je schopný odesílat a přijímat požadavky SIP protokolu.

Poznámka: V dalším textu bude SIP požadavek i odpověď jednotně označován jako *SIP zpráva*.

- *SIP proxy server* neboli SIP proxy – přijímá SIP zprávy, zpracovává je a následně zasílá na další SIP proxy server nebo jiný cíl. Je tedy prostředníkem pro komunikaci a rozhodnutí, kam daná zpráva bude odeslána, závisí na něm. Zpráva je dále směrována tak, aby bylo zajištěno doručení cílovému prvku. Proxy server může provádět kontrolu, zda daný uživatel je oprávněn k zasílání daného požadavku a může ho odmítnout. Často nemají klienti v LAN síti přímý přístup do internetu a pokud chtějí využívat VoIP komunikaci musí komunikovat právě pomocí proxy serveru, který již má přístup do internetu. Pokud je to nutné, proxy server mění části požadavku během zpracovávání a následně odesílá již pozměněnou zprávu.

  Proxy server může být:

  - *bezestavový* (Stateless Proxy) – zprávy pouze přijme a přeposílá na definovaný cíl, neukládá si žádné informace o přijatých nebo odeslaných zprávách,
  - *transakčně stavový* (Transaction Stateful Proxy) – ukládá si informace o probíhajících transakcích, tyto informace pak používá při zpracovávání dalších příchozích zpráv; zprávy může přeposílat i na více cílů,
  - *hovorově stavový* (Call Stateful Proxy) – ukládá si veškeré informace o probíhajících hovorech a dokáže určit, v jakém stavu se hovor nachází, sleduje hovor od počátečního INVITE do ukončovacího BYE požadavku. Hovorově stavová proxy je vždy zároveň transakčně stavová. Na rozdíl od transakčně stavové dokáže zpracovávat menší počet souběžných spojení (hovorů).

- *Redirect Server* zasílá odpovědi na přijaté požadavky ve formě požadavku na přesměrování na určenou adresu. Klient, který se dotazoval redirect serveru přijme odpověď a následně naváže spojení na předanou adresu. Redirect server žádným způsobem nenavazuje komunikaci s cílovým účastníkem.

- *Lokalizační služba* (Location Service) je využívána Redirect serverem nebo proxy serverem. Poskytuje informace, kde se nachází volaný účastník, tedy jeho aktuální IP adresu a port.

- *Registrační server* (Registrar Server) přijímá REGISTER požadavky, ověřuje je a ukládá si informace o umístění účastníka. Tyto informace dále poskytuje lokalizační službě.

- *Brána* (Gateway) zajišťuje spojení do jiných technologií, často do PSTN, GSM sítě.

Proxy, redirect a registrační server jsou většinou implementovány v jednom zařízení, resp. v jedné aplikaci.

1.2.4 Ukázka průběhu hovoru

Na obrázku 1.1 je uvedena ukázka navázání, průběhu a ukončení hovoru:

Alice zavolá Bobovi, jehož SIP URI je `sip:bob@sip2.cz` (`sip2.cz` je doména Bobova poskytovatele SIP služeb). VoIP telefon Alice nezná konkrétní umístění Boba, proto zašle požadavek na SIP proxy server, který zajišťuje SIP služby pro Alici (`sip1.cz`). Určení SIP proxy serveru je dáno nastavením telefonu (nastavení může být načteno např. pomocí Dynamic Host Configuration Protocol – DHCP).

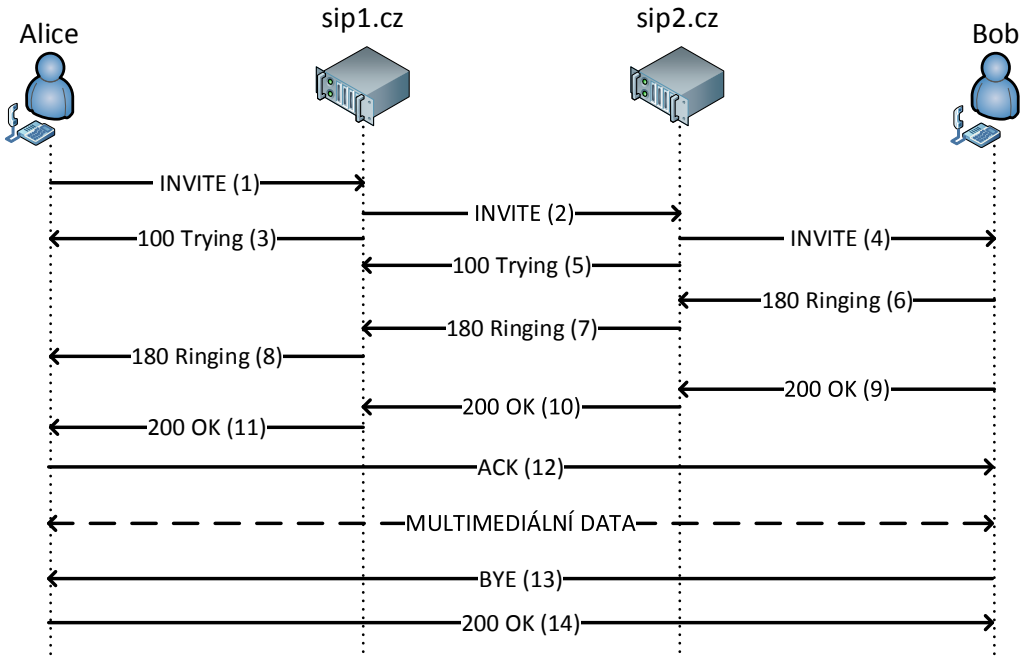
Proxy server `sip1.cz` obdrží INVITE požadavek a zašle odpověď `100 Trying` zpět do telefonu Alice. Poté `sip1.cz` nalezne proxy server `sip2.cz` pomocí služby Domain Name System (DNS) a na získanou IP adresu přepošle INVITE požadavek. `Sip2.cz` obdrží požadavek a odpoví pomocí `100 Trying` proxy serveru `sip1.cz`. Proxy server `sip2.cz` pomocí lokalizační služby nalezne umístění Boba a přepošle mu INVITE požadavek. Bobův telefon začne vyzvánět a odešle odpověď `180 Ringing`, která je přes proxy servery směrována do telefonu Alice. Po přijmutí `180 Ringing` začne telefon Alice informovat Alici o vyzvánění (zvukově nebo zobrazením informace v textové podobě).

V tomto příkladu, Bob přijme hovor a jeho telefon zašle odpověď `200 OK`, která obsahuje informace pro zajištění přenosu multimediálních dat a dále obsahuje *Contact* hlavičku, která umožňuje přímé navázání následné komunikace pro signalizaci mezi koncovými telefony. Jakmile telefon Alice přijme `200 OK` odpověď, přestane informovat o vyzvánění a zašle ACK požadavek přímo telefonu Boba (v této ukázce proxy servery nedokážou zjistit další průběh hovoru). Následuje vzájemná výměna multimediálních dat, tato data mohou být zasílána jinou cestou než signalizační protokol.

Obecně mohou být během hovoru zasílány opětovně INVITE požadavky pro změnu parametrů hovoru, druhá strana potvrdí přijmutí změny odpovědí `200 OK` a následuje potvrzení ACK tou stranou, která požadovala provést změnu parametrů.

Ukončení hovoru je provedeno Bobem, jeho telefon zasílá BYE požadavek a telefon Alice potvrdí přijmutí `200 OK` odpovědí. Tímto dochází k ukončení hovoru. Protože BYE požadavek byl odeslán telefonem Boba, je uvedeno jeho SIP URI v hlavičce *From* a hlavička *To* obsahuje SIP URI Alice.

Poznámka: Číselné hodnoty uvedené v závorkách udávají pořadí zasílání jednotlivých požadavků a odpovědí.



Obrázek 1.1: Ukázka průběhu hovoru

1.2.5 Hlavičky zasílané v požadavcích a odpovědích

Každý zaslaný SIP požadavek nebo odpověď na něj obsahuje definované hlavičky protokolu. Nejdůležitější jsou:

- *Request-URI:* požadavek je směrován právě díky této hlavičce a může být změněna SIP proxy serverem; uložen je v první řádce požadavku; při generování nové zprávy UA by měla být shodná s *To:* hlavičkou (mimo výjimky, která nastává u REGISTER požadavků)
- *To:* identifikace cílového příjemce (volaného účastníka); většinou obsahuje SIP URI, může však obsahovat i jiná podporovaná schémata pro telefon uvedená v RFC3966 [41]; může být zadán také popis, který je zobrazován jako popisné kontaktní jméno
- *From:* identifikace zdrojového odesílatele (volající účastník); typ obsahu je stejný jako u *To:* hlavičky; musí se počítat s tím, že *From:* hlavička nemusí obsahovat zdrojovou IP adresu nebo FQDN hosta, na kterém běží UA
- *Call-ID:* textový řetězec, který identifikuje skupinu zasílaných zpráv (hovor); každý UA musí zajistit, aby jím vygenerované *Call-ID:* bylo unikátní a nemohlo být vygenerováno stejné *Call-ID:* jiným UA

- *CSeq:* slouží k identifikaci pořadí zaslaných požadavků a odpovědí
- *Max-Forwards:* určení maximálního počtu „hopů“ při směrování požadavku, pokud je počet překročen, zpráva je odmítnuta kódem 483 (Too Many Hops)
- *Via:* určuje cestu, kterou požadavek prošel, než dosáhl cílového příjemce a tedy i cestu, kterou se bude vracet odpověď na daný požadavek; když vytváří UAC požadavek, musí vložit *Via:* hlavičku; hlavička dále obsahuje tzv. „branch“ parametr, který je používán k identifikaci transakce vytvořené požadavkem (je využíván jak UAC, tak UAS)
- *Contact:* adresa účastníka odesílající zprávu, na které může být kontaktován
- *Route:* seznam proxy serverů, které se mají použít při směrování požadavku
- *Record-Route:* hlavička přidávaná proxy servery do požadavku, která zajišťuje, aby zprávy mohly být zasílány přes uvedené proxy servery (lze označit jako záznam cesty přes proxy servery)
- *User-Agent:* obsahuje informace o UAC, které vytvořilo požadavek (identifikace hardwarového zařízení nebo softwarové aplikace, často včetně používané verze)
- *Content-Type:* určuje typ dat zasílaných v těle požadavku
- *Content-Length:* určuje délku dat zasílaných v těle požadavku
- *Allow:* seznam metod, které podporuje UA, který vytvořil požadavek
- *Supported:* seznam rozšíření, které podporuje UAC nebo UAS

Ukázka INVITE požadavku:

```
INVITE sip:420789123456@1.2.3.4 SIP/2.0
Via: SIP/2.0/UDP 5.6.7.8:5060;branch=z9hG4bK3297872b
Max-Forwards: 70
From: "420111222333" <sip:4201112223335.6.7.8>;tag=as5074bd90
To: <sip:420789123456@1.2.3.4>
Contact: <sip:420111222333@5.6.7.8:5060>
Call-ID: 7ee72be42c3752aa76cb6a9409724a65@5.6.7.8:5060
CSeq: 102 INVITE
User-Agent: FPBX-2.10.1(1.8.7.1)
Date: Fri, 03 Oct 2014 10:00:00 GMT
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, \
       NOTIFY, INFO, PUBLISH
```

```
Supported: replaces, timer
Content-Type: application/sdp
Content-Length: 262
```

1.2.6 Směrování odpovědi

Směrování probíhá pomocí hlavičky *Request-URI*, která obsahuje vždy dalšího příjemce v „cestě“ sítí. Každý UAS uloží do hlavičky *Via* svou adresu, aby mohly být zasílány odpovědi na požadavky stejnou cestou. UA, který na požadavek odpovídá, zkopíruje všechny položky *Via* do odpovědi. Všechny prvky, které přeposílají danou odpověď, odstraní svou adresu z *Via* a zašlou odpověď na další prvek v pořadí. Jakmile dojde k výměně požadavku a následné odpovědi, mohou klienti začít komunikovat přímo bez přeposílání zpráv přes SIP proxy servery. Často je však potřebné, aby SIP proxy server byl také informován o průběhu hovoru a právě pro tyto případy je využíváno *Route* a *Record-Route* hlaviček popsaných výše, které zajišťují směrování požadavků. *Via* hlavička se i v tomto případě používá pro směrování odpovědi na požadavek.

1.2.7 Používané porty pro komunikaci

Výchozí používaný port protokolu SIP je 5060 pro UDP i TCP. Pokud je komunikace šifrovaná pomocí TLS, výchozí port je 5061.

Protokol SIP může být používán na jakémkoliv portu, jediné, co je potřeba zajistit, je správná konfiguraci na všech zařízeních a v softwarových aplikacích využívajících SIP protokol.

1.3 VoIP ústředny

VoIP ústředna je specializované zařízení nebo softwarový nástroj s mnoha funkcemi, které jsou vyžadovány především v organizacích. Zajišťuje správu telefonních hovorů uvnitř organizace pomocí vlastního číselného schématu (většinou tří až čtyřciferná čísla). Umožňuje volat na externí čísla s použitím nastavené předvolby, podporuje automatické přehrávání pozdravu a možnost automatické odpovědi. Poskytuje rozsáhlé možnosti správy a obsluhy hlasových schránek, umožňuje využívat pouze jedno telefonní číslo z veřejné telefonní sítě pro celou organizaci a volající si po spojení hovoru může zvolit, s jakým konkrétním uživatelem bude spojen. Často dokáže vytvořit konferenční hovor s více účastníky a umožňuje převzetí volání jiným uživatelem. Další funkce se zaměřují na ukládání informací o využívaných službách, statistiky jako je doba jednotlivých hovorů, jejich kvalita a statistiky pro jednotlivé uživatele. [42]

Často využívané softwarové VoIP ústředny, které jsou vyvíjeny jako open-source řešení, jsou např. Asterisk [1] a Kamailio [5].

1.4 Útoky na technologii VoIP a protokol SIP

1.4.1 Hlavní cíle útoků

Hlavní a nejnebezpečnější cíl útoku je finanční prospěch, který může dosahovat až jednotek miliónů českých korun [44]. Ve VoIP telefonii se jedná především o zneužití telefonních čísel se zvýšenou sazbou, tzv. prémiových telefonních čísel, které si založí sám útočník a za příchozí hovory získává finance. Prémiová telefonní čísla pro volání v České republice jsou devítimístná čísla začínající předvolbami 900, 906, 908 a 909. Cena hovoru pro volající stranu v českých korunách je vyjádřena čtvrtou a pátou číslicí volaného čísla. Podle ceníku [36] může být hovorné účtováno za každou započatou minutu nebo za jedno spojení. Útočníky je většinou využíváno volání do zahraničí, kde jsou používány odlišné předvolby.

Již v historii útočníci využívali tzv. „dialery“ pro změnu vytáčeného čísla pro internetové připojení modemem. Útoky, které se v dnešní době používají ve VoIP telefonii lze považovat za pozměněnou formu dříve využívaného útoku.

V České republice byl medializován úspěšný útok, kdy došlo ke zneužití špatně zabezpečené VoIP ústředny. Jak bylo uvedeno v reportáži České televize [44], škoda se vyšplhala až k 1,4 miliónu českých korun. Útočník v tomto případě přesměrovával hovory do Lichtenštejnska.

Experti v uveřejněné zprávě [22] odhadují, že v roce 2013 bylo díky podvodům celosvětově v telekomunikacích ztraceno 46,3 miliard amerických dolarů, což je oproti roku 2011 zvýšení o 15 %. Nejvyšší počet útoků byl směrován do těchto cílových destinací: Lotyšsko, Gambie, Somálsko, Sierra Leone a Guinea. Útoky ve VoIP jsou odhadovány na 3,62 miliard amerických dolarů, zneužití podnikové telefonní ústředny na 4,42 miliard amerických dolarů.

Další útoky jsou cíleny na:

- zjišťování citlivých údajů, které mohou být zneužity,
- skrytí vlastní identity (např. vydírání),
- odposlech, záznam či úprava hovoru,
- jiné důvody – např. vyzkoušení návodu nalezeného na internetu [46].

1.4.2 Popis útoků

Kapitola čerpá ze zdroje [25].

Technologie VoIP používá na síťové vrstvě Internet Protocol (IP), dále využívá pro svůj běh ostatní služby a protokoly sítě (např. Address Resolution Protocol – ARP, DHCP, DNS). Pokud je proveden úspěšný útok na tyto podpůrné služby, má to následky i pro VoIP technologii.

Hardwarové VoIP zařízení (telefony, ústředny, brány) jsou embedded zařízení, na kterých běží operační systém spolu se softwarem zajišťujícím požadovanou funkcionalitu VoIP služeb. Proveden může být útok i na samotný operační systém zařízení nebo serveru, případně na službu, která je na něm spuštěna. Jakákoli bezpečnostní chyba, která dovolí nahrání a spuštění škodlivého kódu, získání přístupu na dané zařízení nebo dokonce získání přístupu s oprávněním správce má důsledky pro všechny provozované služby včetně VoIP.

V další části se budu zabývat popisem jednotlivých typů útoků, které jsou cíleny na protokol SIP:

Odepření služby – (Distributed) Denial of Service

Pomocí zahlcení cílového zařízení (telefon, ústředna, brána, služba serveru) mnoha požadavky způsobí útočník prodloužení času reakce na požadavky až nedostupnost služby. Do útoku může být zapojeno více zdrojů (Distributed Denial of Service – DDoS), zahlcení cíle se provádí pomocí mnoha SIP požadavků v krátkém časovém úseku. Jedná se většinou o požadavky typu INVITE, REGISTER a OPTIONS.

Podvržení identity

Útoky typu podvržení identity lze provést úpravou samotného SIP požadavku. Výsledkem může být změna identifikace volajícího. Úpravu požadavku lze provést velmi jednoduše, pokud není používáno šifrování. Současně může být zfalšována zdrojová IP adresa (IP spoofing). Změnu zdrojové IP adresy může provést jakékoliv zařízení, přes které se přeposílá komunikace.

Odposlouchávání, úprava komunikace – Man in the Middle

Útočník dokáže odposlouchávat komunikaci (na síťovém zařízení) nebo dokáže přesměrovat komunikaci na své zařízení a následně přeposílat komunikaci požadovanému cíli. Komunikaci může také pozměnit.

Rušení spojení

Útočník může rušit spojení (hovor) zasíláním BYE požadavků. Musí však zjistit nebo odhadnout obsah *Call-ID*, *From*, *To* hlaviček a správně směrovat požadavek k požadovanému cíli.

Druhým typem je zasílání REGISTER požadavku s položkou „Expire=0“, tato zpráva je obvykle zasílána při vypínání telefonu. Útočník tímto způsobí, že uživatel nemůže přijímat ani vytvářet žádné hovory.

Odhalení jednoduchých hesel

Hesla, která jsou příliš krátká nebo nedosahují dostatečné bezpečnosti (neobsahují velká a malá písmena, číslice a speciální znaky) lze rychle odhalit hrubou silou. Znalost hesla pro přístup do konfiguračních rozhraní ústředen, proxy serverů nebo síťových zařízeních umožňují libovolné přenastavení a neomezené možnosti v provádění různých typů útoků.

Využití bezpečnostních chyb v softwaru

V případě, že VoIP telefon má integrovaný webový server (běžně používaný pro administraci telefonu) a žádný firewall nám neblokuje připojení na něj, může být proveden útok pomocí zranitelnosti ve webovém serveru. Některá zařízení dokonce umožňují stahování samotné konfigurace (včetně hesel v otevřené podobě) z ústředny pomocí Trivial File Transfer Protocol (TFTP) – stačí zjistit název konfiguračního souboru a stáhnout ho. Uhodnutí názvu konfiguračního souboru ulehčuje skutečnost, že název je většinou odvozen z Media Access Control (MAC) adresy telefonu. [26]

Některá administrační rozhraní VoIP telefonů jsou dokonce přímo přístupné z internetu, pro jejich nalezení lze dokonce využít běžný internetový vyhledávač, např. Google.com. Díky tomu lze velmi snadno zjistit mnoho informací (IP, porty, nastavení), které jdou využít pro následné útoky. [46]

Zasílání záměrně poškozených zpráv – tedy zpráv, které nejsou v souladu se specifikací SIP protokolu, může v případě chyby v softwaru rovněž vyvolat bezpečnostní problém – např. přetečení zásobníku. Tyto chyby mohou vést k nedostupnosti služby, případně i ke spuštění vlastního škodlivého kódu na daném zařízení. Další metodou může být obnovení VoIP zařízení do továrního nastavení a následné přehrání vlastním firmwarem. [32]

Pro zjištění typu operačního systému a softwaru používaného na vzdáleném zařízení včetně jeho verze lze využít následujících možností:

- *zachycení hlavičky User-Agent* – při generování SIP požadavku se většinou vyplní i hlavička User-Agent, která obsahuje používaný software a většinou i jeho přesnou verzi

- *zaslání příkazu OPTIONS* – zařízení zašle odpověď obsahující podporované protokoly a kodeky, někdy však také zašle verzi používaného softwaru
- *protokol SNMP* – v případě, že není zabezpečen přístup k SNMP operacím, můžeme jednoduše zjistit model zařízení a software spolu s verzí, který je na něm spuštěn
- *hlasová schránka* - na základě přehrávané výchozí hlášky, případně i na samotném chování hlasové schránky je možné odhadovat použitý software

Neoprávněné volání do sítě PSTN

Ústředny v závislosti na nastavení povolí volání do veřejné telefonní sítě (PSTN), pokud klient přidá určenou předvolbu před samotné volané číslo. V některých případech je ústředna dokonce nastavena tak, že nevyžaduje ani autorizaci pro volání iniciované z důvěryhodné IP adresy. Za důvěryhodné mohou být považovány všechny IP adresy z lokální sítě. Útočník v tomto případě zjišťuje nastavení ústředny pomocí zkoušení volání na číslo s různou předvolbou.

Tato práce se zaměřuje především na tento typ útoku. Podrobně bude vysvětlen v kapitole 2.

Pokusy o útoky, které jsou uskutečňovány ve velmi malých intenzitách a v delších časových intervalech, aby nedošlo k neobvyklému nárůstu VoIP provozu v síti, jsou velmi špatně detekovatelné.

1.5 Zavedení šifrování ve VoIP telefonii

Řešení, které by snížilo možnosti útoků, je zavedení šifrování veškerého VoIP provozu, tedy signalizačních i multimediálních dat.

Pro zabezpečení SIP protokolu lze použít Secure/Multipurpose Internet Mail Extensions (S/MIME) nebo TLS spojení. VoIP zařízení (ústředny, proxy servery) však potřebují číst, měnit a přidávat hlavičky SIP protokolu, proto je TLS spojení realizováno vždy mezi dvěma sousedními VoIP zařízeními. Přijatá komunikace je tedy na každém VoIP zařízení dešifrována a následně je znovu zašifrována a novým TLS spojením zaslána dalšímu zařízení v cestě. Multimediální data lze zabezpečit pomocí Secure Real-time Transport Protocol (SRTP). [27]

Aktuálně však pro zavedení šifrování veškerého VoIP provozu existují tyto problémy:

- šifrování přináší vyšší náročnost na výkon pro všechna zařízení v síti, tím pádem dochází ke snížení maximálního počtu možných současně uskutečnitelných hovorů v síti a zvyšování zpoždění komunikace,

- velmi málo VoIP zařízení podporuje šifrování, obecně samotná implementace v softwaru není v době psaní této práce stále úspěšně dokončena a otestována, většinou by tedy bylo nutné v organizacích kompletně vyměnit veškeré VoIP telefony a ostatní zařízení, aby byla zaručena spolehlivá funkčnost celé VoIP telefonie se šifrováním v organizaci.

Zavedení šifrování také žádným způsobem neřeší Denial of Service (DoS) útoky.

1.6 Softwarové nástroje

Existují běžně dostupné softwarové nástroje, které automatizovaně naleznou VoIP zařízení v síti (telefon, proxy server, bránu, ústřednu) a zjistí používaný software. Dokážou generovat požadavky SIP protokolu a zjišťovat nastavení a přihlašovací údaje. Tyto nástroje lze využít pro penetrační testy, mohou je však využívat i útočníci k ilegálním účelům.

SIPVicious [10] je software, který lze spustit na jakémkoli systému podporující Python verze 2.6 nebo vyšší. Je jednoduchý na ovládání, lze ho využít v rámci vlastních skriptů a programů. Dokáže identifikovat aktivní telefonní čísla na VoIP ústředně.

Dalším jednoduchým nástrojem pro diagnostiku VoIP, který lze ovládat pomocí příkazové řádky je *Sipsak* (SIP Swiss Army Knife) [8]. Podporuje testy pro zahlcení cílového VoIP zařízení mnoha požadavky.

Pro provedení testů bezpečnosti SIP implementace pomocí možných zranitelností (obsahující např. chyby přetečení zásobníku) lze použít nástroj *SiVuS* (SiP Vulnerability Scanner) [11].

Software *SIPp* [9] je testovací nástroj a generátor SIP provozu, zobrazuje statistiky ohledně zasílaných a zpracovávaných požadavků. Může být využit pro simulaci tisíců UAC, kteří volají přes ústřednu, kterou potřebujeme otestovat.

1.7 Obecná doporučení pro zajištění bezpečnosti ve VoIP

Prvním krokem je zajištění základní bezpečnosti v samotné počítačové síti (používání firewallů, jejich vhodné nastavení, používání monitorovacích systémů), dále je vhodné VoIP provoz oddělit od ostatního provozu (např. pomocí Virtual Local Area Network – VLAN). Článek [27] dále popisuje vhodné použití VoIP tunelování (pomocí Virtual Private Network – VPN), pokud jsou účastníci hovoru umístěny každý v jiné privátní síti. Tato metoda také umožňuje vyhnout se potenciálním problémům s firewally v souvislosti s technologií VoIP.

Na všech VoIP zařízeních a ve VoIP softwaru je důležité používat bezpečná hesla, zároveň je potřeba vyvarovat se použití speciálních znaků, které nelze zadat na hardwarových VoIP telefonech. Stejně důležitá je pravidelná kontrola a aktualizace používaného softwaru v zařízeních včetně operačních systémů z důvodu oprav nalezených bezpečnostních chyb. Vhodným doporučením je také vypnutí všech nepoužívaných služeb.

Jednou z nejdůležitějších částí pro zajištění bezpečnosti ve VoIP telefonii je správné nastavení používaných VoIP ústředen, proxy serverů a bran. Nutností je správně porozumět jednotlivým volbám v nastavení a vždy přečíst dokumentaci k používanému VoIP systému, protože různé typy zařízení a softwaru se mohou chovat odlišným způsobem. Po nastavení by měly následovat testy, které ověří požadovanou funkčnost. Umožnění volání určitým zařízením (uživatelům) v síti bez jakékoli vyžadované autentizace by mělo být v rámci sítě využíváno co nejméně.

Kapitola 2

Monitorování a analýza detekce zneužití VoIP ústředen

Hlavní snahou útočníků je získat finanční prospěch. Ve VoIP je jejich cílem úspěšné navázání mezinárodního nebo vnitrostátního hovoru přes ústřednu, která dokáže volat do veřejné telefonní sítě PSTN. Nejvýhodnějším útokem je dosáhnout volání na telefonní čísla se zvýšenou sazbou. Není podstatné, zda ústředna volá do PSTN přímo pomocí své brány či zprostředkovaně díky službě jiné ústředny. Dokonce mohou útočníci volání dále přeprodávat a získávat tímto finance. Útok se snaží provést takovým způsobem, aby nemohl být jednoduše identifikovatelný, proto jsou pokusy o volání uskutečňovány ve velmi malých intenzitách (řádově jednotky pokusů) v delších časových intervalech (např. po 10 minutách, případně i v hodinových intervalech).

Útočníci mohou využívat jako zdroj útoků i lokální IT prostředky organizace, na které mají vzdálený přístup díky infiltraci malwaru nebo jiného škodlivého softwaru. Při konfigurování ústředen je často předpokládáno, že rozsahy zdrojových IP adres dané organizace jsou důvěryhodné. Díky těmto skutečnostem je nutné předpokládat, že každý účastník ve VoIP může být potenciální útočník.

Pro volání do sítě PSTN je většinou nutné přidat předvolbu před samotné telefonní číslo. Pomocí předvolby telefonní ústředna zjistí, zda se jedná o požadavek na volání v rámci sítě VoIP nebo do sítě PSTN. Způsob zpracovávání požadavku a následné povolení volání do PSTN závisí na nastavení ústředny, případně na nastavení oddělené brány do PSTN. Předvolby jsou většinou nastavovány jako numerické, také pro jejich jednoduché zadávání na hardwarových VoIP telefonech.

Cíl práce

Cílem této práce je vyvinout modul, který má být schopný detekovat zneužití nedostatečně zabezpečených telefonních VoIP ústředen na počítačové síti. Měl by odhalit a následně nahlásit snahu útočníků o vytvoření telefonního hovoru z počítačové sítě na telefonní číslo do sítě PSTN. Schéma, jak takový útok může probíhat, je znázorněno na obrázku 2.1.

Postup útočníka je následující:

1. Založení telefonního čísla, které je určeno pro speciální placené služby. Volání na toto číslo je zpoplatněno zvýšeným tarifem a PSTN operátor zasílá vyúčtované prostředky pronajímateli telefonního čísla, tedy útočníkovi.

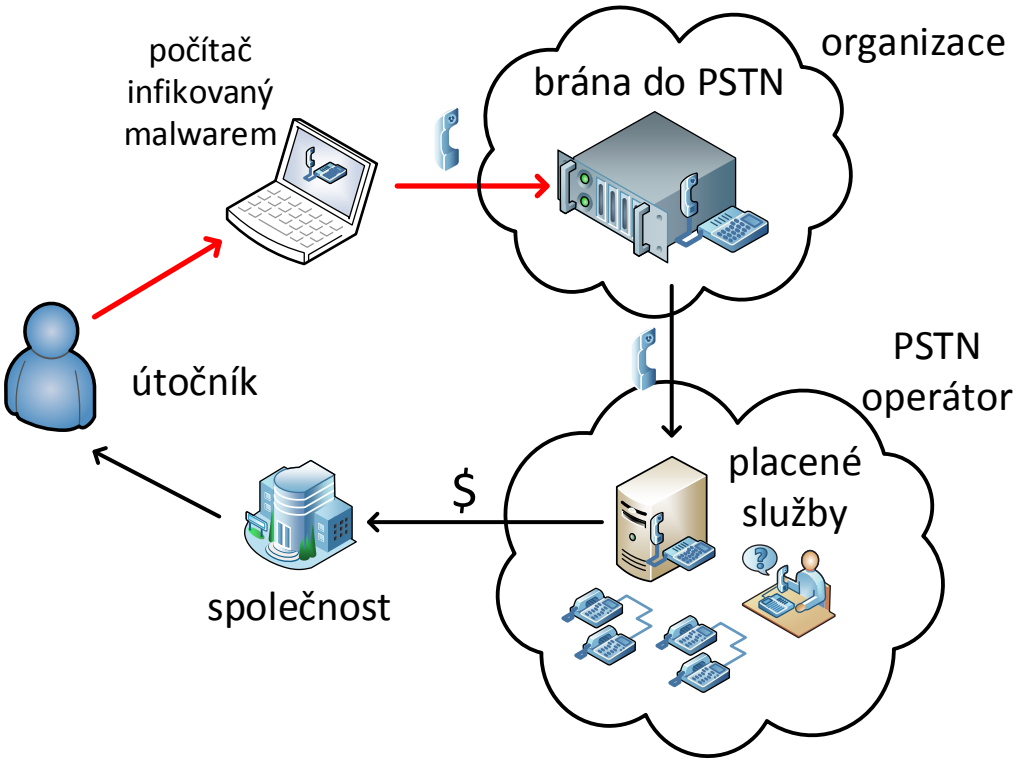
2. Ovládnutí počítačů v organizacích např. pomocí malwaru.

3. Díky ovládaným počítačům, vytvoří útočník požadavek v rámci dané organizace na lokální VoIP ústřednu, která pomocí brány do PSTN vytvoří telefonní hovor na útočníkem založené telefonní číslo. Platbu za volání na toto číslo hradí daná organizace.

Za účelem skrytí totožnosti útočníka, může být vytvořena nová společnost, která má za cíl pouze přeposílat finance útočníkovi. Aby byla ztížena možnost odhalení volání na telefonní čísla se zvýšenou sazbou, je telefonní číslo zakládáno u PSTN operátora v zahraničí.

Před odesláním požadavku na lokální VoIP ústřednu je nutné nejdříve najít její IP adresu. Následně je nutné nalézt předvolbu, která umožňuje volání do sítě PSTN. Pro úspěšné vytvoření hovoru je potřebné, aby nastavení ústředny umožňovalo volání z daného infikovaného počítače a aby nebyla požadována autorizace nebo aby útočník zjistil potřebné přihlašovací údaje.

Cílem těchto útoků mohou být specializovaná VoIP zařízení: ústředny, brány do PSTN nebo jejich ekvivalenty v podobě softwaru, který běží jako služba na serveru.



Obrázek 2.1: Schéma možného útoku ve VoIP

2.1 Monitorovací sonda

Monitorovací sonda je hardware, který monitoruje provoz na počítačové síti a exportuje tyto informace většinou ve formátu NetFlow nebo IPFIX.

NetFlow

NetFlow je protokol vyvinutý a patentovaný společností Cisco Systems v roce 1996. Poskytuje detailní pohled na síťovou komunikaci pomocí tzv. *IP toků* (flows). IP tok představuje sekvenci paketů, u kterých odpovídá zdrojová a cílová IP adresa, port, typ protokolu, QoS a vstupní síťové rozhraní. NetFlow záznamy mohou využívat např. správci sítí pro monitorování využívaných služeb, detekování útoků, účtování počtu přenesených dat či statistické informace. [23]

IPFIX

Internet Protocol Flow Information eXport (IPFIX) je Internet Engineering Task Force (IETF) standard a rovněž jako NetFlow je určen pro monitorování síťové komunikace pomocí toků. Byl vytvořen jako univerzální standard, který vychází z NetFlow a je více flexibilní. Rovněž definuje, jaký je formát zasílaných zpráv a jakým způsobem se samotná data odesílají. Více informací o IPFIX lze nalézt v RFC 7011 [24].

Monitorovací sonda nijak nezasahuje do sledované komunikace. Exportovaná data jsou zasílána pomocí jiné než sledované linky a nezatěžuje tedy žádným způsobem samotnou komunikační linku. Není nikdy viditelná na 2. ani 3. vrstvě referenčního modelu ISO/OSI a může být připojena kdekoliv v síti. Množství sond v síti není teoreticky nijak omezen, zapojují se většinou do mirror portu přepínače (případně směrovače). Možné provedení útoku v počítačové síti vůči samotné monitorovací sondě je velmi obtížně proveditelné. [30]

Poznámka: Sdružení CESNET vyvinulo monitorovací karty COMBO s rozhraním PCI Express, které jsou využitelné pro monitorování optických linek v reálném čase s propustností až 100 Gbps. Karty jsou založeny na technologii FPGA [20]. V síti CESNET2 se využívá tento typ monitorovacích karet na hraničních linkách.

2.1.1 VoIP plugin

Pro úspěšné monitorování některých typů komunikací je vyžadováno zpracovávání a zasílání dalších specifických informací ze síťové komunikace. Pro VoIP to znamená číst obsah IP paketu (payload), získat SIP hlavičky a informace z protokolů RTP a RTCP. V rámci této práce bude využíván rozšiřující plugin, který byl implementován v diplomové práci Ing. Jana Vojtěcha [45] pro produkt FlowMon Exportér společnosti INVEA-TECH [31]. Uvedený plugin umožňuje exportovat informace o VoIP komunikaci pomocí IPFIX do kolektoru. Kolektor označuje zařízení, které shromažďuje informace o zaznamenané komunikaci ze všech monitorovacích sond. Na straně kolektoru musí být zajištěna podpora pro zasílané rozšiřující položky. Použitý plugin zajišťuje identifikaci SIP komunikace na libovolném portu a SIP požadavků, dále páruje jejich odpovědi a výstupem je ucelená informace o proběhlém hovoru.

2.2 Nemea framework

Kapitola čerpá ze zdroje [16].

Nemea[1] (NEtwork MEasurement Analysis) je framework umožňující vytvořit systém pro monitorování a analýzu síťového provozu, byl navržen a vyvinut pro proudové zpracování dat v reálném čase. Tímto se liší od tradičního přístupu podobných systémů, které nejprve získávají, shromažďují a ukládají veškeré informace do společného datového úložiště a až následně v určitých intervalech provádějí samotnou analýzu provozu nad uloženými daty. Např. kolektor NfSen podporuje pluginy pro analýzu nasbíraných dat v 5 minutových intervalech. Nemea, pokud to z nějakého důvodu přímo nevyžadujeme, neukládá sledovaný provoz a ihned ho zpracovává. Vstupní data z počítačové sítě jsou očekávána ve formě IP toků (NetFlow nebo IPFIX) z kolektoru.

Z hlediska struktury se celý systém vytvořený Nemea frameworkem skládá z tzv. modulů. Každý modul je samostatně běžící proces v operačním systému určený pro plnění dílčí funkce systému. Moduly tedy mohou přistupovat k lokálním prostředkům daného stroje, mají možnost zapisovat na disk, komunikovat přes síťová rozhraní a využívat připojená zařízení, stejně jako jakýkoli jiný proces v operačním systému. Zatížení procesoru a spotřebu paměti jednotlivými moduly lze také zjišťovat pomocí běžných prostředků operačního systému. Modulární struktura umožňuje jednoduché spouštění a ukončování každého modulu, to je výhodné zvláště při aktualizaci modulu na novou verzi nebo přenastavení modulu vyžadující jeho restart, kdy není nutné restartovat celý systém, ale stačí daný modul vypnout, aktualizovat resp. přenastavit a znovu zapnout. Stejně tak nefunkční modul nezpůsobí pád celého systému. Díky uvedené struktuře lze systém také dobře škálovat při větším množství zpracovávaných dat nebo při náročnějších modulech. Využít lze jak horizontální škálování, kdy se systém rozšiřuje na větší počet PC nebo serverů, tak vertikální, při kterém lze přidáním procesorů, paměti, případně disků zvýšit výkon celého systému.

Více informací lze dohledat v technické zprávě [16] nebo na webových stránkách projektu [17], včetně zdrojových kódů a postupu instalace.

Knihovna TRAP

Komunikace mezi Nemea moduly je zajištěna knihovnou Traffic Analysis Platform (TRAP), která poskytuje rozhraní pro jednosměrnou komunikaci s ostatními moduly přes TCP nebo UNIX sokety. V případě chyby při komunikaci

[1] V době psaní diplomové práce byl Nemea framework ve verzi 2.0.1. Nemea je funkční na většině Linuxových distribucí. Vývoj a produkční prostředí je provozováno v operačním systému Scientific Linux 6.5 (Carbon).

zabezpečuje její automatické zotavení. Při návrhu knihovny byl kladen velký důraz na vysokou propustnost.

Datový formát UniRec

Předávaná data mezi Nemea moduly (pomocí knihovny TRAP) mohou být obecně libovolná, téměř výhradně se však používá speciální formát dat Unified Record (UniRec). Tento formát je paměťově úsporný, zjednodušeně ho lze popsat jako C strukturu definovanou za běhu modulu. Každý UniRec záznam se skládá z několika položek (fields) a jednotlivé položky mohou být i proměnné délky. Množina položek se označuje jako *šablona*. Dva komunikující moduly musí používat stejnou šablonu, musí tedy mít shodný UniRec formát. Šablona vstupních a výstupních rozhraní modulu je definována při jeho inicializaci. UniRec formát přináší do systému Nemea jednoduchý přístup k jednotlivým přenášeným položkám bez nutnosti parsování.

Nemea moduly

Většina modulů je implementována v programovacím jazyku C nebo C++. Důraz je kladen především na rychlost a nízkou spotřebu paměti. Modul však lze implementovat v libovolném programovacím jazyku, pouze se musí zajistit podpora pro TRAP knihovnu, aby byla funkční komunikace mezi jednotlivými moduly. Každý modul má určený počet (0 ... N) vstupních a výstupních rozhraní.

Obecně moduly provádí tyto kroky:

1. načtení dat ze vstupních rozhraní *
2. provedení dané akce (např. spočítání a případně uložení statistiky, detekování určitého typu síťového útoku atp.)
3. zapsání dat do výstupních rozhraní *

* 1. a 3. krok je závislý na počtu vstupních a výstupních rozhraní modulu.

Zjištěné útoky se výstupním rozhraním zasílají do modulů, které výsledky detekce mohou agregovat a dále hlásí nalezený útok emailem, ukládají do databáze či zasílají zprávu jinému systému pro vyhodnocení bezpečnostních událostí.

Nemea obsahuje základní moduly zajišťující import dat, následnou filtraci, spojování dat a export. Další pokročilé moduly jsou určeny pro detekování různých typů síťových útoků nebo k tvorbě statistik. Nemea je v době psaní práce stále ve vývoji, počet modulů se postupně zvyšuje a také funkčnost rozšiřuje.

2.3 Infrastruktura monitorovacího systému

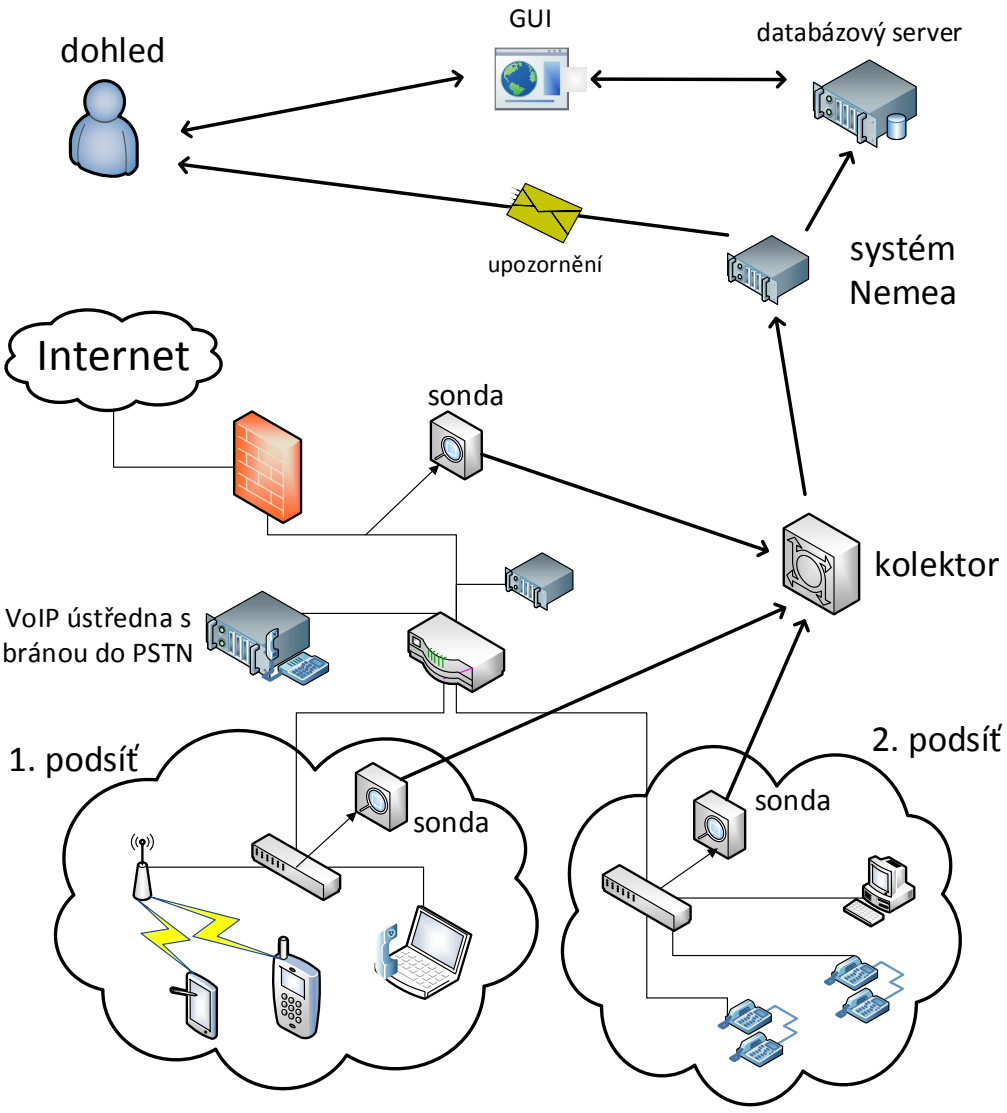
Monitorovací sondy zasílají záznamy o zachycené komunikaci do kolektoru. Kolektor tyto záznamy shromažďuje ze všech monitorovacích sond. Pomocí těchto dat lze počítat různé statistiky, ukládat vybrané informace, provádět vizualizace a mnoho dalších akcí jako detekování útoků či odesílat tyto informace do další systémů.

Monitorovací systém se tedy skládá z dílčích částí. Pro správnou funkci celého monitorovacího systému je vyžadován spolehlivý běh všech jeho součástí.

Ukázku, jak může být nasazen monitorovací systém v počítačové síti, zobrazuje schéma 2.2. Schéma popisuje počítačovou síť organizace se dvěma lokálními podsítěmi, ve kterých jsou připojeny stolní počítače, VoIP telefony, notebooky a další přenosná zařízení zaměstnanců (mobilní telefony a tablety připojené bezdrátově). V každé této podsíti je instalována monitorovací sonda, která je připojena do mirror portu přepínače. Mimo uvedené dvě podsítě je umístěn server poskytující potřebné služby (DNS, DHCP, emailový server, webový server), VoIP ústředna s bránou do sítě PSTN a centrální směrovač. Organizace je připojena k internetu a přístup do její sítě je chráněn firewallem. Na lince, která připojuje celou síť k internetu, je umístěna další monitorovací sonda.

Záznamy o zachycené komunikaci jsou zasílány z monitorovacích sond oddělenou sítí do kolektoru. V této oddělené síti jsou pouze zařízení určené pro monitoring a správu sítě, přístup k této síti je běžným uživatelům zakázán. Systém Nemea proudově zpracovává tyto informace z kolektoru a nalezené bezpečnostní incidenty zasílá přímo elektronickou poštou dohledovým pracovníkům, zároveň jsou informace ukládány do databáze. Záznamy z databáze lze prohlížet, filtrovat a vytvářet statistiky díky webovému rozhraní.

VoIP komunikace je v rámci sítě organizace realizována prostřednictvím hardwarových VoIP telefonů, softwarových klientů v počítačích a VoIP ústřednou s bránou do PSTN. Veškeré požadavky na volání do externích sítí musí být zahájeny pomocí VoIP ústředny.



Obrázek 2.2: Ukázkové schéma síťové infrastruktury v organizaci

2.4 Seznámení s testovacím prostředím

Modul byl testován na reálných datech z produkční instance systému Nemea nasazeného na síti CESNET2, kde je komunikace sledována na hraničních linkách. Vysokorychlostní síť CESNET2 propojuje vysoké školy a univerzity, výzkumné a zdravotnické organizace. Jedná se o National Research and Education Network (NREN), využívána je pro akademické, vědecké a výzkumné účely. Operátorem této sítě je CESNET z.s.p.o. a tato síť je zdrojem velkého množství informací o síťové komunikaci. [19]

2.4.1 Prvotní sledování SIP provozu v reálné síti

Jako první krok před samotným návrhem modulu pro detekci zneužití VoIP ústředen jsem vytvořil jednoduchý modul pro systém Nemea, který zaznamenával SIP provoz a ověřoval funkčnost zasílání potřebných informací z monitorovacích sond v rámci testovacího prostředí. Díky tomuto kroku jsem zjistil, jak velké množství SIP komunikace se nachází v reálné síti a byl jsem schopen dále navrhovat modul tak, aby byl vhodný i pro nasazení právě do tohoto prostředí.

2.4.2 Chyba v určování komunikace VoIP pluginem

Během testování jednoduchého modulu pro základní ověření funkčnosti systému Nemea, jsem nalezl chybu ve VoIP pluginu (popsaném v sekci 2.1.1), která způsobuje zasílání prázdných resp. neplatných SIP hlaviček. Chyba pravděpodobně spočívá v nedůsledné identifikaci SIP komunikace. Plugin určí typ komunikace na základě obsahu paketu (payload), kde zjednodušeně řečeno vyhledává klíčová slova, která se vyskytují v požadavcích a odpovědích SIP protokolu. Použitá metoda však v některých případech selhává a plugin chybně považuje komunikaci za SIP, i když je to ve skutečnosti komunikace pomocí jiného protokolu. Proto následně v obsahu komunikace nelze najít očekávané hlavičky protokolu SIP. Na základě měření bylo možné získat minimální datovou sadu způsobující popsané potíže. Chyba byla následně nahlášena autorům k opravě.

2.5 Navržené metody detekce

Hlavním principem mnoha útoků, které dokážou generovat i běžně dostupné softwarové nástroje (viz sekce 1.6) je zasílání velkého množství INVITE, REGISTER, případně OPTION požadavků, které mají za cíl buď odhalit přihlašovací jména a hesla hrubou silou nebo zahltit cílovou ústřednu mnoha požadavky, případně odhalit nastavenou předvolbu pro volání do PSTN.

Tento typ útoku, kdy dojde k intenzivnímu nárůstu počtu např. INVITE požadavků, které přicházejí na ústřednu v několika jednotkách, někdy až ve

stovkách za sekundu, je relativně snadno identifikovatelný a útočníka lze odhalit pomocí běžných detekčních prostředků. V systému Nemea může být detekován díky modulům, které sledují počty jednotlivých toků v síti. DoS útoky lze identifikovat stejnými nástroji. DoS útok lze však zjistit i nepřímo, např. díky nástrojům běžně instalovaných na serverech pro monitorování celkového zatížení systému, které budou hlásit vysoké zatížení procesoru, velké využití paměti nebo neobvyklé zvýšení datové komunikace na síťových rozhraních. Obrana může být provedena např. zakázáním dané IP adresy ve firewallu, kterou lze provádět automaticky[2].

Cílem této práce je však zaměřit se na útoky, které jsou špatně identifikovatelné běžnými nástroji, a které mohou způsobit finanční ztrátu. Snahou je vytvořit modul takovým způsobem, aby byl schopný zachytit útok nezávisle na tom, zda VoIP ústředna vyžaduje autentizaci klienta (uživatele). Nemělo by být také podstatné, zda IP adresa uvedená v toku jako zdrojová pro požadavky, respektive cílová pro odpovědi patří klientovi ústředny nebo samotné ústředně.

Po důkladné analýze možností detekce zneužití VoIP ústředen jsem navrhl dvě nezávislé metody, na základě kterých by měl vyvinutý modul detekovat útok:

- detekování pokusů o nalezení nastavené předvolby pro volání do PSTN, včetně identifikace, zda došlo k úspěšnému zahájení hovoru v rámci zkoušených čísel,
- neobvyklé volání na cílový prvek v síti (např. ústřednu) umístěný v jiné zemi než je obvyklé volání pro daný zdroj.

[2]Pokud bude prováděno automatické zakázání komunikace z dané IP adresy, bude služba ústředny resp. serveru nedostupná pro danou IP adresu, která může představovat velké množství koncových uživatelů. Je tedy nutné pečlivě zvážit tento způsob řešení útoku.

Kapitola 3

Návrh detekčního modulu

V této kapitole se budu zabývat samotným návrhem modulu pro detekování zneužití VoIP ústředen. Modul je navrhován pro použití v Nemea frameworku a pro nasazení v počítačové síti. Modul bude provádět tyto činnosti:

- přijímat zachycenou komunikaci SIP protokolu z monitorovaných sítí,
- zpracovávat a ukládat přijaté informace,
- pravidelně kontrolovat, zda nedošlo k útoku,
- v případě útoku nahlásit tuto informaci.

Postupně budu popisovat uvedené činnosti z hlediska návrhu modulu.

3.1 Přijímání zachycené komunikace z monitorovaných sítí

Pro zajištění úspěšné detekce zneužití VoIP, resp. SIP ústředen je potřeba monitorovat požadavky a odpovědi SIP protokolu, včetně informací o zdrojové a cílové IP adrese toku, ve kterém se daný provoz nacházel. Pro zachycení veškerého provozu je vhodné neomezovat se na standardně používaný port protokolu ani pouze na protokol UDP nebo pouze TCP, ale monitorovat veškerý provoz nezávisle na zdrojovém a cílovém portu.

Pro zjištění informací o telefonním hovoru je potřeba sledovat:

- INVITE požadavky – zahájení nového hovoru,
- `200 OK` odpovědi (a ACK požadavky) – zjištění úspěšného navázání hovoru.

Pro získání více informací o probíhajících útocích budou sledovány:

- ACK, CANCEL, BYE požadavky,
- `180 RINGING`, `100 TRYING`, `401 UNAUTHORIZED`, `403 FORBIDDEN`, `407 PROXY AUTH REQUEST` odpovědi.

Úspěšné zahájení hovoru nastane až po zaslání ACK požadavku volajícím klientem. Proto by měla detekce úspěšně zahájeného hovoru kontrolovat i ACK požadavky. Během testování modulu však bylo zjištěno, že velmi často nastává situace, kdy není zachycen tento požadavek, protože byl v síti směrován jinou cestou (linkou), na které není instalována monitorovací sonda. Požadavek ACK je totiž zasílán přímo koncovému klientovi a nemusí procházet stejnou cestou jako INVITE požadavek (tedy přes linku, kde je SIP proxy server). Stejný problém je u sledování BYE příkazu. Z tohoto důvodu je za úspěšně navázaný hovor považováno přijmutí `200 OK` odpovědi na INVITE požadavek, který je zasílán stejnou cestou (pouze opačným směrem). Modul však umí jednoduše přepnout do módu, kdy považuje úspěšně navázaný hovor až po přijetí ACK požadavku, stačí provést změnu v konfiguračním souboru modulu (`configuration.h`).

V rámci každého požadavku nebo odpovědi je nutné sledovat tyto hlavičky protokolu:

- *To* – zjištění volaného čísla a současně pro identifikaci daného hovoru
- *Request-URI* – zjištění směrování požadavku
  (teoreticky může nastat situace, kdy bude útočníkem vyplňováno odlišné volané číslo v Request-URI a To hlavičce)
- *Call-ID* – pro identifikaci daného hovoru
- *User-Agent* – informace o použitém softwaru nebo zařízení, které vygenerovalo požadavek
- *From* – pomocná informace, určená pouze pro informativní účely (útočníci mohou vyplňovat libovolné hodnoty)

Pro další informativní účely budou sledována RTP data, bude ukládána informace, zda byla RTP data k danému hovoru zachycena nebo nikoliv.

Požadované položky ze SIP protokolu nám zajistí VoIP plugin, popsaný v sekci 2.1.1. Moduly v systému Nemea mohou díky němu přijímat informace o SIP provozu včetně rozparsovaných hlaviček protokolu pomocí definovaných položek datového formátu UniRec. Plugin bylo ale nutné rozšířit o zasílání *Request-URI* a *User-Agent* hlavičky. Po detailním seznámení se zdrojovým

kódem VoIP pluginu, jsem implementoval rozšíření o uvedené položky a díky kolegům z projektu Liberouter [3] byla upravená verze nasazena na měřící body v síti CESNET2.

SIP komunikace může být zaznamenána, pokud není použito šifrování signalizačních dat (např. pomocí TLS spojení). Pokud se použije šifrování, nelze zjistit typ SIP zprávy ani obsah jejich hlaviček bez znalosti klíče pro dešifrování.

3.2 Zpracování a ukládání přijatých informací

Pro každou monitorovanou IP adresu musí být ukládány informace o požadavcích na volání ve vhodné datové struktuře, aby mohly být prováděny rychlé detekce útoků. Musí být možno v paměti detekovat pokusy na volání stejných čísel, ale s různými předvolbami. Počet předvoleb zkoušených útočníky musí být zjistitelný v co nejkratším možném čase. Dále musí být podporována možnost smazání volaných čísel, vztahujících se k danému útoku, aby se po úspěšné detekci útoku nemusely mazat všechny informace k dané IP adrese. Zároveň seznam IP adres musí být uložen v takové formě, aby bylo možné rychle zjistit, zda je již uložena v paměti a je sledována modulem.

Každá IP adresa bude mít dále uložené informace o seznamu zemí, do kterých jsou uskutečňovány hovory. Pokud bude úspěšně navázán hovor na cílovou ústřednu, která se nachází v jiné zemi, bude nahlášeno upozornění. Cílová země bude zjištěna pomocí domény (IP adresy) z části *host* ze SIP URI volaného. Modul bude podporovat učící režim po časově definovanou dobu, při kterém budou zaznamenávány země, do kterých dané IP adresy volají. Seznam zemí musí být při ukončování modulu uložen a při jeho opětovném zapnutí obnoven do původní podoby.

Jako datovou strukturu pro ukládání informací o sledovaných IP adresách jsem vybral hašovací tabulku využívající tzv. kukaččí hašování (Cuckoo hashing), které umožňuje efektivně využívat alokovanou paměť. Pro ukládání informací o volaných číslech pro danou IP adresu jsem zvolil suffixový strom.

Hašovací tabulka s kukaččím hašováním (Cuckoo hash table)

Na rozdíl od tradiční hašovací tabulky, kde je používána pouze jedna funkce pro nalezení cílového umístění podle klíče, je v kukaččím hašování [37] využíváno více funkcí, což zajišťuje vyšší zaplnění tabulky. Některé implementace využívají více hašovacích tabulek, kdy položka může být umístěna pouze v jedné z tabulek. Nalezení a smazání položky v tabulce je v nejhorším případě

[3] Projekt Liberouter je výzkumným projektem oddělení nástrojů pro monitorování a konfiguraci sítě sdružení CESNET.

realizováno v konstantním čase, je nutné zkontrolovat tolik položek, kolik je použito tabulek resp. hašovacích funkcí. Při vkládání nové položky může nastat situace, že místo v tabulce je již obsazeno, v tomto případě je původně uložená položka vyjmuta a vložena na alternativní pozici. Pokud alternativní pozice není volná, je opět vyjmuta původně uložená položka a celý proces se opakuje. V závislosti na implementaci, pokud počet opakování překročí definovanou hodnotu, je provedeno znovu vytvoření tabulky s novou hašovací funkcí nebo může být položka z alternativní pozice odstraněna.

Uvedená datová struktura je součástí Nemea frameworku, v implementaci je využíváno metody pouze s jednou alokovanou tabulkou a s použitím celkem 3 hašovacích funkcí. Pokud při vkládání nové položky dojde k překročení hodnoty 15 v počtu opakování při přesouvání položek z alternativních pozic, je položka z tabulky odstraněna.

Suffixový strom

Suffixový strom umožní ukládat volaná čísla do stromové struktury. Každý uzel označuje část volaného čísla a má uloženo, kolik uzlů existuje v jeho podstromu. To umožňuje rychlé zjištění počtu různých předvoleb pro daný suffix. Bude využita implementace, která vznikla v rámci bakalářské práce Bc. Zdeňka Rosy [39]. Aby mohla být využita v modulu pro detekci zneužití VoIP ústředen, přidal její autor podporu pro určení typu vytvářeného stromu na prefixový nebo suffixový a dále rozšířil funkce o možnost ukládání dalších dodatečných dat do každého uzlu stromu a možnost mazání určitého podstromu. V rámci vývoje byly odstraněny nalezené chyby, ke kterým došlo v rámci úprav a rozšiřování funkcí suffixového stromu.

3.3 Programovací jazyk

Detekční modul musí být navržen tak, aby byl schopný fungovat i v tak velké síti, jakou je síť CESNET2. Jedním z důležitých parametrů vytvořeného modulu je tedy úspornost z hlediska spotřebovávané paměti i procesorového času. Pro implementaci modulu jsem zvolil programovací jazyk C díky vhodné integraci se systémem Nemea a možnosti kompletní správy paměti.

3.4 Geografická lokalizace IP adresy

Pro ukládání a detekování volání do jiných zemí než je obvyklé je potřeba zajistit určení geografického umístění IP adresy. Přesné lokalizování podle IP adresy není možné, ale pro určení země je metoda dostatečně spolehlivá. Pro nalezení země, ve které se nachází cílová IP adresa, je nutné využít databázi, která bude tyto informace schopna poskytnout. Využívat by šlo také metod

založených pouze na reverzních DNS dotazech, tyto metody by však byly mnohem více nespolehlivé. Při hledání vhodné databáze je velmi důležitá možnost její budoucí aktualizace.

Vybral jsem geolokační databázi společnosti MaxMind [34], která poskytuje i informace o anonymních proxy serverech a o satelitních připojeních, které jsou využívány ve více zemích současně. Firma poskytuje pro svou databázi API pro různé programovací jazyky včetně C. Nabízí i placené služby, které zahrnují obsáhlejší databáze s častějšími aktualizacemi, detailnější informace u IP adres jako je např. PSČ, typ internetového připojení, typ uživatele (kavárna, škola, vládní organizace, knihovna, CDN (Content Delivery Network) či poskytovatel hostingových služeb). Dále firma poskytuje možnosti detekce, zda daná IP adresa je anonymní nebo otevřený proxy server, případně VPN připojení včetně skóre vyjadřující pravděpodobnost možného podvodu z těchto adres.

Na webu společnosti [35] se uvádí, že více než 5 000 společností využívá GeoIP data pro lokalizaci návštěvníků, efektivní směrování internetového provozu a mnoho činností pro statistické a marketingové účely. Z nezávislé studie z roku 2011 [18] vyplývá, že databáze společnosti MaxMind se shodují se všemi srovnávanými databázemi z více než 90 %.

Databáze poskytovaná zdarma *GeoLite Country* je méně přesná, aktualizace databáze probíhají 1x měsíčně, ale pro využití modulem bude plně dostačovat.

Kapitola 4

Implementace detekčního modulu

Modul má v rámci systému Nemea definováno 1 vstupní a 1 výstupní rozhraní, které je obsluhováno knihovnou TRAP. Spuštěním modulu dochází nejprve k inicializaci této knihovny a pak k nastavení obsluhy SIGTERM a SIGINT signálů. Po načtení parametrů spouštěného programu dojde k jejich uložení do příslušných proměnných. Pokud není zadán volitelný parametr, je použita výchozí hodnota, která je definovaná v souboru configuration.h. Následně je načten `event_id` soubor, ve kterém je uložen číselný identifikátor naposledy detekovaného útoku.

Po úspěšné alokaci potřebných proměnných a vytvoření šablon pro datový formát UniRec vstupuje modul do nekonečné smyčky, ve které jsou prováděny veškeré funkce programu. Na začátku této smyčky je volána knihovní funkce `trap_recv()`, která zajišťuje načtení dat ze vstupního rozhraní modulu. Po úspěšném ověření přijatých dat je kontrolován typ přijaté zprávy. VoIP plugin (popsaný v sekci 2.1.1) rozděluje SIP požadavky na servisní a na požadavky týkající se hovoru, u každého uvedeného typu dochází ještě k rozdělení na samotné požadavky a jejich odpovědi. Celkem tedy VoIP plugin zasílá 4 typy zpráv. Detekční modul zpracovává požadavky a odpovědi týkající se hovoru a odpovědi na servisní požadavky. Pokud přijatá zpráva neodpovídá požadovanému typu, není s těmito daty jakkoliv pracováno a čeká se na další příchozí data.

Po přijetí dat je pomocí datového formátu UniRec načtena zdrojová a cílová IP adresa toku, *Request-URI*, *From*, *To*, *Call-ID* a *User-Agent* hlavička. Po načtení všech těchto informací dochází ke zpracování přijatých SIP URI pomocí funkce `cut_sip_identifier()`, která kontroluje platný formát, odstraňuje počáteční identifikátor SIP URI „sip:“, resp. „sips:“ a parametry uvedené na konci.

V případě, že je definováno makro `PRINT_DETAIL_INVALID_SIPURI` a přijatá data obsahují neplatnou hlavičku, je vypsána na výstup programu detailní

informace o neplatné hlavičce, včetně identifikace monitorované linky, ze které data přišla, zdrojového a cílového portu.

V následujících dvou podkapitolách bude vysvětleno, jak probíhá další zpracovávání přijatých informací z monitorovaných sítí v závislosti na metodě detekce útoků. Pokud je provedeno zahájení sledování komunikace z IP adresy jakoukoli metodou, jsou vytvořeny datové struktury pro obě uvedené metody.

4.1 Detekce pokusů o nalezení předvolby

Před dalším zpracováváním je prováděna kontrola, zda SIP URI volaného v SIP požadavku odpovídá telefonnímu číslu s povolenými speciálními znaky „+“, „*“, „#“, „-“ a „:“. Zároveň je kontrolována minimální délka volaného čísla, která je definována pomocí parametru programu (`-d`). SIP URI volaného je převzato z *To* hlavičky. Pokud je v souboru `configuration.h` definováno makro `CHECK_DIFFERENT_REQUEST_URI`, provádí se porovnávání volaného čísla z *To* hlavičky s *Request-URI*. V případě, že telefonní číslo nesouhlasí, je vypsána tato informace a výpis položek na standardní výstup a do log souboru.

Následná práce programu závisí na typu obdržené zprávy z VoIP pluginu, lze ji zjednodušeně popsat následujícím pseudokódem:

```
1: switch typ obdržené zprávy do
2:     case zpráva obsahující požadavky týkající se hovoru
3:         if zdrojová IP adresa není uložena v hašovací tabulce then
4:             if přijatá data neobsahují žádný INVITE požadavek then
5:                 return
6:             else
7:                 uložit IP adresu do hašovací tabulky (začátek sledování)
8:             end if
9:         end if
10:        if čas poslední kontroly+interval detekce > aktuální čas then
11:            if čas posledního útoku+pozastavení detekce > aktuální čas then
12:                kontrola útoku pro zdrojovou IP adresu
13:            end if
14:        end if
15:        if počet INVITE požadavků > 0 then
16:            uložit SIP URI volaného do suffixového stromu (pokud již nee-
    xistuje)
17:            uložit Call-ID z požadavku do vytvořeného uzlu stromu
18:        else
19:            if SIP URI volaného není uloženo v suffixovém stromě then
20:                return
```

```
21:             end if
22:             if Call-ID není uloženo v daném uzlu suffixového stromu then
23:                 return
24:             end if
25:         end if
26:         uložit INVITE, ACK, CANCEL a BYE požadavky
27:     case zpráva obsahující odpovědi týkající se hovoru
28:         if cílová IP adresa je uložena v hašovací tabulce then
29:             if SIP URI volaného je uloženo v suffixovém stromě then
30:                 if Call-ID je uloženo v daném uzlu suffixového stromu then
31:                     uložit OK, RINGING, TRYING,
32:                           PROXY AUTH REQUEST odpovědi
33:                 end if
34:             end if
35:         end if
36:     case zpráva obsahující odpovědi na servisní požadavky
37:         if cílová IP adresa je uložena v hašovací tabulce then
38:             if SIP URI volaného je uloženo v suffixovém stromě then
39:                 if Call-ID je uloženo v daném uzlu suffixového stromu then
40:                     uložit FORBIDDEN, UNAUTHORIZED odpovědi
41:                 end if
42:             end if
43:         end if
```

Pro každou *sledovanou IP adresu* je vytvářena struktura (`ip_item_t`) a inicializován suffixový strom. Sledování IP adresy začíná přijmutím INVITE požadavku. Informace ukládané ve struktuře jsou:

- ukazatel na suffixový strom,
- čas poslední komunikace IP adresy,
- čas poslední kontroly IP adresy na detekci pokusů o nalezení předvolby,
- čas prvního přijmutí INVITE požadavku z této IP adresy (= začátek sledování IP adresy),
- čas posledního útoku,
- volané číslo, které bylo vyhodnoceno v posledním útoku + délka zkoušené předvolby pro toto číslo,
- identifikátor posledního útoku (`event_id`),
- počet detekovaných událostí a počet detekovaných útoků.

Detekovanou událostí se rozumí zjištění útoku na dané zdrojové IP adrese, je vyhlašována vždy při nalezení útoku. Na rozdíl od detekovaného útoku, který není vyhlašován, pokud poslední detekovaná událost pochází ze stejné zdrojové IP adresy a pokus byl o volání na stejné číslo (bez zkoušené předvolby útočníky). Tedy pokud bude probíhat dlouhodobě útok z jedné IP adresy na volání stejného čísla (s jinými předvolbami) je tento útok opakovaně modulem vyhodnocován a je přičítán do počtu detekovaných událostí. V počtu detekovaných útoků bude však započítán pouze 1x.

Každý *uzel suffixové stromu* si mimo svého textového popisu ukládá i další informace (`node_data_t`):

- počet INVITE požadavků,
- počet OK odpovědí,
- počet ACK požadavků,
- počet CANCEL, BYE požadavků a TRYING, RINGING, FORBIDDEN, UNAUTHORIZED, PROXY AUTH REQUEST odpovědí pro detailnější statistiky,
- určení, zda byla zachycena RTP data – pokud byla zachycena data pro dané volané číslo (uzel), je hodnota v daném uzlu 1, v opačném případě 0,
- seznam haší Call-ID z požadavků na volání daného čísla (uloženo kvůli nutnosti párování odpovědí na požadavky). Z důvodu úspory spotřebované paměti je ukládán pouze haš z textové hodnoty *Call-ID*. Je využita funkce *SuperFastHash*, kterou implementoval Paul Hsieh a která dle uveřejněných testů vypočítá otisk velmi rychle [29]. Uvedená metoda není kryptografickou hašovací funkcí. Pro použití v modulu však není důležitá vlastnost jednosměrnosti a bezkoliznosti, prioritní je rychlost výpočtu.
- haš *User-Agent* hlavičky, která byla naposledy použita při volání na dané telefonní číslo. Modul si udržuje seznam všech použitých *User-Agent* hlaviček v samostatné hašovací tabulce. Uzel suffixového stromu má pak již uloženou pouze haš hodnotu (pro výpočet hodnoty je použita funkce *SuperFastHash*). Použitý přístup dále přispívá ke snížení spotřeby paměti.

Kontrola útoku pro zdrojovou IP adresu

Samotná detekce útoku z dané zdrojové IP adresy probíhá na základě tohoto algoritmu:

1. *Vzestupná fáze:*

   Volaná čísla jsou uložena v suffixovém stromě. Postupně je ze všech jeho listů provedeno nalezení předchůdce, který označuje číslici (číslice) volaného čísla umístěnou na pozici, jejíž vzdálenost je rovna nejvýše délce zkoumané předvolby (parametr `-m`). Vzdálenost pozice je počítána od prvního znaku volaného čísla (první znak je umístěn na pozici 1). Následně je provedeno zjištění počtu volaných čísel v podstromu, jehož kořen je rodič nalezeného předchůdce.

   Pokud je v průběhu nalezen počet volaných čísel v podstromu převyšující mez počtu unikátních předvoleb (parametr `-t`), pokračuje se bodem 2, v opačném případě algoritmus končí a nebyl detekován žádný útok.

2. *Sestupná fáze:*

   Po zjištění počtu volaných čísel v podstromu je spuštěno vyhodnocení pomocí funkce `prefix_examination_minus_detection()`. Z uvedeného počtu volaných čísel jsou odečteny uzly, které vyjadřují úspěšné zahájení alespoň jednoho hovoru na číslo určené daným uzlem. Dále jsou odečtena volaná čísla, která se liší volanou předvolbou o počet číslic větší než je hodnota parametru `-m`. Pokud po odečtení hodnot je stále převyšována mez počtu unikátních předvoleb, je detekován útok a pokračuje se bodem 3. Pokud útok nebyl nalezen, provede se návrat do bodu 1 a pokračuje se ve vykonávání algoritmu.

3. *Po nalezení útoku:*

   Pokud byl detekován útok, je provedeno vypsání informací o útoku a nahlášení do výstupního rozhraní modulu. Poté dojde k vymazání podstromu, na kterém byl detekován útok. Tímto algoritmus pro kontrolu útoku pro zdrojovou IP adresu končí.

   Informace o útoku obsahují:

   - identifikace útoku (`event_id`) – novému útoku je přiřazena identifikace, která je vypočtena přičtením 1 k hodnotě posledního útoku; pokud útok pokračuje, identifikace útoku je stejná,
   - datum a čas detekování útoku,
   - zdrojová IP adresa, ze které pochází útok,
   - datum a čas prvního zaznamenaného INVITE požadavku z dané IP adresy,

- jedno z volaných čísel, na které probíhal neúspěšný pokus o volání (vybíráno je číslo s nejdelší zkoušenou předvolbou) a uvedena je délka zkoušené předvolby pro toto číslo,
- *User-Agent* hlavička, která byla použita na volání čísla uvedeného v předchozím bodě,
- počet zkoušených telefonních čísel, na které probíhaly neúspěšné pokusy v rámci daného podstromu,
- počet úspěšně navázaných hovorů a celkový počet INVITE požadavků v rámci podstromu,
- další informace, které vyjadřují součty zaznamenaných dat v rámci podstromu:
  - zachycená RTP data
  - požadavky ACK, CANCEL, BYE
  - odpovědi OK, TRYING, RINGING, FORBIDDEN, UNAUTHORIZED, PROXY AUTH REQUEST
- celkový počet detekovaných událostí a útoků pro zdrojovou IP adresu.

V průběhu detekce útoku jsou ukládány do paměti (`cache_node_no_attack`) uzly stromu, na kterých nebyl detekován útok a tedy ani na jejichž následnících nemůže být detekován útok. Při vkládání nového uzlu do této paměti je zajištěno smazání všech následníků vkládaného uzlu. Uvedená paměť je před novým spuštěním algoritmu vždy smazána. Během zpracovávání se před spuštěním další části algoritmu kontroluje, zda daný uzel nebo jeho předchůdce již není uložen v paměti. Pokud je nalezen, není již dále prováděna kontrola na tomto uzlu a pokračuje se dalším v pořadí. Popsaná optimalizace snižuje výpočetní náročnost při zjišťování útoku.

4.2 Detekování volání do jiné země

Pokud je zahájen nový hovor, je provedeno zjištění cílové IP adresy, respektive domény z cílové SIP URI pomocí funkce `get_domain()`. Pomocí domény je následně vyhledána cílová země v GeoIP databázi. Po úspěšném nalezení země je provedena další akce podle aktuálního stavu modulu:

- *Modul je v učícím režimu:* Země jsou pouze ukládány do seznamu povolených zemí.
- *Modul je v režimu aktivní detekce:* Při zjištění volání do jiné země je nahlášeno upozornění.

Učící režim je spuštěn automaticky po startu modulu po dobu definovanou parametrem `-a`. Po této době se přepne do režimu aktivní detekce volání do jiných zemí. Při přepnutí režimu je tato informace zapsána na standardní výstup a do log souboru.

Pokud zdrojová IP adresa ještě není sledována a nejsou vytvořeny potřebné datové struktury, je zajištěno nejprve jejich vytvoření a inicializace. Pro detekování volání do jiné země obsahuje struktura `ip_item_t` pro IP adresu tyto položky:

- čas poslední detekce volání do jiné země,
- identifikátor poslední detekce volání do jiné země (`event_id`),
- země, do které bylo voláno při poslední detekci,
- seznam povolených zemí pro danou IP adresu,
- počet detekovaných volání do jiné země.

Úspěšně navázaný hovor

Zahájení nového hovoru je v rámci detekce volání do jiné země určováno odlišným způsobem v závislosti na obsahu *To* hlavičky:

1. Pokud *To* hlavička *obsahuje telefonní číslo*, které vyhovuje požadavkům pro detekci zkoušení předvoleb (minimální délka zkoumaného čísla a obsah pouze povolených znaků – viz sekce 4.1), je za úspěšně navázaný hovor považováno zaznamenání INVITE požadavku, následovaného OK odpovědí:
   - INVITE požadavek na volání daného telefonního čísla
   - následná `200 OK` odpověď, která odpovídá INVITE požadavku (souhlasí cílová IP adresa požadavku a *Call-ID* hlavička)
2. U *ostatního volání* (včetně volání na SIP URI obsahující text) je úspěšné navázání hovoru určeno pouze zaznamenáním `200 OK` odpovědi.

Uvedený způsob zjišťování zahájení nového hovoru umožňuje modulu neukládat velké množství informací o požadavcích na volání „textových“ SIP URI. Tímto opatřením se snižuje spotřebovávaná paměť a výpočetní náročnost.

Seznamy povolených zemí

Seznamy povolených zemí pro všechny IP adresy jsou ukládány do tzv. `countries` souboru. Při spouštění modulu jsou seznamy z tohoto souboru načteny do paměti, při ukončování modulu jsou z paměti uloženy do souboru. Pokud je definován v souboru `configuration.h` interval `COUNTRIES_FILE_SAVING_INTERVAL` jsou seznamy ukládány i za běhu modulu.

Po nahlášení upozornění o volání do jiné země je standardně tato země společně s IP adresou přidána do seznamu povolených zemí a při dalším uskutečněném volání do stejné země z dané IP adresy již není zasíláno upozornění. Pokud se požaduje, aby do seznamu nebyly automaticky přidávány nahlášené země, lze toto chování vyžádat použitím parametru `-w` při spouštění modulu. Pak budou opakovaně nahlašována upozornění a `countries` soubor nebude měněn.

V `countries` souboru je možné dále definovat *globálně povolené země*. Při volání do těchto zemí pak není zasíláno upozornění. Definování se provádí úpravou řádku, který začíná textem `ALLOWED_COUNTRIES=`. Tato řádka se může vyskytovat v `countries` souboru nejvýše jednou. Země se uvádí ve dvouznakovém formátu podle ISO 3166 s použitím oddělovače „`:`“. Za každou definovanou zemí musí být vždy přidán tento oddělovač.

Ukázka nastavení pro globální povolení volání do České a Slovenské republiky pro všechny IP adresy:

```
ALLOWED_COUNTRIES=CZ:SK:
```

Hlavním obsahem `countries` souboru jsou seznamy povolených zemí pro dané IP adresy, které se vytváří během učícího režimu a dále jsou doplňovány za běhu modulu (pokud to není zakázáno parametrem `-w`). Seznamy mohou být také upravovány ručně, mohou být přidávány povolené země nebo naopak odebírány. Definování IP adresy se provádí pomocí počátečního znaku „`-`“ na začátku nové řádky souboru, následující řádky začínající znakem „`=`“ určují seznam povolených zemí pro naposledy definovanou IP adresu. Pro povolení volání např. do Rakouska pro IP adresu 1.2.3.4 je formát následující:

```
-1.2.3.4
=AT:
```

Před každou změnou `countries` souboru je *důrazně doporučována jeho záloha*! Pokud modul zjistí neplatnou strukturu souboru, bude ho ignorovat a při ukončení modulu bude tento soubor přepsán. Řádky začínající znakem „#“ jsou pouze komentáře a nejsou nijak zpracovávány.

Cesta ke `countries` souboru lze změnit parametrem `-c` a případné vypnutí detekce volání do jiných zemí lze zajistit parametrem `-o`.

4.3 Správa paměti

Modul je navržen pro nepřetržitý provoz. S vypínáním modulu je počítáno pouze ve výjimečných případech, jako jsou aktualizace modulu na novou verzi, provedení změny v konfiguraci, aktualizace na serveru vyžadující restart operačního systému nebo při výměně hardwaru.

Modul za běhu zpracovává velmi mnoho informací a některé z nich si ukládá v paměti. Pokud z důvodu nedostatku paměti serveru nelze provést alokaci nové paměti, přestane modul ukládat nové informace a pokračuje ve svém běhu. Paměť serveru může být vyčerpána např. jiným spuštěným procesem v operačním systému. Po uvolnění paměti modul pokračuje standardně ve své práci a v ukládání informací.

Při dlouhodobém běhu modulu je potřeba průběžně uvolňovat nepoužívanou paměť. To je zajištěno díky pravidelně volané funkci `check_and_free_module_memory()`, která má za úkol spravovat alokovanou paměť. Tato funkce zajišťuje provedení kontroly vždy po definovaném intervalu `CHECK_MEMORY_INTERVAL` (čas je vyjádřen v sekundách od předchozí kontroly a je definován v souboru configuration.h). K uvolňování paměti dochází, pokud nebyla po definovaně dlouhou dobu používána nebo při překročení limitu počtu uložených položek. Funkce prochází postupně všechny uložené IP adresy a kontroluje:

1. *čas poslední komunikace* IP adresy – zda nepřekročila čas definovaný parametrem programu `-x`. V případě překročení dochází ke smazání všech uložených informací ohledně dané IP adresy.

2. *počet uložených položek v suffixovém stromě* pro jednu IP adresu – pokud je překročen limit definovaný parametrem programu `-q`, dochází ke smazání všech položek v suffixovém stromě dané IP adresy. Ostatní informace uložené k IP adrese jsou zachovány.

Poznámka: Při útoku hrubou silou, kdy přichází velké množství požadavků, může docházet k výrazné spotřebě paměti. Je proto doporučováno nastavovat limit počtu uložených položek v suffixovém stromě s ohledem na velikost monitorované sítě a dostupné paměti na serveru. Při testování modulu v rozsáhlé síti CESNET2 se ukázala být vhodná hodnota 100000.

4.4 Textový výstup modulu

Detekované události a další informativní výstupy modulu jsou zapisovány na standardní výstup (stdout) a případné chyby na standardní chybový výstup (stderr). Parametrem při spouštění modulu lze určit také log soubor, do kterého jsou zaznamenávány pouze detekované události. Jednoduchou úpravou v kódu modulu lze určit, zda daný textový výstup má být zapisován na standardní výstup, standardní chybový výstup nebo do log souboru, případně

do více výstupů zároveň. Pro změnu daného textového výstupu stačí změnit makro, které již zařídí zapsání do požadovaných výstupů:

```
PRINT_OUT(...);          // výstup do stdout
PRINT_ERR(...);          // výstup do stderr
PRINT_LOG(...);          // výstup do log souboru
PRINT_OUT_LOG(...);      // výstup do stdout a log souboru
PRINT_ERR_LOG(...);      // výstup do stderr a log souboru
```

Počet parametrů uvedených maker je libovolný. Makra jsou nahrazena voláním variadic funkcí (funkce s libovolným počtem parametrů).

Všechna výše uvedená makra zajistí vypsání aktuálního data a času na začátku každého textového výstupu, jsou však definována i další makra s názvem `*_NOTDATETIME`, která aktuální datum a čas nevypisují.

Ukázka definování makra pro výpis na standardní výstup:

```
#define PRINT_OUT(...) write_to_stream(stdout, \
        get_actual_time_string(), ";", __VA_ARGS__, NULL)
```

Samotná deklarace volané variadic funkce `write_to_stream()` je pak následující:

```
void write_to_stream(FILE * stream, char * str, ...);
```

V posledním parametru předávaném funkci je vždy nutné předávat hodnotu `NULL`, která tímto ukončuje procházení dalších předaných parametrů.

Po spuštění modulu se vypíše aktuálně používané nastavení, ze kterého se lze ujistit, zda odpovídá našim požadavkům a především umožňuje jeho zpětné zjištění z log souborů. Zároveň je tímto zapsán datum a čas spuštění a verze modulu. Ukázka výstupu modulu při spuštění:

```
2014-11-14 15:30:00;-----------------------------------------------------
2014-11-14 15:30:00;Start VoIP fraud detection module (version:1.0.0) ...
2014-11-14 15:30:00;   Module configuration:
2014-11-14 15:30:00;    - countries detection=ON
2014-11-14 15:30:00;    - max_prefix_length=10
2014-11-14 15:30:00;    - min_length_called_number=0
2014-11-14 15:30:00;    - prefix_examination_detection_threshold=10
2014-11-14 15:30:00;    - detection_interval=10
2014-11-14 15:30:00;    - detection_pause_after_attack=30
2014-11-14 15:30:00;    - max_item_prefix_tree=100000
2014-11-14 15:30:00;    - clear_data_no_communication_after=604800
2014-11-14 15:30:00;    - countries file="countries.dat"
2014-11-14 15:30:00;    - event_id file="event_id"
2014-11-14 15:30:00;    - log file:"log"
2014-11-14 15:30:00;-----------------------------------------------------
```

Při ukončování modulu (při přijmutí SIGTERM nebo SIGINT signálu běžícím procesem) se vypíšou statistiky o množství zpracovaných dat, počtu unikátních *User-Agent* hlaviček v rámci INVITE požadavků, počty detekovaných událostí, detekovaných útoků a vypíše se informace o ukončování modulu.

4.5 Nahlášení detekované události do výstupního rozhraní modulu

O detekování útoku modulem je potřeba také informovat odpovědné správce sítí nebo bezpečnostní týmy, které se zabývají řešením bezpečnostních incidentů v rámci dané sítě a jsou schopné reagovat na vzniklé situace.

V rámci systému Nemea je používán modul Report Handler, který ukládá nahlášené útoky do společné databáze[4]. Dotazy nad databází lze získat přehled o detekovaných událostech všech modulů.

Modul pro detekci zneužití VoIP ústředen zasílá na své výstupní rozhraní při detekci položky v závislosti na typu události:

1. Při detekování pokusu o nalezení předvolby jsou zasílány následující položky:

   `EVENT_ID`, `EVENT_TYPE`, `SRC_IP`, `DETECTION_TIME`, `TIME_FIRST`, `VOIP_FRAUD_SIP_TO`, `VOIP_FRAUD_USER_AGENT`, `VOIP_FRAUD_PREFIX_LENGTH`, `VOIP_FRAUD_PREFIX_EXAMINATION_COUNT`, `VOIP_FRAUD_SUCCESSFUL_CALL_COUNT`, `VOIP_FRAUD_INVITE_COUNT`

2. Při zahájení hovoru do jiné země se zasílají položky:

   `EVENT_ID`, `EVENT_TYPE`, `SRC_IP`, `DST_IP`, `DETECTION_TIME`, `VOIP_FRAUD_SIP_TO`, `VOIP_FRAUD_SIP_FROM`, `VOIP_FRAUD_USER_AGENT`, `VOIP_FRAUD_COUNTRY_CODE`

Význam a popis jednotlivých položek je uveden v příloze D.

Detekované události mohou být dále v databázi zpracovávány a vyhodnocovány, mohou být i např. zasílána upozornění pomocí triggeru samotné databáze. Modularita systému Nemea umožňuje výstup zasílat také na jiný modul, který zasílá např. emailové upozornění nebo upozornění do libovolného externího systému.

[4]Je používána dokumentová databáze MongoDB, která patří do skupiny NoSQL databází. Tento typ databáze umožňuje, aby jednotlivé moduly v rámci systému Nemea mohly zasílat různé informace o detekovaných událostech. Každý tzv. dokument v databázi může mít odlišný formát.

V rámci sdružení CESNET je vyvíjen systém Warden pro sdílení informací o detekovaných bezpečnostních událostech. Jak se uvádí na webových stránkách projektu:

> *„Systém Warden umožňuje jednoduše a efektivně týmu CERTS/CSIRT a dalším zapojeným bezpečnostním týmům rychlé sdílení a využití informací o detekovaných anomáliích, které byly zjištěny nasazenými nástroji v jimi monitorovaných sítích. Tato data jsou systémem předávána a poskytují týmům další užitečné informace potřebné pro zajištění bezpečnosti a monitoringu zdraví sítě.“* [21]

Vyvinutý modul podporuje zasílání hlášení do systému Warden a je tedy možno do budoucna zprovoznit zasílání informací o zaznamenaných útocích i do tohoto systému.

Kapitola 5

Testování a ověření funkčnosti detekčního modulu

Tato kapitola popisuje průběh testování detekčního modulu, analýzu zaznamenaných útoků a statistiky z provedených testů.

5.1 Hledání nastavené předvolby útočníky

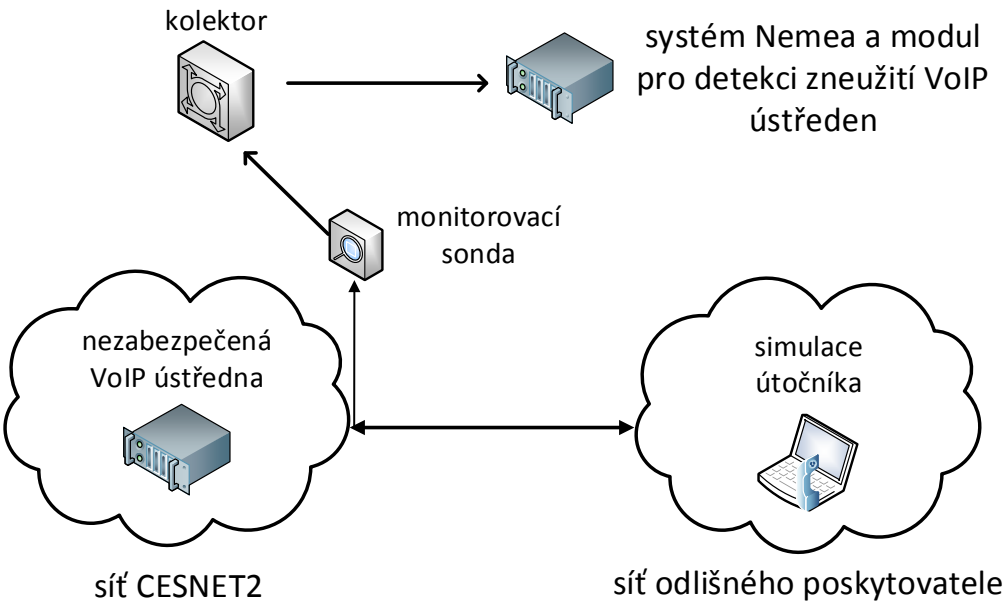
V první části se zaměřím na pokusy o nalezení nastavené předvolby útočníky, kteří se snaží vytvořit telefonní hovor do sítě PSTN.

5.1.1 Detekce vlastního útoku

Pro ověření funkčnosti implementovaného detekčního modulu, jsem provedl test se simulací útoku. Počítač umístěný mimo síť CESNET2 (označím jej „simulace útočníka“) zasílal INVITE požadavky na počítač s nainstalovanou ústřednou Asterisk (označím ji „nezabezpečená VoIP ústředna“), který byl připojen v síti CESNET2. Tato komunikace je sledována monitorovacími sondami a informace o zaznamenané komunikaci jsou zasílány do systému Nemea, kde je spuštěn modul pro detekci zneužití VoIP ústředen. Celé schéma testu je zobrazeno na obrázku 5.1.

Nainstalovaná ústředna Asterisk byla nakonfigurována tak, aby povolila volání bez autentizace na číslo 100135666531, zahájila hovor, přehrála uvítací zprávu a zavěsila hovor. Toto chování simuluje VoIP ústřednu, která pomocí brány dokáže volat do sítě PSTN.

Chování „útočníka“ bylo simulováno pomocí nástroje *SIPVicious* [10]. Konkrétně pomocí svwar.py, který je určen pro hledání aktivního telefonního čísla na ústředně. Při každém spuštění nástroje je ověřováno spojení k ústředně a její chování. Bylo nutné upravit zdrojový kód svwar.py, aby při spuštění nebyl



Obrázek 5.1: Schéma testu detekce vlastního útoku

zasílán INVITE požadavek na volání náhodného čísla, který by mohl ovlivňovat prováděný test, ale definoval jsem první INVITE požadavek na volání stále stejného čísla (9999). Dále jsem přidal parametr –sleeptime, který udává čas v sekundách mezi jednotlivými zasílanými požadavky a implementoval jsem zasílání dalšího požadavku až po čase určeném tímto parametrem. Nástroj byl spuštěn s nastavením pro odesílání INVITE požadavků na „nezabezpečenou VoIP ústřednu“ v síti CESNET2. Požadavky byly zasílány postupně na volání následujících čísel:

```
000135666531, 001135666531, 010135666531, 020135666531
030135666531, 040135666531, 050135666531, 060135666531
070135666531, 080135666531, 090135666531, 100135666531
110135666531
```

Simulace útočníka úspěšně nalezla aktivní telefonní číslo 100135666531 a zahájila hovor. Útok byl úspěšně detekován modulem a byla ověřena požadovaná funkčnost. [5]

[5]Na obou testovacích počítačích jsem použil operační systém Debian 7. Asterisk ústřednu jsem nainstaloval ve verzi 1.8.13.1.

5.1.2 Krátkodobé testy

Modul byl v průběhu celé implementace postupně testován a byly odstraňovány nalezené chyby. První testy modulu jsem spustil s nastavením meze počtu unikátních předvoleb pro jednu IP adresu na hodnotu 5 (při překročení této hodnoty je nahlášen útok) a maximální délku zkoumaného prefixu jsem nastavil na hodnotu 4, která měla podle mých prvotních předpokladů vyhovovat pro zaznamenání většiny útoků. Po přibližně třech hodinách od spuštění první testovací verze v reálné síti začal modul hlásit první nalezené útoky. Byly zaznamenány útoky na čísla odpovídající následujícímu formátu:

(zkoušená předvolba)97259(pokračování čísla)

Na ukázku uvádím několik konkrétních volaných telefonních čísel:

0000972595243897
1200972598380527
6600972592168549
7000972598126387
9000972598884330
20000972595561283
88880972598380527
9400011972595358613
+1972592338234
00+00972598380546
***00**972592168549
*****972592337882
*****011**972592375315

Jak se ukázalo při analýze průběhu jednotlivých útoků, zkoušené předvolby jsou často delší, než jsem původně předpokládal, délka předvolby se nejvíce pohybovala v rozmezí $5 - 8$ znaků, resp. číslic. Při dalším testování jsem používal jako parametr maximální zkoumané délky předvolby hodnotu 10.

Po odstranění útočníky zkoušené předvolby zůstává předvolba volaného čísla 972, ta může představovat:

1. volání v rámci České Republiky – předvolba odpovídá volání do společnosti České dráhy, a.s.

2. mezinárodní volání – předvolba je přiřazena Izraeli.

Požadavky byly směrovány na různé cílové IP adresy v síti. Lze však pozorovat, že útočníci zasílají většinou stejné nebo velmi podobné požadavky postupně na více IP adres patřících do stejné podsítě (většinou mají shodných prvních 16 bitů pro IP adresy verze 4). Požadavky většinou obsahovaly v SIP

From hlavičce až čtyřmístné číslo (např. 16, 101, 105, 888, 1101, 6001) doplněné „@“ a cílovou IP adresou, na který daný INVITE požadavek směřoval.

Během prvních 48 h testování se vyskytly ještě dva útoky s jiným průběhem. Jednalo se o intenzivní (bruteforce) útoky, kdy bylo odesíláno mnoho INVITE požadavků na volání různých až 6 místných čísel, v některých chvílích dosahoval počet požadavků až 700 za 10 sekund. Všechny požadavky byly pokusy na volání různých kombinací čísel s délkou až 5 znaků.

Další prováděné krátkodobé testy probíhaly opakovaně po celou dobu vývoje a vykazovaly stejné charakteristiky. Další čísla, na která byly zaznamenávány pokusy o vytvoření hovoru, jsou po odstranění zkoušené předvolby např.:

44870875[7818|6604]
448708752[633|623|617|629]
448704903946
48587313039
441322517105
970595185401
48587314494
201272995281
442036037786
48222198030
48846413945
46184952088
34865670014

5.1.3 Dlouhodobý test č. 1

Po řádném otestování funkčnosti modulu byl spuštěn dlouhodobý test na serveru po dobu 18 dní (15. 11. 2014 – 2. 12. 2014). Modul byl spuštěn s tímto nastavením:

- délka zkoumané předvolby: 10
- minimální délka volaného čísla: 0
- interval detekce pro IP adresu: 10 sekund
- mez počtu unikátních předvoleb pro jednu IP adresu: 10
- pauza po detekování události pro danou IP adresu: 30 sekund
- maximální počet položek v suffixovém stromě: 100 000
- uložení informací k dané IP adrese bez komunikace po dobu: 14 dní

Celkové statistiky běhu modulu jsou uvedeny v tabulce 5.1. INVITE požadavky představují volání na telefonní čísla (s povolenými znaky). Do uvedených ACK, CANCEL požadavků a OK, RINGING odpovědí jsou zahrnuty pouze ty, kterým předcházel odpovídající INVITE požadavek a které měly odpovídající *Call-ID* a *To* hlavičku.

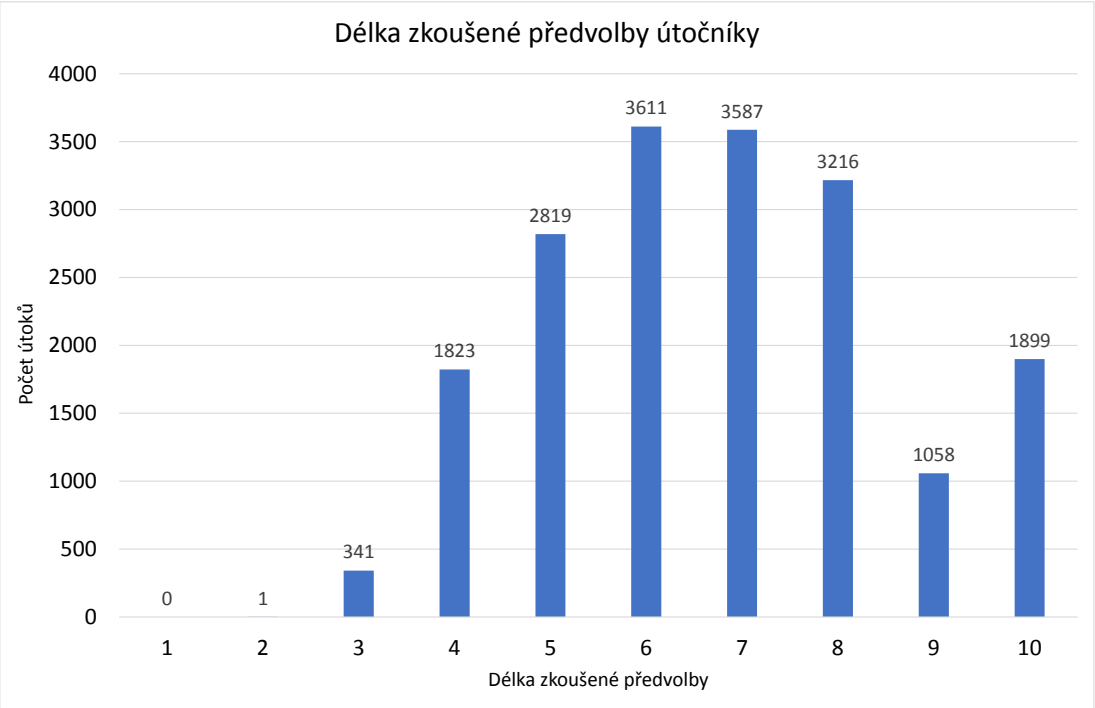
Tabulka 5.1: Statistika dlouhodobého testu č. 1 pro hledání předvoleb útočníky

popis	**počet**
detekovaných událostí	19 648
detekovaných útoků	18 355
zpracovaných rozšířených toků s alespoň jedním INVITE požadavkem	12 905 428
celkem INVITE požadavků	13 031 329
celkem OK odpovědí	1 177 167
celkem ACK požadavků	467 052
celkem CANCEL požadavků	17 506
celkem RINGING odpovědí	149 325
poměr OK odpovědí a INVITE požadavků	0,09
průměrně INVITE požadavků za 1 den	723 963
průměrně detekovaných událostí za 1 den	1 092
průměrně detekovaných útoků za 1 den	1 020
průměrně detekovaných událostí za 1 hodinu	46
průměrně detekovaných útoků za 1 hodinu	43

Z celkových statistik vyplývá, že v síti bylo zachyceno velmi mnoho INVITE požadavků, ale pouze na malý počet z nich byla zachycena OK odpověď (poměr OK odpovědí a INVITE požadavků: 0,09). Velmi mnoho požadavků pravděpodobně mělo za cíl zjistit, zda daný cílový prvek odpovídá na SIP komunikaci, postupně proskenovat celou síť a nalézt ústředny, na které pak může být proveden útok.

Zachycených ACK požadavků je dle očekávání méně než OK odpovědí. Požadavky byly pravděpodobně směrovány linkou, kde není instalována monitorovací sonda (bližší vysvětlení je uvedeno v sekci 3.1). Počet ACK požadavků je menší než polovina z počtu OK odpovědí.

Ve sledované síti jsou útoky velmi časté, průměrně bylo zaznamenáno 43 útoků za hodinu. Z histogramu 5.2 lze pozorovat, že délka zkoušené předvolby útočníky je nejčastěji 6 nebo 7 znaků.



Obrázek 5.2: Histogram: Délka zkoušené předvolby útočníky v dlouhodobém testu č. 1

Útoky byly zaznamenány z celkem 149 zdrojových IP adres. Celkový počet adres, ze kterých byl zachycen alespoň jeden INVITE požadavek je 2527. Všechny zaznamenané IP adresy jsou verze 4, kvůli zajištění anonymity nejsou konkrétně uváděny. Četnosti útoků podle zdrojových adres jsou uvedeny v tabulce 5.2. Zajímavým zjištěním je, že byl zaznamenán pouze jediný útok pro přibližně 27 % adres, ze kterých byly detekovány útoky. Při analýze průběhu útoků z těchto adres jsem zjistil, že některé útoky probíhaly pouze jeden den a zasíláno bylo maximálně 10 požadavků za hodinu. Dalším typem nalezeného útoku je zasílání stejných požadavků na více cílů současně. Při útoku, který pocházel z jedné zdrojové adresy a probíhal celkově 4 dny (15.-16. 11., 20.-21. 11.) byly zasílány požadavky na celkem 261 cílových adres.

Při útocích detekovaných z IP adresy, která měla nejvyšší počet útoků za celý dlouhodobý test (2040 útoků), směřovaly požadavky pouze na 7 konkrétních cílů a to v období od 20. 11. do 1. 12.

Rozložení útoků podle jednotlivých dnů je znázorněno v histogramu 5.3.

Tabulka 5.2: Četnosti útoků z dlouhodobého testu č. 1 podle IP adres

počet IP adres	**počty útoků pro každou IP adresu**
1	2 040
1	1 285
1	1 241
1	1 152
6	600-750
33	100-550
39	10-100
27	2-10
40	1

Obrázek 5.3: Histogram: Počty detekovaných útoků za jednotlivé dny v dlouhodobém testu č. 1

V rámci zaznamenaných útoků (18 355) byly vyhlášeny detekce modulem na pokusy o volání 2 401 unikátních čísel, 10 nejčastějších čísel uvádím v tabulce 5.3. Zvýrazněna je zkoušená předvolba, kterou se snaží pravděpodobně útočníci nalézt a využít nedostatečně zabezpečené ústředny k volání do PSTN.

Tabulka 5.3: Nejčastěji zkoušená čísla z hlášení detekcí modulem (dlouhodobý test č. 1)

volané číslo	počet zaznamenaných útoků
6666600972548700000	349
54700441904899300	205
999900972598601000	193
972595079175	191
90081048587314400	140
1100972592627640	130
9911100972595080000	128
100048587314494	123
9999000972548700000	117
972595081667	115

Při testu se také objevily pokusy na volání s méně obvyklými předvolbami: 1**, 1***, ***0, #+, ##, 10#, 123456+, 123400, 1234123400.

Celkem u 181 útoků byl zaznamenán alespoň jeden úspěšně navázaný hovor v rámci zkoušených předvoleb, to představuje přibližně necelé 1 % z celkového počtu útoků. Tyto útoky lze označit za úspěšné, došlo k nalezení předvolby a úspěšnému zahájení hovoru. Bohužel nelze ověřit, zda cílová ústředna opravdu zahájila hovor do PSTN nebo ústředna byla nakonfigurována jako tzv. honeypot, kdy dochází k zaznamenávání činnosti útočníka a pouze k simulaci uskutečněných hovorů do PSTN.

Úspěšné útoky pocházely z 98 unikátních zdrojových IP adres, podle geolokace jsou umístěny především ve Spojených státech amerických, dále ve Velké Británii, Francii, Brazílii, Řecku, Nizozemsku, Německu, Palestině a některé také v České republice. Při detekovaných útocích z některých českých IP adres byly zasílány požadavky na cílové prvky nacházející se v zahraničí a tyto požadavky byly na volání 3 znakových čísel. Nejvíce požadavků bylo 99 zasílaných po dobu 8 dnů. Dle počtu a průběhu zasílání jednotlivých požadavků tyto mohou být falešně pozitivní detekce. Oproti ostatním pokusům není pozorovatelné zkoušení všech možných předvoleb. (Pokud bychom nechtěli modulem detekovat události, při kterých je voláno na takto krátká čísla, stačí parametrem modulu zvýšit hodnotu určující minimální délku volaného čísla.) Ostatní útoky vykazují typický průběh pro hledání předvolby a volání na čísla do sítě PSTN začínající 972*, 970*, 448*, 348*.

Typický průběh zaznamenaného zkoušení předvolby uvádím v příloze A, kde je ukázán průběh útoku, který začal 25. 11. 2014. INVITE požadavky byly zasílány v počtu 2-3 za 1 hodinu.

5.1.4 Dlouhodobý test č. 2

Druhý dlouhodobý test byl spuštěn po dobu 10 dní (19. 12. 2014 – 28. 12. 2014). Na rozdíl od testu č. 1 byl modul spuštěn v několika instancích s různým parametrem minimální délky volaného čísla (parametr `-d`). Pro test jsem zvolil hodnoty parametru: 0, 6, 9, 12. Cílem testu bylo zjistit, zda zvolené hodnoty parametru `-d` budou ovlivňovat počty detekovaných událostí a útoků.

Výsledky testu jsou uvedeny v tabulce 5.4. Uvedené počty INVITE požadavků představují volání na telefonní čísla (s povolenými znaky) o minimální délce volaného čísla definované parametrem `-d`.

Tabulka 5.4: Statistika dlouhodobého testu č. 2 pro hledání předvoleb útočníky

popis	**-d 0**	**-d 6**	**-d 9**	**-d 12**
detekovaných událostí	15 455	15 131	15 095	15 035
detekovaných útoků	14 996	14 914	14 897	14 888
celkem INVITE požadavků	1 071 361	652 185	646 264	638 827

Z výsledků testu vyplývá, že zvolené hodnoty minimální délky volaného čísla výrazně neovlivňovaly počet zaznamenaných událostí ani útoků, rozdíl mezi počtem událostí u hodnoty parametru 0 a 12 představuje pouze necelé 3 % z maximálního počtu událostí. Téměř všechny útoky byly tedy detekovány u pokusů na volání telefonních čísel s minimální délkou 12 znaků.

V rámci testu bylo také prováděno zaznamenávání *User-Agent* hlaviček u všech INVITE požadavků na volání telefonních čísel. Celkem bylo zachyceno 401 unikátních hlaviček. Nejčastějších 5 hlaviček je uvedeno v tabulce 5.5. Dvě nejčastější hlavičky identifikují nástroje pro generování SIP požadavků. Požadavky zaslané s *User-Agent* hlavičkou „sipcli/v1.8“ budou pravděpodobně vygenerovány nástrojem *SipCLI* [6], který umožňuje generovat požadavky pomocí příkazové řádky v operačních systémech Windows. Hlavička „friendly-scanner“ ukazuje na použití nástroje *SIPVicious* [10]. Během testu bylo zaznamenáno, že více než 99 % ze všech INVITE požadavků bylo pravděpodobně vygenerováno pomocí nástrojů určených pro testování SIP protokolu. Tak jako ostatní hlavičky však mohou být pozměněny. U celkem 2440 požadavků nebyla uvedena žádná hlavička nebo byla prázdná.

Tabulka 5.5: Nejčastější *User-Agent* hlavičky v dlouhodobém testu č. 2 pro hledání předvoleb útočníky

User-Agent hlavička	**počet výskytů**
sipcli/v1.8	643 312
friendly-scanner	424 178
Cisco-SIPGateway/IOS-12.x	6 304
FPBX-2.10.1(1.8.7.1	1 153
Asterisk PBX 11.11.0	570

V zachycené SIP komunikaci se vyskytovaly i *User-Agent* hlavičky, které obsahovaly typ zařízení, verzi firmwaru a dokonce MAC adresu (z důvodu zajištění anonymity není uváděna celá MAC adresa):

```
User-Agent: Well T20 hw7.0.0.54 fw9.43.9.4 00:15:65:XX:XX:XX
```

5.1.5 Test s různými parametry modulu

Další provedený test měl za úkol zjistit závislost počtu detekovaných událostí a útoků na nastavené mezi počtu unikátních předvoleb pro jednu IP adresu (parametr `-t`) a na maximální délce zkoumané předvolby (parametr `-m`). Během testu bylo spuštěno celkem 12 instancí modulu, každá instance měla nastavený pouze jeden odlišný parametr. Ostatní nastavené parametry byly stejné jako v dlouhodobém testu č. 1. Test probíhal po dobu 9 dní (25. 12. 2014 – 2. 1. 2015). Počet zaznamenaných útoků v závislosti na nastavených parametrech uvádí tabulka 5.6.

V každé spuštěné instanci modulu bylo zpracováno celkem 1 221 053 INVITE požadavků na volání telefonních čísel.

Postupně se zvyšujícím se parametrem `-m` dochází ke zvyšování počtu detekovaných událostí a útoků. Výjimka je u hodnoty parametru 12, kdy počet událostí i útoků je nižší než u hodnoty 10. Způsobeno je to příliš vysokou hodnotou parametru. Po detekování útoku dochází k vymazání celého podstromu, na kterém byl detekován útok. Docházelo tak k mazání velké části stromu. Počty detekovaných událostí se pro hodnotu 8 a 10 příliš neliší. Pro detekování útoků je tedy dle provedeného testu vhodné používat hodnotu parametru 8 nebo 10. Nižší hodnota parametru `-m` také znamená nižší výpočetní náročnost pro provedení detekčního algoritmu modulem.

Tabulka 5.6: Statistika testu s různými parametry modulu pro hledání předvolby útočníky

parametr spuštěné instance modulu	detekovaných událostí	detekovaných útoků
-m 2	1 125	1 108
-m 4	12 774	12 526
-m 6	20 869	20 534
-m 8	24 580	24 235
-m 10	25 175	24 798
-m 12	24 617	24 218
-t 5	39 470	38 866
-t 10	25 175	24 798
-t 15	17 634	17 325
-t 20	12 675	12 386
-t 30	8 257	8 003
-t 40	6 043	5 807

Naopak se zvyšujícím se parametrem `-t` dochází ke snižování počtu detekovaných událostí a útoků. Zajímavé výsledky ukazuje tabulka 5.7 se součtem všech pokusů na volání v rámci všech detekovaných událostí zaznamenaných instancí modulu. Příliš nízká hodnota parametru 5 způsobuje velmi časté („brzké“) detekování útoku a následné smazání celého odpovídajícího podstromu. Díky tomu dochází k zaznamenávání menšího počtu pokusů. Se zvyšujícím se parametrem od hodnoty 10 dochází ke snižování počtu zaznamenaných pokusů. Pro detekování všech útoků je dle provedeného testu vhodné používat hodnotu parametru 10.

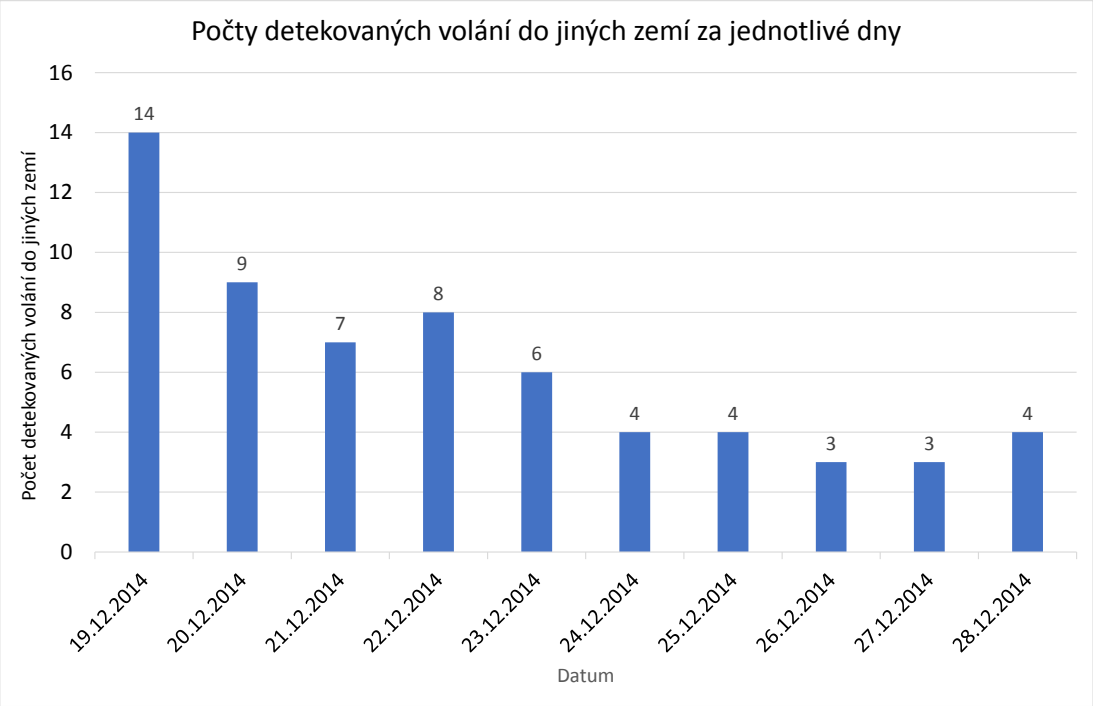
Tabulka 5.7: Test s různým parametrem `-t`: Počet pokusů na volání v rámci všech detekovaných událostí

parametr spuštěné instance modulu	počet pokusů na volání v rámci všech detekovaných událostí
-t 5	463 213
-t 10	469 274
-t 15	464 049
-t 20	443 125
-t 30	426 391
-t 40	411 123

5.2 Detekování volání do jiné země

Po ověření funkčnosti detekování úspěšně navázaných telefonních hovorů do jiných zemí byl modul spuštěn v učícím režimu po dobu 7 dní (12. 12. 2014 – 18. 12. 2014) v reálné síti. `Countries` soubor nastavoval Českou republiku jako globálně povolenou cílovou zemi. Po ukončení učícího režimu seznam povolených zemí obsahoval celkem 15 různých zemí pro celkem 193 zdrojových IP adres.

Modul v režimu aktivní detekce volání do jiných zemí byl spuštěn po dobu 10 dní (19. 12. 2014 – 28. 12. 2014). Byl použit výchozí režim, kdy po detekci a nahlášení volání do jiné země je přidána daná země do seznamu povolených zemí. Během testu bylo zaznamenáno celkem 62 detekcí. Počty detekcí podle jednotlivých dnů zobrazuje histogram 5.4. Z histogramu je patrné postupné snižování počtu detekcí modulem.



Obrázek 5.4: Histogram: Počty detekovaných volání do jiných zemí za jednotlivé dny

Po ukončení testu seznam povolených zemí obsahoval celkem 18 různých zemí pro celkem 251 zdrojových IP adres. Četnosti 5 nejčastějších zemí jsou uvedeny v tabulce 5.8.

Tabulka 5.8: Četnosti 5 nejčastějších zemí zaznamenaných během testu pro detekci volání do jiných zemí

cílová země	**počet zdrojových IP adres**
Spojené státy americké	132
Slovensko	24
Mongolsko	16
Tádžikistán	11
Rusko	8

5.3 Další poznatky z prováděných testů a analýzy dat

Během ladění a testování detekčního modulu jsem v monitorovaných datech narazil na SIP hlavičky, které obsahovaly apostrofy, přesný obsah byl následující:

```
sip:'or''='@X.X.X.X
```

Textový řetězec `X.X.X.X` označuje různé IP adresy. S velkou pravděpodobností toto může ukazovat na pokusy o vyvolání *SQL injection* u některého z podpůrných systémů.

Velmi pravidelně se v datech také nacházely *From* hlavičky přesně s tímto obsahem:

```
sip:anonymous@anonymous.invalid
```

Po dohledání v RFC 3323 [38] jsem zjistil, že tato SIP URI je doporučována v případě potřeby skrytí volající strany.

5.4 Zhodnocení provedených testů

Při testování modulu v síti CESNET2 bylo detekováno mnoho útoků, které vykazují pokusy o nalezení nastavené předvolby na VoIP ústředně pro volání do sítě PSTN. V rámci provedených testů nebyly zjištěny útoky, které by vykazovaly záměrně odlišné telefonní číslo uvedené v *Request-URI* a *To* hlavičce. Lze tedy konstatovat, že v dnešní době pro detekování pokusů o nalezení předvolby účastníky postačuje zaznamenávání *To* hlavičky. Rozdílné hlavičky byly především z důvodu uvedení mezinárodní telefonní předvolby v *Request-URI*, na rozdíl od *To* hlavičky, kde bylo uvedeno číslo bez mezinárodní předvolby. Často byla uváděna předvolba 420, která označuje Českou republiku.

Naprostá většina INVITE požadavků, na základě kterých byl následně detekován útok, měla jako *User-Agent* hlavičku uvedenou hodnotu „sipcli/v1.8“. Podle popisu softwaru by mohly být tyto požadavky vygenerovány nástrojem *SipCLI* [6].

Detekce volání do jiných zemí je užitečnou funkcí modulu, která může upozornit na zneužití VoIP ústředny. V počítačových sítích, kde málo hovorů směřuje do zahraničí, budou detekce málo časté. V případě rozsáhlých sítí jako je síť CESNET2 budou detekce častější a je nutné následně provést jejich detailnější vyhodnocení. V každém případě, pokud začnou být zaznamenávány úspěšně navázané hovory do různých zemí ve větší intenzitě, může to naznačovat neobvyklou aktivitu v síti. Během prováděného testu k takovýmto událostem nedocházelo.

5.5 Používaný software

- Pro překlad a sestavení spustitelného binárního souboru detekčního modulu byl využíván v rámci systému Nemea balík *Autotools – Autoconf* [2], *Automake* [3], *Libtool* [7].
- Ukládání jednotlivých verzí detekčního modulu během vývoje zajistil verzovací systém *Git* [4].
- Hledání chyb v uvolňování paměti pomocí nástroje *Valgrind* [13].
- Pro sledování síťové komunikace byl využíván program *Tcpdump* [12], následná analýza pak v programu *Wireshark* [15].
- Pro testovací účely virtualizační nástroj *VMware Player* [14], ústředna *Asterisk* [1]. Nástroj *SIPVicious* [10] pro generování INVITE požadavků v rámci testu detekce vlastního útoku.

Závěr

V první části diplomové práce jsem se zaměřil na technologii VoIP a detailní popis protokolu SIP. Zabýval jsem se útoky, které mohou být použity ve VoIP telefonii a vysvětlil principy zneužití nedostatečně zabezpečených telefonních VoIP ústředen v počítačových sítích. Dále jsem uvedl obecná doporučení pro zajištění bezpečnosti ve VoIP telefonii a popsal běžně dostupné softwarové nástroje, které lze využít pro testování.

V druhé části jsem úspěšně provedl analýzu a návrh detekčních metod pro identifikaci zneužití VoIP ústředen v počítačových sítích na základě rozšířených informací o tocích z monitorovacích sond. Následně jsem na základě těchto metod navrhl a úspěšně implementoval modul pro systém Nemea, který je schopný pracovat i ve velkých sítích.

Během vývoje jsem nalezl chybu v určování komunikace ve VoIP pluginu monitorovací sondy a chybu jsem společně s vedoucím nahlásil autorům k opravě. Během své práce jsem navíc rozšířil VoIP plugin o potřebné položky.

V průběhu testovacího provozu mohlo docházet k výpadkům zpráv z monitorovacího systému a bylo potřeba takový stav řešit. Zajistil jsem proto funkčnost modulu i v případě, že monitorovací sondy nezachycují celou komunikaci protokolu SIP.

Vzniklý modul detekuje hádání číselných předvoleb útočníky, kteří se snaží vytvořit telefonní hovor z počítačové sítě na telefonní čísla veřejné telefonní sítě, a sleduje, zda se útok zdařil, tedy zda v rámci zkoušených volání došlo k úspěšnému navázání hovoru. Modul je navržen pro nepřetržitý běh a je schopný detekovat útoky, při kterých jsou jednotlivé požadavky na volání zasílány ve velmi malých intenzitách. Dále kontroluje, do jakých zemí jednotliví klienti volají. Podporuje učící režim pro ukládání zemí, který se automaticky po nastavené době deaktivuje. Všechny informace o povolených zemích jsou ukládány do souboru, který je svou strukturou dobře čitelný a může být v případě potřeby ručně upravován a dále zpracováván. Pokud modul rozpozná úspěšně navázaný hovor do jiné země, než je pro daného klienta obvyklé, je vytvořeno upozornění. Detekované události jsou zasílány do společné data-

báze, která je využívaná v rámci systému Nemea. Podporovány jsou IP adresy verze 4 a verze 6, podpora je zajištěna i v rámci lokalizace IP adresy.

Modul jsem průběžně testoval v reálné síti a v práci jsem podrobně popsal výsledky testů a zaznamenané útoky zachycené modulem. Pro ověření funkčnosti jsem provedl simulaci útoku na nezabezpečenou ústřednu.

Díky této práci jsem získal praktické zkušenosti a poznatky o tom, jak probíhá monitorování v tak velké síti, jakou je síť CESNET2. Velkou výhodou při implementaci byla možnost průběžného testování modulu na reálných datech. Po každé nové verzi modulu byly zjištěné nedostatky odstraněny a napomohly k dalším implementovaným vylepšením. Pokud by byl modul testován pouze na simulovaných útocích, pravděpodobně by se nedosáhlo takto odladěného detekčního modulu schopného produkčního nasazení pro monitorování počítačových sítí.

Věřím, že se vyvinutý modul pro detekci zneužití VoIP ústředen stane součástí veřejné verze balíku Nemea a bude pomáhat správcům sítí v obraně před útočníky a při vyhledávání nedostatečně zabezpečených VoIP ústředen. Díky upozornění modulu může být zabráněno velkým finančním ztrátám.

Literatura

[1] Asterisk Software. [cit. 2014-12-20]. Dostupné z: http://www.asterisk.org/products/software

[2] Autoconf. [cit. 2014-12-12]. Dostupné z: https://www.gnu.org/software/autoconf/

[3] Automake. [cit. 2014-12-12]. Dostupné z: https://www.gnu.org/software/automake/

[4] Git – Distributed version control system. [cit. 2014-12-12]. Dostupné z: http://git-scm.com

[5] Kamailio SIP Server. [cit. 2014-12-20]. Dostupné z: http://www.kamailio.org

[6] KaplanSoft: SipCLI. [cit. 2014-12-28]. Dostupné z: http://www.kaplansoft.com/sipcli/

[7] Libtool – A generic library support script. [cit. 2014-12-12]. Dostupné z: http://www.gnu.org/software/libtool/

[8] SIP Swiss Army Knife (Sipsak). [cit. 2014-12-14]. Dostupné z: http://sourceforge.net/projects/sipsak.berlios/

[9] SIPp (Open Source test tool / traffic generator for the SIP protocol). [cit. 2014-12-14]. Dostupné z: http://sipp.sourceforge.net

[10] SIPVicious (Tools for auditing SIP based VoIP systems). [cit. 2014-12-14]. Dostupné z: https://code.google.com/p/sipvicious/

[11] SiVuS (SiP Vulnerability Scanner) – User Guide v1.07. [cit. 2014-12-14]. Dostupné z: http://www.voip-security.net/pdfs/SiVuS-User-Doc1.7.pdf

[12] Tcpdump – A powerful command-line packet analyzer. [cit. 2014-12-12]. Dostupné z: http://www.tcpdump.org

[13] Valgrind – System for debugging and profiling Linux programs. [cit. 2014-12-12]. Dostupné z: http://valgrind.org

[14] VMware Player – Desktop virtualization application. [cit. 2014-10-11]. Dostupné z: http://www.vmware.com/cz/products/player/

[15] Wireshark – Network protocol analyzer. [cit. 2014-12-12]. Dostupné z: https://www.wireshark.org

[16] Bartoš, V.; Žádník, M.; Čejka, T.: *CESNET Technical Report 9/2013, Nemea: Framework for stream-wise analysis of network traffic*. CESNET z.s.p.o., [cit. 2014-11-03]. Dostupné z: http://www.cesnet.cz/wp-content/uploads/2014/02/trapnemea.pdf

[17] Bartoš, V.; Žádník, M.; Čejka, T.: *Liberouter / Cesnet TMC group: Nemea project*. CESNET z.s.p.o., [cit. 2014-11-03]. Dostupné z: https://www.liberouter.org/nemea/

[18] Bradley Huffaker, k. c., Marina Fomenkov: *Geocompare: a comparison of public and commercial geolocation databases*. CAIDA, University of California, 2011, [cit. 2014-11-25]. Dostupné z: http://www.caida.org/publications/papers/2011/geocompare-tr/geocompare-tr.pdf

[19] CESNET z.s.p.o.: *CESNET | Síť CESNET2*. [cit. 2014-11-08]. Dostupné z: http://www.cesnet.cz/sluzby/pripojeni/sit-cesnet2/

[20] CESNET z.s.p.o.: *Liberouter / Cesnet TMC group: Cards*. [cit. 2014-12-04]. Dostupné z: https://www.liberouter.org/technologies/cards/

[21] CESNET z.s.p.o.: *Warden - CESNET-CERTS*. [cit. 2014-11-10]. Dostupné z: https://csirt.cesnet.cz/Warden

[22] CFCA: *Communications Fraud Control Association announces results of Worldwide Telecom Fraud Survey*. 2013, [cit. 2014-11-21]. Dostupné z: http://www.cfca.org/pdf/survey/CFCA2013GlobalFraudLossSurvey-pressrelease.pdf

[23] Cisco Systems, Inc.: *CISCO IOS NETFLOW OVERVIEW*. 2004, [cit. 2014-11-03]. Dostupné z: http://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_presentation0900aecd80311f57.pdf

[24] Claise, B.; Trammell, B.; Aitken, P.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011 (INTERNET STANDARD), Září 2013. Dostupné z: http://www.ietf.org/rfc/rfc7011.txt

[25] El-Moussa, F.; Mudhar, P.; Jones, A.: Overview of SIP attacks and countermeasures. In *Information Security and Digital Forensics*, Springer, 2010, s. 82–91.

[26] Endler, D.; Collier, M.: *Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions*. New York, NY, USA: McGraw-Hill, Inc., první vydání, 2007, ISBN 0072263644, 9780072263640, chapter 3.

[27] Fernandez, E.; Pelaez, J.; Larrondo-Petrie, M.: Security Patterns for Voice over IP Networks. In *Computing in the Global Information Technology, 2007. ICCGI 2007. International Multi-Conference on*, March 2007, s. 33–33, doi:10.1109/ICCGI.2007.57.

[28] Goode, B.: Voice over Internet protocol (VoIP). *Proceedings of the IEEE*, ročník 90, č. 9, Sep 2002: s. 1495–1517, ISSN 0018-9219, doi:10.1109/JPROC.2002.802005.

[29] Hsieh, P.: *SuperFastHash*. [cit. 2014-11-18]. Dostupné z: `http://www.azillionmonkeys.com/qed/hash.html`

[30] INVEA-TECH a.s.: *FlowMon sondy*. [cit. 2014-12-19]. Dostupné z: `https://www.invea.com/cs/produkty-sluzby/flowmon/flowmon-sondy`

[31] INVEA-TECH a.s.: *High-Speed Networking and FPGA Solutions*. [cit. 2014-12-19]. Dostupné z: `https://www.invea.com`

[32] Iossifov, V.; Totev, T.; Tochatschek, A.: Experiences in VoIP Telephone Network Security Policy at the University of Applied Sciences (FHTW) Berlin. In *Proceedings of the 2007 International Conference on Computer Systems and Technologies*, CompSysTech '07, New York, NY, USA: ACM, 2007, ISBN 978-954-9641-50-9, s. 3:1–3:18, doi:10.1145/1330598.1330602. Dostupné z: `http://doi.acm.org/10.1145/1330598.1330602`

[33] Kulkarni, S. R.; Khasnis, C.: *VoIP Cell Phones: Security concerns and Countermeasures*. 2012, [cit. 2014-12-12]. Dostupné z: `http://www.ijser.org/researchpaper/VoIP-Cell-Phones-Security-concerns-and-Countermeasures.pdf`

[34] MaxMind: *GeoIP Products – Maxmind Developer Site*. [cit. 2014-11-19]. Dostupné z: `http://dev.maxmind.com/geoip/`

[35] MaxMind: *MaxMind - IP Geolocation and Online Fraud Prevention*. [cit. 2014-11-19]. Dostupné z: `http://www.maxmind.com`

[36] O2 Czech Republic a.s.: *O2 | Ceník*. [cit. 2014-12-08]. Dostupné z: `http://www.o2.cz/osobni/219236-cely_cenik/`

[37] Pagh, R.; Rodler, F. F.: Cuckoo Hashing. 2001. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.25.4189&rep=rep1&type=pdf

[38] Peterson, J.: A Privacy Mechanism for the Session Initiation Protocol (SIP). RFC 3323 (Proposed Standard), Listopad 2002. Dostupné z: http://www.ietf.org/rfc/rfc3323.txt

[39] Rosa, Z.: *Detekce síťových tunelů v počítačových sítích*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2014.

[40] Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; aj.: SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), Červen 2002, updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141, 6665, 6878. Dostupné z: http://www.ietf.org/rfc/rfc3261.txt

[41] Schulzrinne, H.: The tel URI for Telephone Numbers. RFC 3966 (Proposed Standard), Prosinec 2004, updated by RFC 5341. Dostupné z: http://www.ietf.org/rfc/rfc3966.txt

[42] Unuth, N.: *Functions of a PBX - Main Technical Roles*. [cit. 2014-11-20]. Dostupné z: http://voip.about.com/od/hardware/a/functionsPBX.htm

[43] Varshney, U.; Snow, A.; McGivern, M.; aj.: Voice over IP. *Commun. ACM*, ročník 45, č. 1, Leden 2002: s. 89–96, ISSN 0001-0782, doi:10.1145/502269.502271. Dostupné z: http://doi.acm.org/10.1145/502269.502271

[44] Česká televize: *Pozor na hackery! Za tři dny můžete provolat přes milion korun*. 2009, [cit. 2014-10-23]. Dostupné z: http://www.ceskatelevize.cz/ct24/domaci/71247-pozor-na-hackery-za-tri-dny-muzete-provolat-pres-milion-korun/

[45] Vojtěch, J.: *Měření VoIP provozu pomocí IP toků [online]*. Diplomová práce, Masarykova univerzita, Fakulta informatiky, 2014 [cit. 2014-10-02]. Dostupné z: http://is.muni.cz/th/325458/fi_m/

[46] Vozňák, M.: *Security and Frauds in IP telephony*. VŠB - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, [cit. 2014-11-22]. Dostupné z: http://www.ip-telefon.cz/data/downld/20.pdf

Příloha A

Typický průběh zkoušení předvolby útočníky

Průběh útoku, který začal 25. 11. 2014, ilustruje typický průběh zaznamenaného zkoušení předvolby. INVITE požadavky byly zasílány v počtu 2-3 za 1 hodinu:

```
2014-11-25 01:25:41 - volání na: "009725925779560X.X.X.X"
2014-11-25 01:44:19 - volání na: "0009725925779560X.X.X.X"
2014-11-25 02:05:08 - volání na: "9009725925779560X.X.X.X"
2014-11-25 02:26:11 - volání na: "+972592577956@X.X.X.X"
2014-11-25 02:47:09 - volání na: "972592577956@X.X.X.X"
2014-11-25 03:08:15 - volání na: "1009725925779560X.X.X.X"
2014-11-25 03:29:17 - volání na: "8009725925779560X.X.X.X"
2014-11-25 03:50:14 - volání na: "6009725925779560X.X.X.X"
2014-11-25 04:11:24 - volání na: "7009725925779560X.X.X.X"
2014-11-25 04:32:54 - volání na: "4009725925779560X.X.X.X"
2014-11-25 04:54:20 - volání na: "3009725925779560X.X.X.X"
2014-11-25 05:15:33 - volání na: "2009725925779560X.X.X.X"
2014-11-25 05:36:41 - volání na: "5009725925779560X.X.X.X"
2014-11-25 05:58:13 - volání na: "999009725925779560X.X.X.X"
2014-11-25 06:19:37 - volání na: "9999009725925779560X.X.X.X"
2014-11-25 06:40:51 - volání na: "99999009725925779560X.X.X.X"
2014-11-25 07:02:18 - volání na: "999999009725925779560X.X.X.X"
2014-11-25 07:23:28 - volání na: "9999999009725925779560X.X.X.X"
2014-11-25 07:44:31 - volání na: "99999999009725925779560X.X.X.X"
2014-11-25 08:05:51 - volání na: "999999999009725925779560X.X.X.X"
2014-11-25 08:27:19 - volání na: "9999999999009725925779560X.X.X.X"
2014-11-25 08:48:41 - volání na: "90009725925779560X.X.X.X"
2014-11-25 09:09:56 - volání na: "0972592577956@X.X.X.X"
2014-11-25 09:31:28 - volání na: "00009725925779560X.X.X.X"
2014-11-25 10:36:02 - volání na: "000000097259257795 6@X.X.X.X"
2014-11-25 10:57:46 - volání na: "0000000097259257795 6@X.X.X.X"
2014-11-25 11:19:20 - volání na: "00000000097259257795 6@X.X.X.X"
```

```
2014-11-25 11:40:53 - volání na: "00000000009725925779956@X.X.X.X"
2014-11-25 12:24:11 - volání na: "910009725925779956@X.X.X.X"
2014-11-25 12:45:34 - volání na: "99009725925779956@X.X.X.X"
2014-11-25 13:04:50 - volání na: "91009725925779956@X.X.X.X"
2014-11-25 13:25:14 - volání na: "92009725925779956@X.X.X.X"
2014-11-25 14:07:12 - volání na: "97009725925779956@X.X.X.X"
2014-11-25 14:27:48 - volání na: "95009725925779956@X.X.X.X"
2014-11-25 15:09:23 - volání na: "98009725925779956@X.X.X.X"
2014-11-25 15:30:22 - volání na: "66009725925779956@X.X.X.X"
2014-11-25 15:51:28 - volání na: "55009725925779956@X.X.X.X"
2014-11-25 16:33:12 - volání na: "33009725925779956@X.X.X.X"
2014-11-25 16:54:05 - volání na: "22009725925779956@X.X.X.X"
2014-11-25 17:14:56 - volání na: "11009725925779956@X.X.X.X"
2014-11-25 17:36:00 - volání na: "09009725925779956@X.X.X.X"
2014-11-25 17:57:09 - volání na: "01009725925779956@X.X.X.X"
2014-11-25 18:38:47 - volání na: "9993009725925779956@X.X.X.X"
2014-11-25 18:59:43 - volání na: "9980097259257956@X.X.X.X"
2014-11-25 19:20:34 - volání na: "9930097259257956@X.X.X.X"
2014-11-25 20:02:45 - volání na: "7100972592577956@X.X.X.X"
2014-11-25 20:45:06 - volání na: "4100972592577956@X.X.X.X"
2014-11-25 21:06:29 - volání na: "8100972592577956@X.X.X.X"
2014-11-25 21:27:49 - volání na: "0800972592577956@X.X.X.X"
2014-11-25 21:48:29 - volání na: "6000972592577956@X.X.X.X"
2014-11-25 22:08:47 - volání na: "8800972592577956@X.X.X.X"
2014-11-25 22:28:31 - volání na: "7700972592577956@X.X.X.X"
2014-11-25 22:47:55 - volání na: "9811009725925779956@X.X.X.X"
2014-11-25 23:07:22 - volání na: "9811009725925779956@X.X.X.X"
2014-11-25 23:26:34 - volání na: "098009725925779956@X.X.X.X"
2014-11-25 23:44:48 - volání na: "9093009725925779956@X.X.X.X"
2014-11-26 00:01:13 - volání na: "9911009725925779956@X.X.X.X"
2014-11-26 00:17:13 - volání na: "7788009725925779956@X.X.X.X"
2014-11-26 00:32:43 - volání na: "9977009725925779956@X.X.X.X"
```

Příloha B

Doporučený postup pro zprovoznění modulu

Prvním krokem by měla být kontrola konfigurace modulu v souboru configuration.h Zde mohou být ovlivněny:

- výchozí hodnoty modulu,
- nastavení využívaných databází pro geolokaci IP adres (odlišné databáze pro IPv4 a IPv6),
- interval provádění kontroly paměti modulu,
- vstupní a výstupní šablona pro příjem a zasílání UniRec položek v systému Nemea,
- formát data a času používaných modulem,
- nastavení návratových hodnot procesu modulu při chybě a úspěšném ukončení,
- považování úspěšně navázaného hovoru až po přijmutí ACK požadavku (nutné zvážit na základě struktury monitorované sítě a rozložení monitorovacích sond),
- nastavení ohledně velikostí jednotlivých dočasných pamětí a určení maximální možné délky používaných textových řetězců,
- interval ukládání informací o IP adresách a zemích (uskutečněných hovorech) do definovaného souboru,
- výpis informací o neplatných SIP hlavičkách,
- zapnutí nebo vypnutí DEBUG režimu.

Pro úspěšnou kompilaci modulu je vyžadována knihovna TRAP (libtrap), UniRec a Nemea-common. Zdrojové kódy aktuálních verzí těchto knihoven jsou přiloženy na CD. Ke stažení jsou také k dispozici na webových stránkách Nemea frameworku [17]. Mimo uvedených knihoven je dále vyžadován gcc, make a pkg-config.

Pro plnou funkčnost je nutná nainstalovaná podpora pro GeoIP (libgeoip-dev) v systému již během kompilace modulu ze zdrojových kódů, v opačném případě bude modul zkompilován bez podpory detekce volání do jiné země (jiná funkce modulu není tímto ovlivněna).

Samotnou kompilaci a instalaci modulu lze provést pomocí příkazů:

```
tar -zxf voip_fraud_detection-1.0.0.tar.gz
cd voip_fraud_detection
./configure --prefix=[cílové umístění instalace]
make install
```

Doporučovaný postup při nasazování modulu je následující:

1. Vhodně si určit parametry spouštěného modulu v závislosti na velikosti monitorované sítě, počtu odhadovaných zařízení s VoIP provozem a velikosti paměti serveru, na kterém bude modul spuštěn.

2. Spuštění modulu v režimu učení po definovaný čas pro detekování volání do jiné země (např. po dobu 14 dní).

3. Ukončení modulu.

4. Vyhodnocení výsledků detekcí pro hledání předvoleb útočníky a případné úpravy parametrů modulu pro další spuštění.

5. Kontrola `countries` souboru (název souboru je závislý na konfiguraci), zda obsahuje očekávané hodnoty (úspěšně navázané hovory pro dané IP adresy do daných cílových zemí). Možno přidat globálně povolené cílové země nebo cílové země pro určité IP adresy, do kterých se obvykle volá. V případě nalezení neobvyklých a podezřelých záznamů je vhodné tyto záznamy odstranit.

6. Spuštění modulu (s upravenými parametry) v ostrém provozu.

7. Sledování výstupů z modulu a řešení nalezených incidentů v síti.

Příloha C

Parametry při spouštění modulu

Při spouštění modulu je nutné určit vstupní a výstupní rozhraní modulu, které je obsluhováno TRAP knihovnou. Rozhraním může být TCP/IP socket (`t`) nebo Unix domain socket (`u`). U výstupního rozhraní je možnost použít navíc typ „blackhole“ (`b`), který zajistí, že výstupní informace nejsou nikam zasílány a jsou zahozeny. Rozhraní jsou definovány pomocí parametru `-i` a následnou specifikací vstupního a výstupního rozhraní. Specifikace je určena zřetězením následujících položek:

- typ vstupního rozhraní
- typ výstupního rozhraní
- (oddělovač) `;`
- parametry vstupního rozhraní
- (oddělovač) `;`
- parametry výstupního rozhraní

Pokud chceme např. přijímat data na TCP/IP portu 7601 a odesílat hlášení o detekcích na Unix domain socket 11001 použijeme parametr `-i` následovně:

```
-i "tu;7601;11001"
```

Pokud bychom nechtěli zasílat hlášení o detekcích na výstupní rozhraní, použili bychom:

```
-i "tb;7601;"
```

Definování ostatních parametrů je volitelné a ovlivňuje chování samotného modulu. Všechny volitelné parametry jsou uvedeny v tabulce C.1.

Tabulka C.1: Volitelné parametry modulu pro detekci zneužití VoIP ústředen

parametr	**popis**	**výchozí hodnota**
-h	vypsání nápovědy k modulu	—
-l	cesta k log souboru	—
-e	cesta k `event_id` souboru (zakázání používání souboru lze provést zadáním hodnoty „disabled“), soubor obsahuje identifikaci posledního nahlášeného útoku	event_id
-c	cesta ke `countries` souboru (zakázání používání souboru lze provést zadáním hodnoty „disabled“)	countries.dat
-o	vypnutí detekce volání do jiných zemí	—
-a	nastavení učícího módu pro detekci volání do jiných zemí po nastavenou dobu v sekundách	1209600
-w	zakázání ukládání nových zemí do seznamu povolených zemí po nahlášení upozornění (každé volání do jiné země bude opětovně nahlašováno)	—
-m	maximální délka zkoumané předvolby	10
-d	minimální délka volaného čísla, která je kontrolována modulem	0
-s	interval v sekundách, po kterém probíhá vyhodnocení dat pro danou IP adresu, lze ovlivnit především celkovou zátěž procesoru po delší časový úsek	10
-t	mez počtu unikátních předvoleb pro jednu IP adresu, detekce je vyvolána při překročení této hodnoty	10
-p	pauza v sekundách po detekování události pro danou IP adresu, tímto nastavením lze ovlivnit minimální časový interval detekcí v případě dlouhotrvajícího intenzivního útoku	30
-q	limit maximálního počtu ukládaných volaných čísel pro jednu IP adresu	100000
-x	čas vyjádřený v sekundách, po který se nechávají uložené informace k IP adrese bez komunikace	1209600

Ukázka příkazu pro spuštění modulu se všemi volitelnými parametry:

```
./voip_fraud_detection -i "tu;7601;11001" -h -l /data_voip/log
-e /data_voip/event_id -c /data_voip/countries.dat -o -a 600 -w
-m 4 -d 2 -s 30 -t 20 -p 60 -q 10000 -x 604800
```

Příloha D

Popis položek zasílaných do výstupního rozhraní modulu

Položky zasílané do výstupního rozhraní modulu (např. modulu Report Handler, který ukládá události do databáze) závisí na typu odesílané události (pokus o nalezení předvolby útočníky nebo volání do jiné země). Popis jednotlivých položek, které se zasílají je následující:

- `EVENT_ID` – ID detekované události
- `EVENT_TYPE` – identifikace typu události v systému Nemea (použity předdefinované konstanty):
  - `EVT_T_VOIP_PREFIX_GUESS` – pokus o nalezení předvolby útočníky
  - `EVT_T_VOIP_CALL_DIFFERENT_COUNTRY` – volání do jiné země
- `SRC_IP` – zdrojová adresa, ze které pochází útok nebo volání do jiné země
- `DST_IP` – cílová adresa, na který směroval požadavek v rámci volání do jiné země
- `DETECTION_TIME` – čas detekce
- `TIME_FIRST` – začátek sledování SIP komunikace dané IP adresy (čas prvního INVITE požadavku)
- `VOIP_FRAUD_SIP_TO` – jedna ze SIP_TO hlaviček, které byly vyhodnoceny jako pokusy o útok (je vybrána z množiny s nejdelší zkoušenou předvolbou) nebo SIP_TO hlavička z požadavku na volání do jiné země
- `VOIP_FRAUD_SIP_FROM` – SIP_FROM hlavička, ze které byl zahájen hovor do jiné země

- `VOIP_FRAUD_USER_AGENT` – identifikace zařízení nebo softwaru, který vygeneroval požadavek na volání
- `VOIP_FRAUD_PREFIX_LENGTH` – délka předvolby z `VOIP_FRAUD_SIP_TO`, v rámci které byly zkoušeny různé předvolby
- `VOIP_FRAUD_PREFIX_EXAMINATION_COUNT` – počet unikátních SIP_TO, které byly vyhodnoceny jako pokusy o útok
- `VOIP_FRAUD_SUCCESSFUL_CALL_COUNT` – počet úspěšně spojených hovorů na unikátní čísla v rámci zkoumaných předvoleb
- `VOIP_FRAUD_INVITE_COUNT` – celkový počet zachycených INVITE požadavků v rámci zkoumaného prefixu
- `VOIP_FRAUD_COUNTRY_CODE` – identifikace země, do které bylo voláno, dle standardu ISO 3166 s označením země právě dvěma znaky

Příloha E

Seznam použitých zkratek

API Application Programming Interface

ARP Address Resolution Protocol

AS Autonomní systém

CDN Content Delivery Network

DNS Domain Name System

DHCP Dynamic Host Configuration Protocol

FQDN Fully Qualified Domain Name

IETF Internet Engineering Task Force

IP Internet Protocol

IPFIX Internet Protocol Flow Information Export

HTTP Hypertext Transfer Protocol

MAC Media Access Control

MGCP Media Gateway Control Protocol

mVoIP mobile VoIP

Nemea Network Measurement Analysis

NREN National Research and Education Network

PC Personal Computer

PSTN Public Switched Telephone Network

QoS Quality of Service

RFC Request for Comments

RTP Real-time Transport Protocol

RTCP Real-time Transport Control Protocol

S/MIME Secure/Multipurpose Internet Mail Extensions

SIP Session Initiation Protocol

SNMP Simple Network Management Protocol

SRTP Secure Real-time Transport Protocol

TCP Transmission Control Protocol

TFTP Trivial File Transfer Protocol

TLS Transport Layer Security

TTL Time to live

TRAP Traffic Analysis Platform

UAC User Agent Client

UAS User Agent Server

UDP User Datagram Protocol

UniRec Unified Record

VoIP Voice over Internet Protocol

VLAN Virtual Local Area Network

VPN Virtual Private Network

Příloha F

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `doc` ..... dokumentace k detekčnímu modulu
  - `html` ..... dokumentace ve formátu HTML
  - `pdf` ..... dokumentace ve formátu PDF
- `lib` ..... adresář obsahující knihovny Libtrap, UniRec a Nemea-common
- `src`
  - `impl` ..... balík se zdrojovými kódy implementace detekčního modulu
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `svwar` ..... adresář obsahující upravený nástroj `svwar.py` *(SIPVicious)*
- `text` ..... text práce
  - `DP_Truxa_Lukas_2015.pdf` ..... text práce ve formátu PDF