Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Nasazení systému LaBrea6 do reálného provozu v síti IPv6

Plzeň 2013 Martin Čížek

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů. Nemám námitek proti půjčení práce se souhlasem katedry ani proti zveřejnění práce nebo její části.

V Plzni dne 14. května 2013

Martin Čížek

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé diplomové práce Ing. Radoslavu Bodó za poskytnutí odborných rad, věcné připomínky a ochotu při zpracování této práce.

Abstract

The main purpose of this diploma thesis is to create suitable IPv6 network with all main characters of a real IPv6 network. This new network will be used for deploying the system LaBrea6 which has been created as a previous bachelor thesis.

The new IPv6 network should also cointain a common aplications and services which should help to make the network more attractive for attackers. All systems in the network will log all needed information about attacks in the network for subsequent analysis, which would help with preventive defense of the Center for Information Technology (CIV) at University of West Bohemia.

Obsah

1 Úvod

Stávající aplikace *LaBrea6*, která vznikla jako bakalářská práce studijního oboru FAV/KIV/INIB, byla vytvořena za účelem získání užitečných informací o útocích v tomto druhu sítě. V současné době CIV provozuje IDS systém *LaBrea* na síti WEBnet na protokolu IPv4. Předpokládáme-li přechod k síti na protokolu IPv6, bylo by zajímavé shromáždit informace o současných útocích v tomto druhu sítě, které by pomohly s preventivní obranou.

Cílem této práce je navrhnout a vytvořit IPv6 síť, která má za úkol zatraktivnit adresní prostor systému *LaBrea6* pro přilákání útočníků na tomto druhu sítí. Navržená IPv6 síť by měla obsahovat všechny znaky reálných IPv6 sítí včetně vhodně zvolených služeb, které budou v síti provozovány spolu se systémem *LaBrea6*. Takto navržená síť by měla v konečném důsledku zatraktivnit adresní prostor systému *LaBrea6* a přilákat útočníky pro následnou analýzu případných útoků.

Práce je organizována do následujících částí. V první části jsou shrnuty základní rozdíly mezi protokoly IPv4 a IPv6, které je potřeba v průběhu práce brát v úvahu. Dále následuje popis základní funkcionality IDS systému *LaBrea6*. V práci jsou podrobně popsány techniky objevování cílových stanic v sítích IPv6 a tyto techniky následně použity pro návrh IPv6 sítě. Následuje raelizace navržené IPv6 sítě a nasazení všech systémů do reálného provozu. Práce je zakončena analýzou útoků, které byly provedeny v této navržené síti.

2 IPv6 - Základní rozdíly oproti IPv4

Pro implementaci projektu bylo potřeba vzít v úvahu změny mezi protokoly. IPv6 ve velkém rozsahu zachovává základní funkcionalitu a pravidla protokolu IPv4. Jsou zde ovšem změny, které se naprosto liší od koncepce protokolu IPv4 a pro následující práci jsou důležité.

Větší adresní prostor

Adresy protokolu IPv6 jsou dlouhé 128 bitů, oproti 32 bitům u protokolu IPv4. To znamená, že pomocí protokolu IPv6 je možno adresovat až 2^{128} (3.4 * 10^{38}) síťových zařízeních. Tento fakt však vede k myšlence, zda je vůbec technicky možné provádět skenování tak velkého adresního prostoru pomocí známých technik. Tato problematika bude dále probírána v kapitole 4.

Druhy IPv6 adres

V IPv6 existují 3 druhy adres s odlišným účelem:

1. Individuální (unicast)
2. Skupinové (muticast)
3. Výběrové (anycast)

Unicast adresy reprezentují jednotlivá síťová rozhraní. Paket zaslaný na unicast adresu je doručen konkrétnímu rozhraní. IPv6 unicast adresy se dále rozdělují do následujících základních typů:

- globální unicast adresy
- lokální unicast adresy
- speciální adresy

Multicast adresy jsou používány k definování množiny rozhraní obvykle patřících různým uzlům, nikoli pouze jednomu. Paket zaslaný na multicast

adresu je protokolem doručen všem rozhraním určeným touto adresou. Multicast adresy mají prefix FF00::/8 a jejich druhý oktet určuje dosah adresy, tzn. rozsah v jakém je multicast adresa zviditelněna. Běžně využívány jsou rozsahy lokální (FF02::/8), globální (FF0E::/8) a rozsahy místní (FF05::/8).

IPv6 skupinové (multicast) adresy zcela nahradily adresy oznamovací (broadcast). Stejného výsledku lze totiž dosáhnout pomocí skupiny *all-hosts* (ff02::1) [1].

Anycast adresy jsou také přiřazeny více než jednomu rozhraní, patřící rozdílným uzlům. Paket vyslaný na anycast adresu je doručen pouze jednomu z členských rozhraní, typicky „nejbližšímu“ vzhledem k informacím směrovacího protokolu o vzdálenosti jednotlivých uzlů [1].

IPv6 vs. IPv4 záhlaví

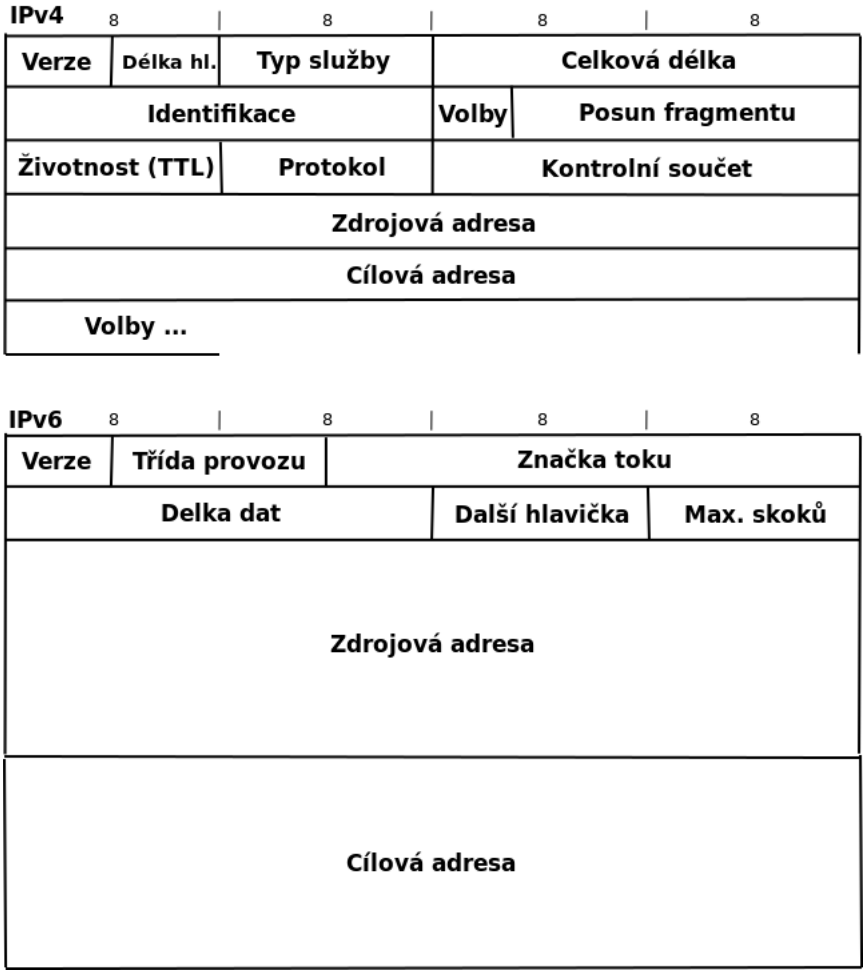
Další změna se týká IPv6 hlavičky (viz Obr. 2.1). Hlavní myšlenkou bylo vynechání nepotřebných položek pro přenos IPv6 paketu. Tato změna omezuje přenášení nadbytečných informací, které nejsou důležité pro přenos a svojí existencí zvyšují režii přenosu. Oproti protokolu IPv4 má IPv6 hlavička konstantní velikost 40B. Položky *Identifikace* (Identification), *Volby* (Flags) a *Posun fragmentu* (Fragment Offset) jsou v IPv6 nahrazeny rozšiřujícími hlavičkami (viz kapitola 2). Položka *Kontrolní součet* (Checksum) je v IPv6 hlavičce zcela vynechána a spoléhá se na kontrolu kontrolním součtem od vyšších vrstev [1].

Autokonfigurace adres

Host připojen do IPv6 síťě může být automaticky nakonfigurován dvěma způsoby:

- Bezstavová autokonfigurace - ICMPv6 (základní)
- Stavová autokonfigurace - DHCPv6 (volitelná)

Bezstavová autokonfigurace představuje zcela nový způsob založen na použití zpráv směrem od ICMPv6 serveru (směrovače). Při prvním připojení k IPv6 síti si host vygeneruje svou vlastní lokální IPv6 adresu složenou



Obrázek 2.1: Porovnání hlaviček IPv4 a IPv6

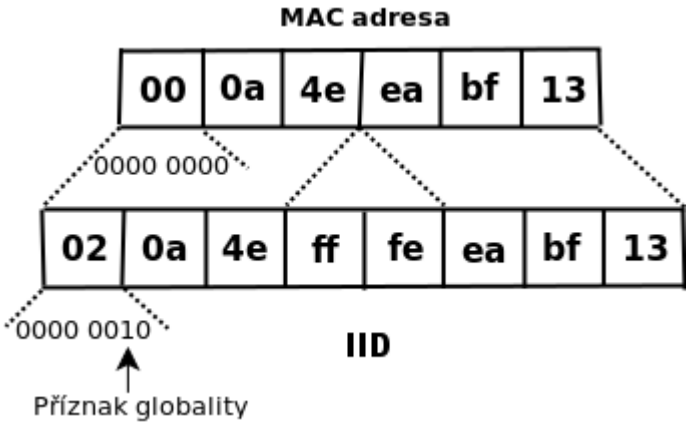
z prefixu (fe80::/10) a 64-bitového identifikátoru rozhraní (IID) vycházejícího z MAC adresy hosta. Host dále čeká na *Ohlášení Směrovače* (router advertisement) s konfiguračními parametry sítě, případně si o tyto parametry sám aktivně požádá prostřednictvím *Výzvy Směrovači* (Router Solicitation). ICMPv6 směrovač odpoví na tuto žádost paketem *Ohlášení Směrovače*, který obsahuje konfigurační parametry síťové vrstvy včetně prefixu IPv6 sítě. Host si po obdržení této konfigurační zprávy vygeneruje IPv6 adresu z přijatého prefixu IPv6 sítě a IID identifikátoru rozhraní.

Stavová autokonfigurace je v IPv6 prováděna pomocí protokolu DHCPv6. Jako u jeho předchůdce v IPv4, DHCPv6 server půjčuje IPv6 adresy nově připojeným hostům z přiděleného rozsahu. Host po připojení rozešle do IPv6 sítě na obecnou adresu všech DHCPv6 serverů(ff02::1:2) dotaz ohledně svých

konfiguračních parametrů pro komunikaci a DHCPv6 server mu je ve své odpovědi sdělí.

Lokální adresy

IPv6 má kromě globálních adres také adresy lokální. Začínají prefixem fe80::/10. Následujících 54 bitů je nulových, za nimi následuje IID identifikátor. IID identifikátor je automaticky odvozen od MAC adresy rozhraní hosta. Odvození IID identifikátoru rozhraní se provádí přidáním hexadecimální hodnoty `fffe` do středu původní MAC adresy (mezi třetí a čtvrtý bajt MAC adresy) a navíc se obrátí příznak globality[1] (viz Obr. 2.2) [1].



Obrázek 2.2: Identifikátor rozhraní (IID) podle modifikovaného EUI-64

Linkové adresy jsou vždy k dispozici a mají jednoznačně dané schéma pro jejich vytváření, což zjednodušuje vývoj konfiguračních a směrovacích protokolů. Přítomnost lokálních IPv6 adres umožňuje okamžitě po zapojení do sítě komunikovat se zařízením na stejné lince bez potřeby získávání globální IPv6 adresy od automatické konfigurace (např. DHCPv6). [1].

Nepřítomnost kontrolního součtu

Kontrolní součet je u protokolu IPv4 odvozen z celé hlavičky IPv4 paketu. Jelikož se některá z polí IPv6 hlavičky mění při přenosu paketu (např. TTL)

[1]Předposlední (druhý nejméně významný) bit v nejvyšším bajtu IID identifikátoru slouží jako příznak globality. Ve standardním EUI-64 zde hodnota 0 signalizuje celosvětově jednoznačnou adresu, zatímco 1 označuje adresu lokální.

na cestě mezi jednotlivými směrovači, musel by se kontrolní součet znovu přepočítávat při každé změně na jednotlivých směrovačích. To by znamenalo zvýšení režie při přepočtu a následné zpomalení přenosu. V dnešních sítích se takové chyby považují za vzácné. Z tohoto důvodu IPv6 nemá žádnou kontrolu chyb. Namísto toho se spoléhá s kontrolou chyb na protokoly vyšších vrstev [1].

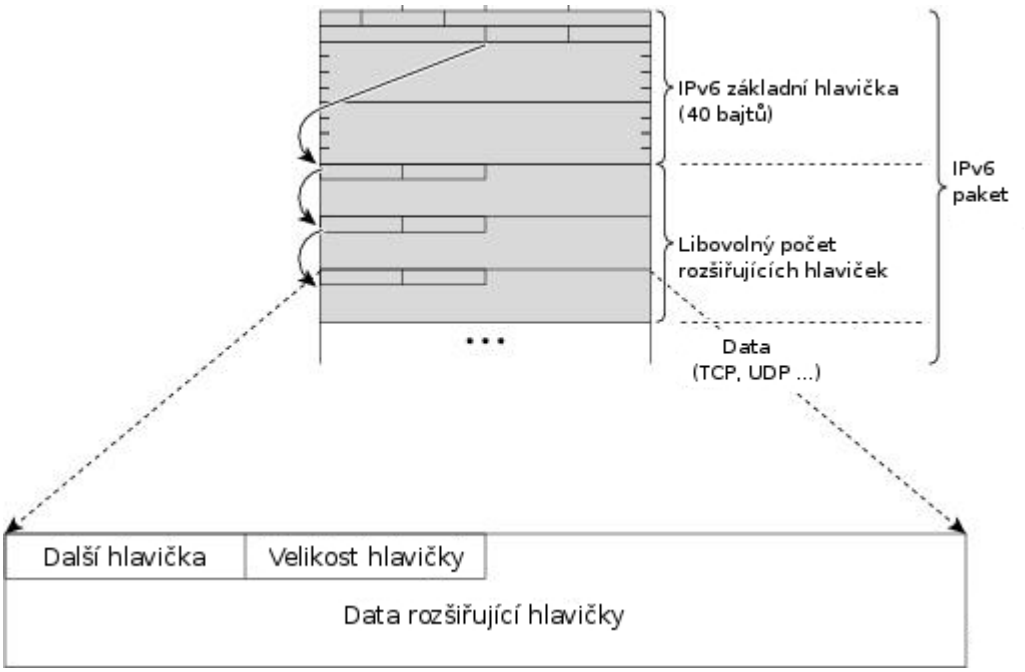
Položka „Další hlavička“

Položka *Další hlavička* (*Next Header*) určuje jaký typ hlavičky následuje za základním IPv6 záhlavím. Rozšiřující hlavičky IPv6 záhlaví jsou pomocí položky *Další hlavička* dále řetězeny za sebou až do hlavičky protokolu vyšší vrstvy (viz Obr. 2.3). Toto zřetězení slouží k tomu, aby zařízení, které paket zpracovává, přeskočilo nepotřebné rozšiřující informace a dostalo se k informaci, kterou pro následné zpracování potřebuje. Tento systém podporuje efektivitu a rychlost zpracování paketů na jednotlivých uzlech (na rozdíl od protokolu IPv4). Rozšiřující hlavičky mají předepsáno následující pořadí [1]:

1. volby pro všechny (hop-by-hop options)
2. volby pro cíl (destination options) – pro první cílovou adresu datagramu a případné další uvedené v hlavičce Směrování
3. směrování (routing)
4. fragmentace (fragment)
5. autentizace (authentication)
6. šifrování obsahu (encapsulating security payload)
7. volby pro cíl (destination options) – pro konečného příjemce datagramu
8. mobilita (mobility)

ICMPv6 vs. ARP

Další zásadní změnou prokolu IPv6 je absence protokolu ARP pro zjišťování sousedních uzlů. Veškerá funkcionalita protokolu ARP je v IPv6 implementována protokolem ICMPv6:



Obrázek 2.3: Položka *Další hlavička*

- Zjišťování MAC adres uzlů v lokální síti.
- Rychlé aktualizace neplatných položek a zjišťování změn v MAC adresách.
- Hledání směrovačů.
- Přesměrování.
- Zjišťování prefixů, parametrů sítě a dalších údajů pro automatickou konfiguraci adresy.
- Ověřování dosažitelnosti sousedů.
- Detekce duplicitních adres.

Nutným rozšířením protokolu ICMPv6 je *bezpečné objevování sousedů* (Secure Neighbor Discovery, SEND), jako zabezpečení proti tzv. *arp cache poisoning* útokům, které útočníkovi umožnují vydávat se v místní síti za jiný počítač. Útočník může například odposlouchávat komunikaci mezi dvěma uzly lokální sítě, tím že podstrčí oběma uzlům svojí MAC adresu a přijatá data posílá dál skutečným adresátům. Cílem SENDu je poskytnout

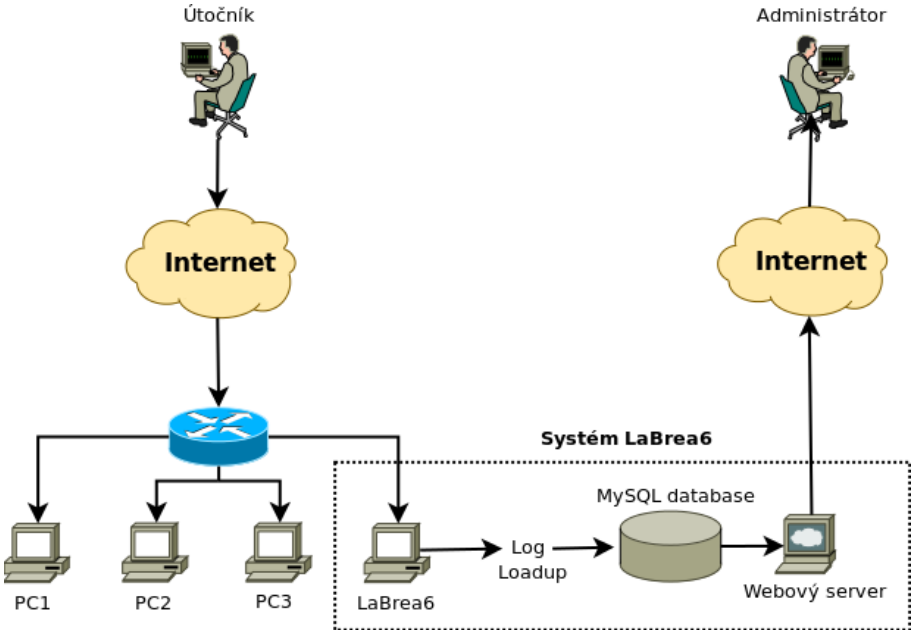
dostatečnou úroveň zabezpečení vyměňovaných zpráv. Zabezpečení spočívá ve využití kryptografických technik (*Kryptograficky Generované Adresy - CGA*) při kontrole autentičnosti ICMPv6 zpráv. Dvě nejdůležitější položky odesílané ICMPv6 zprávy představují *Otisk klíče* (Key hash), jehož prostřednictvím lze identifikovat veřejný klíč pro ověření podpisu, a vlastní *Digitální podpis* (Digital signature). Po příchodu paketu se podepsaná zpráva ověří. Poslouží k tomu veřejný klíč identifikovaný svým otiskem. Pokud digitální podpis odpovídá, zpráva je považována za bezpečnou. V opačném případě, kdy je zpráva považována za nebezpečnou, záleží na administrátorovi sítě, jak se zachová při příchodu nebezpečné ICMPv6 zprávy. Při akceptování nebezpečné zprávy mají však větší prioritu ty bezpečné. Nevýhodou rozšíření SEND je nutnost, aby každý uzel vlastnil klíč/certifikát nutný k ověření podpisu. To však zvyšuje nároky při správě sítě a režii při přenosech [1].

3 LaBrea6

LaBrea6 je IDS systém vytvořený v rámci bakalářské práce studijního oboru FAV/KIV/INIB na Západočeské Univerzitě v Plzni vycházející ze systému LaBrea. Původní systém LaBrea byl implementován jen pro nasazení na sítě IPv4. Systém *LaBrea6* je volným rozšířením tohoto projektu a byl implementován pro použití v sítích IPv6.

3.1 Základní funkcionalita

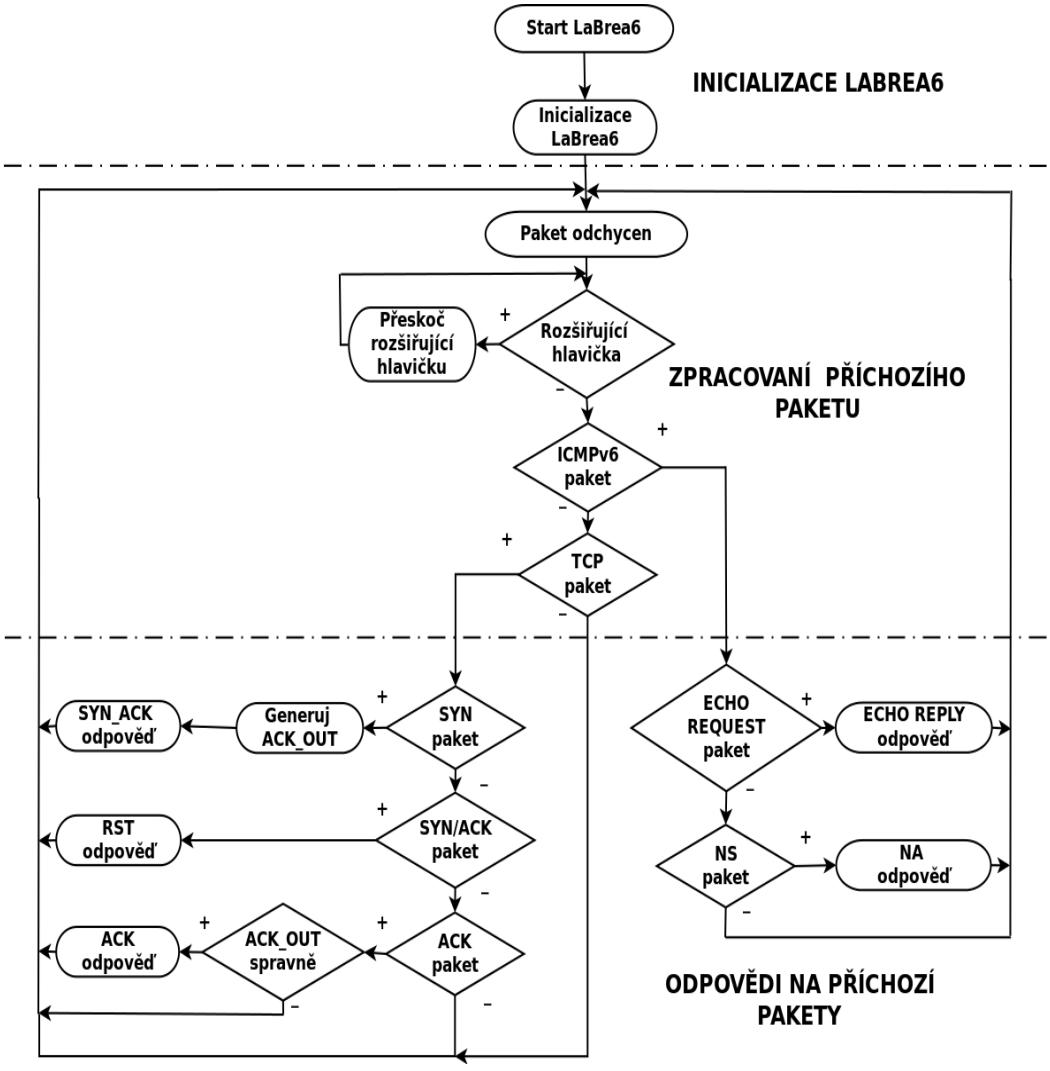
Systém *LaBrea6* běží na počítači zapojeném do IPv6 sítě spolu s dalšími reálnými počítači (viz obrázek 3.1). Pokud se útočník pokusí navázat spojení s IPv6 adresou, *LaBrea6* útočníkovi odpoví specifickým způsobem, který vychází z původního systému LaBrea (viz kapitola 3.1.1). Veškerá činnost systému *LaBrea6* je zaznamenána do *MySQL* databáze. Data v databázi jsou zobrazována přes webové rozhraní a administrátor z těchto dat dostává denní souhrnné reporty o útocích.



Obrázek 3.1: Architektura systému LaBrea6

3.1.1 Zpracování příchozích paketů

Po spuštění systému *LaBrea6* proběhne inicializace systému s nastavením specifikované vstupními parametry a systém začne poslouchat na přiděleném adresním prostoru. Každý příchozí paket je parsován a pokud je správně rozpoznán, *LaBrea6* vytvoří příslušný paket s odpovědí a tento paket zašle zpět příjemci (viz obrázek 3.2).



Obrázek 3.2: Vývojový diagram

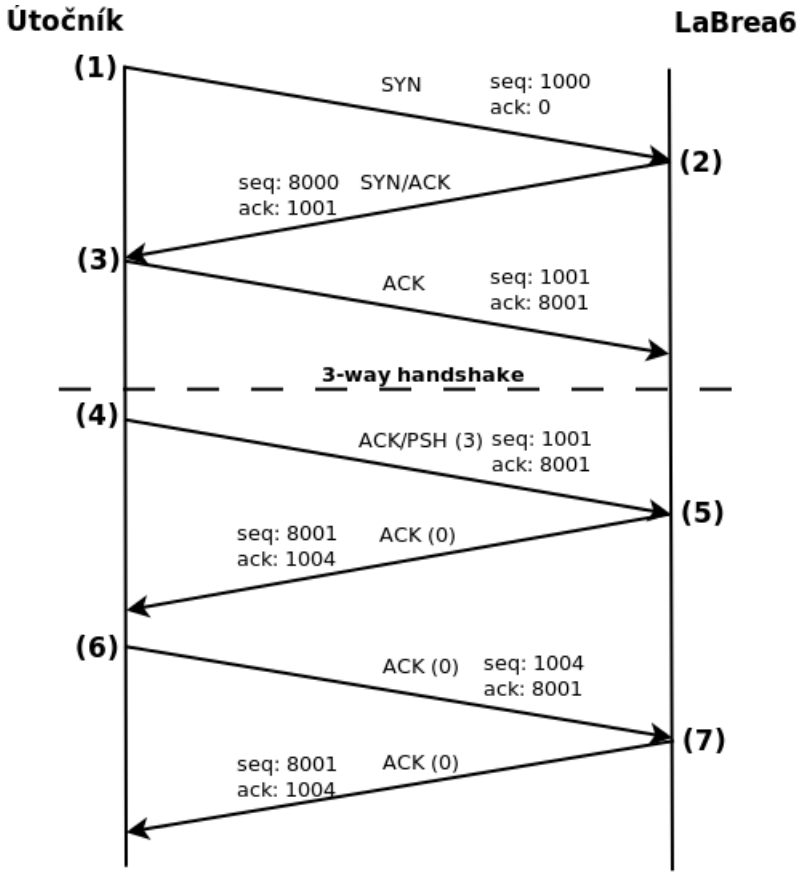
Systém *LaBrea6* dokáže odpovídat na dva základní typy paketů:

- ICMPv6
  - NEIGHBOR:SOLICITATION paketem NEIGHBOR:ADVERTISEMENT s MAC adresou stroje, na kterým LaBra6 běží.
  - ECHO:REQUEST paketem ECHO:REPLY.
- TCP
  - TCP:SYN paketem TCP:SYN/ACK s **velikostí okna 3**.
  - TCP:ACK paketem TCP:ACK s **velikostí okna 0**.
  - TCP:SYN/ACK paketem TCP:RST s **velikostí okna 3**.

Obsluha žádosti o TCP spojení

Pro správnou funkci nestačí aby *LaBrea6* „hloupě“ odpovídala na každý příchozí paket s předem definovanou odpovědí, jako je to v případě odpovědí na ICMPv6 pakety. TCP komunikace vyžaduje jistou logiku, která kopíruje chování reálného počítače (viz Obr. 3.3).

Navázání TCP spojení začíná tzv. *3-way handshakem*. Ten, kdo iniciuje připojení, v našem případě útočník, vyšle příjemci (*LaBrea6*) TCP paket SYN **(1)**. *LaBrea6* odpoví paketem SYN/ACK **(2)**, kterým potvrdí útočníkův přijatý SYN paket. Tento paket sebou nese položku *Velikost okna* (Window size). Ta specifikuje kolik bytů je odesílající strana (v našem případě *LaBrea6*) schopna ještě přijmout. Proto se v paketu SYN/ACK vyplní na hodnotu blízkou nule například 3. Tímto dá systém LaBrea najevo, že je momentálně vytížen a tváří se jako zaneprázdněný stroj, který může v následujícím paketu přijmout jen 3 byty dat. Pro dokončení *3-way handshaku* je ještě nutné, aby útočník potvrdil přijetí SYN/ACK paketu paketem ACK **(3)**. Jakmile *LaBrea6* obdrží ACK **(3)** paket od útočníka, na který zásadně neodpovídá, útočníkovi se jeví připojení navázané a bude se snažit posílat data. Jakmile útočník pošle data **(4)** (počet zaslaných dat se odvíjí od hodnoty *Velikost okna*, kterou útočník obdržel v posledním přijatém TCP paketu), *LaBrea6* potvrdí, že přijala 3 bajty dat (specifikované velikostí okna v SYN/-ACK paketu **(2)**) a odpoví paketem ACK **(5)** s velikostí okna 0. Útočníkovi se jeví napadený počítač jako zaneprázdněný. Po určitých intervalech se bude



Obrázek 3.3: TCP komunikace LaBrea6

snažit zaslání dat zopakovat **(6)**, ale systém *LaBrea6* mu bude stále dávat najevo, že je zaneprázdněný **(7)**. Tímto mechanizmem systém *LaBrea6* zdržuje útočníka na dobu potřebnou například k jeho identifikaci.

Nicméně zde nastává problém, jak rozpoznat ACK paket z *3-way handshaku* **(3)** a ACK paket ověřující, zda je spojení stále aktivní **(6)**? Z předchozího odstavce víme, že na ACK paket z *3-way handshaku* **(3)** se neodpovídá, *LaBrea6* ho ignoruje. *LaBrea6* používá mechanizmus, jak provést toto rozlišení, podobný mechanizmu, který používají například *syn cookies*.

V TCP paketu se nacházejí dvě položky:

1. Pořadové číslo odesílaného bajtu (Sequence number)
2. Pořadové číslo přijatého bajtu (Acknowledgment number)

Pořadové číslo přijatého bajtu určuje jaké je pořadové číslo prvního bajtu TCP segmentu v toku dat od odesílatele k příjemci (TCP segment nese bajty od pořadového čísla odesílaného bajtu až do délky segmentu). Naopak pořadové číslo přijatého bajtu vyjadřuje číslo následujícího bajtu, který je příjemce připraven přijmout, tj. příjemce potvrzuje, že správně přijal vše až do pořadového čísla přijatého bajtu mínus jedna. Tyto položky hrají ve zmíněném mechanizmu podstatnou roli.

Když *LaBrea6* obdrží od útočníka SYN paket **(1)**, spočte si speciální číslo tzv. ACK_OUT. Toto číslo je vypočteno a následné zakódováno z příchozího pořadového čísla přijatého bajtu, ke kterému je přičtena hodnota *Velikost okna* (*LaBrea6* používá hodnotu 3). Toto číslo je po svém zakódování použito jako pořadové číslo odesílaného bajtu v SYN/ACK paketu **(2)** odesílaném útočníkovi systémem *LaBrea6*. Po jeho zpětném dekódování reprezentuje pořadové číslo odesílaného bajtu v příchozím ACK **(6)** paketu od útočníka. Tímto způsobem se dá určit ACK paket, na který má *LaBrea6* odpovědět **(6)** a který ignorovat **(3)**.

3.2 Nutné změny pro nasazení

Ve verzi *LaBrea6 v1.0* bylo provedeno základní nasazení samotného systému *LaBrea6*. Systém sám o sobě však neposkytuje útočníkům žádná dodatečná data (emaulace služeb), tak aby se simulovaná síť jevila atraktivní a dostatečně interaktivní pro zachycení útoku.

Toto zjištění z předchozí práce evokovalo myšlenku pro vytvoření prostředí, které by se tvářilo jako reálná IPv6 síť a tím se stalo pro útočníka atraktivním. Kroky, které byly potřeba provést pro vytvoření reálné IPv6 sítě, jsou následující:

- Zjistit informace o možnostech skenování IPv6 sítí.
- Ze získaných informací navrhnout adresní prostor, který se podobá reálnému.
- Podle navrženého adresního prostoru vytvořit generátor konfigurace pro
  - Systém *LaBrea6*

  - DNS server *Bind9*

- Do navržené IPv6 sítě zapojit SSH honeypot s nízkou interakcí.
- Do navržené IPv6 sítě zapojit SMTP honeypot.
- Do navržené IPv6 sítě zapojit webových honeypot.
- Zaindexovat webových honeypot pomocí *Google* vyhledávače.
- Zalinkovat webových honeypot do existujících webových stránek pro tzv. „prohledávací pavouky“.

Pro lepší přehled o možných útocích se předpokládá ukládání informací ze všech nasazených sub-systémů do databáze, následné zpracování a zobrazování statistik o útocích na webovém administračním prostředí.

4 Skenování adresního prostoru IPv6

Pro navrhnutí adresního prostoru pro systém *LaBrea6* bylo potřeba provést analýzu možností skenování IPv6 sítí. Zdálo by se, že díky velikosti adresního prostoru protokolu IPv6 (cca 1.844 * 10^{19} adres) by nebylo možné z důvodu řídkosti koncových strojů a náročnosti na výpočetní výkon provádět útoky založené na skenování koncových stanic v sítích IPv6.

Následující sekce popisují známé techniky, které útočníkovi pomáhají razantně redukovat velikost skenovaného adresního prostoru a tím zvyšovat možnost útoků založených na skenování koncových stanic v sítích IPv6.

4.1 Skenování vzorů IPv6 adres

Tato technika je založena na využití známých vzorů IPv6 adres, které se v praxi běžně používají. Následující odstavce poskytují základní analýzu toho, jakými technikami jsou přidělovány adresy v sítích IPv6 a jaké vzory IPv6 adres z těchto technik vyplývají. Tyto vzory mohou být použity pro redukci prohledávaného adresního prostoru a nalezení vhodných uzlů k provedení útoku.

4.1.1 SLAAC

SLAAC (StateLess Address Auto-Configuration) je technologie založená na jednoduché myšlence. Každé zařízení, které se chce připojit do sítě, vyšle multicast ICMPv6 žádost (Router Solicitation) o konfigurační parametry pro místní síť. ICMPv6 směrovač odpoví na tuto žádost odpovědí (Router Advertisement), který obsahuje konfigurační parametry síťové vrstvy.

V odpovědi (router advertisement) jsou mimo jiných informací o místní síti i informace o IPv6 prefixu, který je dále použit pro konstrukci výsledné IPv6 adresy přidáním lokálně generovaného identifikátoru rozhraní (IID). Techniky pro generování IID identifikátorů jsou následující [13].

IID využívající IEEE identifikátor

Spousta síťových zařízení generuje 64-bitový IID identifikátor za použití linkové adresy použitého síťového rozhraní. Postup při tvorbě IID identifikátoru z linkové adresy rozhraní je následující:

- Univerzální bit (bit 6 zleva doprava) je nastaven na hodnotu 1.
- Slovo `0xfffe` je vloženo mezi jedinečný identifikátor organizace (OUI) a zbytek linkové adresy.

Například, pro linkovou adresu `00:1b:38:83:88:3c` by vypadal IID identifikátor `021b:38ff:fe83:883c`.

Jednou z vlastností tohoto vzoru adres je fakt, že dva bajty (bajt 4 a 5) jsou pro každý IID identifikátor neměnný (`0xff`, `0xfe`), čímž se snižuje adresní prostor. Dalším uvážením je, že první 3 bajty IID identifikátoru odkazují na OUI identifikátor výrobce síťové karty a protože není možné obsadit všechny OUI identifikátory, tento fakt vede k dalšímu redukování adresního prostoru. Dalším uvážením je, že spoustu OUI identifikátorů odkazuje na starší zařízení, které IPv6 protokol vůbec nepodporují. V konečném důsledku můžeme uvažovat případ, kdy útočník zná výrobce síťových karet koncových stanic skenované sítě. V tomto případě se prostor prohledávaných adres redukuje ještě více.

Tyto úvahy vedou v některých případech k redukování vyhledávacího prostoru IID identifikátoru o velikosti 64-bitů až na 2^24.

Dalším zajímavý faktor přichází při použití virtualizačních technologií, které používají automaticky generované MAC adresy podle specifického vzoru. Pro příklad, veškeré generované MAC adresy virtuálních strojů v aplikaci VirtualBox používají OUI identifikátor ve tvaru `08:00:27`. To však v konečné důsledku znamená, že všechny SLAAC přidělované IPv6 adresy budou obsahovat IID identifikátor ve tvaru `a00:27ff:feXX:XXXX`. V tomto případě je redukován vyhledávací prostor z 64-bitového na 24-bitový.

VMWare ESX server nabízí ještě zajímavější příklad. Jejich automaticky generované MAC adresy mají následující vzor:

1 OUI je nastaveno na 00:05:59.

2 Následujících 16 bitů MAC adresy je nastaveno na stejnou hodnotu jako posledních 16 bitů primární IPv4 adresy operačního systému.

3 Posledních 8 bitů MAC adresy jsou nastaveny na hash hodnotu vypočtenou z názvu konfiguračního souboru virtuálního stroje.

To znamená, že za předpokladu, že útočník zná primární IPv4 adresu operačního systému, pod kterým běží virtuální stroj, se redukuje prohledávaný prostor IID identifikátoru z 64-bitového na 8-bitový.

Na druhé straně, manuálně konfigurované MAC adresy pro *VMWare ESX* server se skládají z OUI ve tvaru 00:50:56 a dolních 3 bajtů v rozsahu 0x000000-0x3fffff (tento rozsah je použit pro odstranění konfliktů s ostatními produkty *VMWare*). Z tohoto důvodu je i pro manuálně konfigurované MAC adresy redukován prohledávaný prostor IID identifikátoru z 64-bitového na 22-bitový [13].

Privacy Extensions

Privacy Extensions je volným rozšířením SLAAC konceptu při vytváření IPv6 adres. Vytvořené IPv6 adresy jsou také známy pod označením "Dočasné adresy". *Privacy Extensions* adresy jsou tvořeny spojením IPv6 prefixu sítě poskytnuté v odpovědi Router Advertisement a náhodně generovaného IID identifikátoru. Kromě jejich nepředvídatelnosti, jsou tyto IPv6 adresy obvykle krátkodobé. V případě, že by se útočník o takového adrese dozvěděl, měl by jen něco málo času na podniknutí dalších kroků k provedení útoku.

Je však nutno brát v úvahu, že privátní adresy jsou brány spíše jako možné rozšíření konceptu SLAAC a ne jako jeho náhrada. To znamená, že použití privátních adres nezabrání útočníkovi v prohledávání adresního prostoru koncové sítě za použití známých vzorů z konceptu SLAAC [13].

Náhodné stabilní identifikátory rozhraní

S cílem zmírnit dopady vyplývající z předvídatelných IPv6 adres odvozených od IID identifikátoru podle IEEE, Microsoft Windows vymyslel schéma pro generování tzv. stabilních IPv6 adres. Toto schéma dokáže vygenerovat IPv6 adresu, která odstraňuje její předvídatelnost bez nutnosti dalšího přegenerování. Výsledný IID identifikátor je konstantní v celé síti.

Za předpokladu, že toto schéma neobsahuje žádné chyby, mohlo by jeho použití odstranit nevýhody používání SLAAC IPv6 adres odvozených od IID identifikátoru podle IEEE. Avšak, protože jsou IID identifikátory konstantní skrze celou IPv6 síť, stále existuje riziko útoku sledování cílové stanice [13].

4.1.2 DHCPv6

DHCPv6 může být v IPv6 síti použít jako mechanizmus pro stavovou konfiguraci IPv6 adres (viz kapitola 2). Samotné přidělování IPv6 adres je prováděno na základě nastavení politiky přidělování DHCPv6 serveru, avšak v mnoha případech jsou adresy přidělovány v sekvenčním pořadí z přiděleného rozsahu. V těchto případech mají IPv6 adresy tendenci být předvídatelnými.

Ve většině případů toto znamená zredukování prohledávaného prostoru IID identifikátorů z 64-bitového na 8-bitový nebo 16-bitový [13].

4.1.3 Teredo

Některé technologie sloužící pro překlad IPv4 adresy na IPv6 adresu mohou také vést k redukci prohledávaného adresního prostoru tím, že specifikují přesný postup, jak mají být výsledné IPv6 adresy generovány. Jedním z takovýchto technologií je například *Teredo*.

Teredo generuje 64-bitový IID identifikátor z IPv4 adresy stroje, který žádá o překlad, z nastavujících příznaků a UDP portu, který je mapován na *Teredo* klienta. S tímto předpisem je útočník schopný redukovat vyhledávací prostor z 64-bitového až na 32-bitový [13].

4.1.4 Manuálně konfigurované IPv6 adresy

V některých případech jsou IPv6 adresy síťových uzlů konfigurovány manuálně. Toto je typický případ například pro konfiguraci směrovačů. Ty většinou nepodporují automatické přidělování adres.

I přes to, že správci sítí mají možnost IID identifikátory volit libovolně z jakékoliv hodnoty v rozmezí 1-264, pro jednoduchost a lepší zapamatování raději volí jeden z následujících vzorů.

Low-byte - adresy, které se používají pro svojí jednoduchost hlavně k adresování směrovačů v sítích IPv6. Adresa obsahuje prefix IPv6 sítě a zbytek adresy jsou nuly. K adresování koncové stanice pak pro jednoduchost slouží pouze poslední bajt výsledné adresy (2001:db8::1).

IPv4-based - tyto adresy vycházejí z IPv4 adresy koncové stanice. Výsledná IPv6 adresa se dostane složením IPv6 prefixu sítě a samotné IPv4 adresy v původním formátu (např. 2001:db8::192.168.1.1).

Wordy - adresy využívají vlastnost IPv6 adresování, kdy se pomocí hexadecimálních písmen složí člověku známé slovo (2001:db8::dead:beef).

První dva zmíněné vzory IPv6 adresování redukují prohledávaný adresní prostor z původních 64 bytů na zhruba 8 bitů (samozřejmě za předpokladu, že pro IPv4-based adresy je znám rozsah IPv4 adres). Na druhé straně pro IPv6 Wordy adresy je vyhledávací prostor větší a více komplexnější, ale stále je velmi redukován s porovnáním původního adresního prostoru o velikosti 64 bitů [13].

4.2 Skenování vzdálené IPv6 sítě

Zatímco útočníci v sítích IPv4 byly schopni najít koncové stanice k napadení pomocí „hrubé síly“ při skenování celého adresního prostoru vybrané sítě, takovéto skenování celé 64-bitové IPv6 sítě by bylo technicky nemožné. Je tedy očekáváno, že útočník použije k redukování prohledávaného adresního prostoru některých se vzorů IPv6 adres popisovaných v kapitole 4.1.

Při skenování vzdálené IPv6 sítě by měl být brán na zřetel ještě jeden faktor, který se při skenování vzdálené IPv4 sítě neprojevoval. Tedy pokud by došlo ke skenování celého 64-bitového adresního prostoru teoreticky by to znamenalo vytvoření 2^64 záznamů v tabulce Neighbor Cache hraničního routeru. Bohužel bylo zjištěno, že většina IPv6 zařízení nezvládá udržovat takovéto množství záznamů v tabulce Neighbor Cache, a proto by takovéto skenování vzdálené IPv6 sítě mohlo vést k vyvolání *DOS* (*Denial of Service*) útoku [13].

4.3 Skenování lokální IPv6 sítě

Skenování lokální IPv6 sítě lze do určité míry považovat za úplně jiný problém než je skenování sítě vzdálené. Hlavním rozdílem je použití lokálních skupinových (multicast) adres, což může útočníkovy velice ulehčit práci v prohledávání velkého adresního prostoru individuálních (unicast) adres (viz kapitola 2).

Útočník může jednoduše vyslat vyhledávací sondu všem uzlům pomocí lokální skupinové adresy ve tvaru ff02::1 definující množinu všech uzlů v lokální síti. Útočník poté obdrží odpovědi od všech aktivních uzlů, které se v síti nachází.

Protože operační systémy typu *Windows* (*Vista*, *7*, atd...) neodpovídají na ICMPv6 Echo Request pakety, které byly poslány na skupinovou IPv6 adresu, mnohé skenovací nástroje většinou vysílají i jiné druhy speciálních paketů, na které již operační systém Windows odpoví. Tím se zajistí vyvolání odpovědi od všech uzlů v síti. Například neexistující IPv6 volba v záhlaví paketu vede k vyvolání ICMPv6 odpovědi o ohlášení chybného parametru.

Mnoho skenovacích nástrojů dokáže prohledávat místo globálních IPv6 adres jenom lokální adresy cílových uzlů. To je způsobeno tím, že je vyhledávací sonda typicky zasílána z útočníkovy lokální adresy. Koncový uzel pak také odpovídá pomocí lokální IPv6 adresy tázaného rozhraní. Toto omezení lze však redukovat pomocí zasílání většího množství vyhledávacích sond obsahující rozdílné prefixy. Tuto techniku používá například nástroj *scan6* z balíku *IPv6 Toolkit* [13].

4.4 Skenování DNS záznamů

Jakýkoliv systém (webový či emailový server atd...) uvedený v DNS záznamech se stává dostupným pro potencionálního útočníka, jelikož adresy těchto systémů jsou veřejně dostupné. Zde je zapotřebí uvědomit si, že umístění jen jediné IPv6 adresy v DNS záznamech může spolu se znalostí používaných vzorů IPv6 adres (viz kapitola 4.1) vést i k odhalení adres ostatních koncových uzlů v této IPv6 síti.

Vhodnou prevencí tohoto typu skenování může být vyloučení používaných vzorů IPv6 adres z DNS záznamů. Útočník by se pak musel spoléhat na

použití slovníkového vyhledávání DNS záznamů pro nalezení všech aktivních systémů v cílové síti, což je obecně méně spolehlivá a více výpočetně náročná metoda, než je mapování uzlů s předvídatelnými IPv6 adresami [13].

4.5 Skenování reverzních DNS záznamů

Metoda která využívá reverzních DNS záznamů k prohledávání IPv6 sítí. Metoda je založena na procházení DNS zóny `ip6.arpa` a hledání záznamů typu `PTR` pro získání informací o koncových uzlech prohledávané sítě. Tato metoda může v některých případech razantně redukovat procházený adresní prostor.

Jak bylo již řečeno, útočník prochází DNS zónu `ip6.arpa` pro cílovou IPv6 síť (např. „0.8.0.0.8.b.d.0.1.0.0.2.ip6.arpa.“ pro „2001:db8:80:/32“) a hledá `PTR` záznamy pro doménová jména

- „0.0.8.0.0.8.b.d.0.1.0.0.2.ip6.arpa.“
- „1.0.8.0.0.8.b.d.0.1.0.0.2.ip6.arpa.“
- „2.0.8.0.0.8.b.d.0.1.0.0.2.ip6.arpa.“
- atd...

Pokud bude nalezen `PTR` záznam, bude útočníkovi navrácena odpověď s číslem RCODE o hodnotě 0 (no errors), v opačném případě bude uvedena hodnota 4 (NXDOMAIN) [13].

4.6 Skenování přenosu DNS zóny

Přenos DNS zóny může často poskytnout informace o potencionálních cílech. Omezení přenosu DNS zóny je tedy důležité v IPv6 sítích, avšak omezení DNS zóny je dobrým zvykem i v sítích IPv4 [13].

4.7 Skenování speciálních protokolů

Množství protokolů také dovoluje překlady lokálních doménových jmen a služeb. Pomocí těchto protokolů lze vhodnou technikou prohledávat zvolenou IPv6 síť. Jsou to například [13]:

- mDNS - multicast DNS
- DNS-SD - DNS Service Discovery
- LLNR - Link-Local Multicast Name Resolution

4.8 Skenování veřejně dostupných archivů

Veřejné archivy například emailových seznamů (mailing-lists) či archivy Usenet zpravodajských zpráv, mohou útočníkovy poskytnout užitečné informace. Jak hostitelská jména, tak IPv6 adresy cílových stanic mohou být snadno získány z hlaviček („`Received from:`“) emailových či zpravodajských zpráv [13].

4.9 Skenování speciálních aplikací a služeb

Peer-to-peer aplikace často využívají centrální server, který koordinuje přenos dat mezi jednotlivými cílovými stanicemi. Pro příklad, *BitTorrent* vytváří sítě uzlů, které si předávají části souborů s informacemi o uzlech, které mají k dispozici potřebná data. Tyto aplikace by pak mohl útočník použít pro získání informací o uzlech a k provedení útoku [13].

4.10 Skenování směrovacích tabulek

Informace o lokálních stanicích mohou být také přístupné z tabulky *Neighbor Cache* nebo z směrovací tabulky jakékoliv stanice připojené do prohledávané IPv6 sítě. Ačkoliv požadavek na přístup do systému v prohledávané

IPv6 síti může omezit použitelnost této techniky, existuje několik scénářů, ve kterých může být tato technika užitečná.

Například některé nástroje či „červy“ dokáží automaticky zjistit veškeré užitečné informace jako jsou *Neighbor Cache* nebo směrovací tabulky všech systémů, do kterých má tento nástroj přístup [13].

4.11 Skenování konfigurace a logů

Konfigurační soubory uzlů v síti obvykle obsahují řadu informací o adresách jiných důležitých uzlech v síti jako jsou například mailové, proxy, DNS, souborové servery a jiné. Příklad takovýchto konfiguračních souborů jsou následující:

- UNIX - soubor /etc/hosts
- UNIX - soubor SSH known_hosts
- Windows - registry

Dalším využitelným zdrojem informací o lokálních uzlech mohou být považovány i systémové logovací soubory včetně logovacích souborů webového serveru [13].

4.12 Skenování směrovacích protokolů

Některé organizace provozují na své IPv6 síti směrovací protokoly pro dynamickou údržbu směrovacích informací napříč celou IPv6 sítí organizace. V takovéto síti by se útočník mohl stát pasivním posluchačem směrovacího protokolu a zjistit tak informace o lokálních stanicích či informace o další platné podsíti této organizace [13].

5 Návrh prostředí pro nasazení

Cílem je vytvoření IPv6 prostředí, které se bude navenek jevit jako reálná IPv6 síť. Do tohoto prostředí bude následně nasazen systém *LaBrea6* a síť honeypotů, kde každý z těchto sub-systémů bude ukládat data o útocích pro následnou analýzu.

Pro zviditelnění sítě pro automatické skenovací nástroje a k propagaci sítě byly stanoveny tyto kroky:

- Návrh adresního prostoru
- Návrh DNS zóny
- Návrh honeypotů pro nasazení
- Návrh výsledného propojení

5.1 Návrh adresního prostoru

Konfigurační soubor systému *LaBrea6* (`ip_list.txt`) popisuje adresní prostor, na kterém systém *LaBrea6* poslouchá. Je zapotřebí tento adresní prostor vhodně navrhnout tak, aby se podobal reálné IPv6 síti.

Při návrhu byl brán v úvahu fakt, že prohledávání celého 64-bitového adresního prostoru cílové IPv6 sítě může být pro útočníka technicky nemožné. Proto se útočník bude spoléhat na skenování cílové IPv6 sítě s použitím známých vzorů IPv6 adres, které jsou v reálných IPv6 sítích běžně používány a tím redukovat prohledávaný adresní prostor (viz kapitola 4.1).

Adresní prostor byl tedy navržen s použitím vzorů IPv6 adres, které jsou v běžné IPv6 síti používány. Tabulky 5.1 a 5.2 popisují jaké techniky konfigurace IPv6 adres a s jakou pravděpodobností jsou v reálných IPv6 sítích používány. Tyto statistiky byly získány z dokumentu [13].

Pro různorodost bylo pro navržení výsledného adresního prostoru zvoleno spojení statistik klientských strojů a statistik směrovačů s doplněním o statistiku používání DHCPv6 konfigurace adres, která v těchto tabulkách není popsaná. Výsledné pravděpodobnostní rozdělení (viz tabulka 5.3) je použito pro

Typ IPv6 adresy	Procento použití
SLAAC	50%
IPv4-based	20%
Teredo	10%
Low-byte	8%
Privacy	6%
Wordy	<1%
Other	<1%

Tabulka 5.1: Statistiky klientských strojů

Typ IPv6 adresy	Procento použití
Low-byte	70%
IPv4-based	5%
SLAAC	1%
Wordy	<1%
Privacy	<1%
Teredo	<1%
Other	<1%

Tabulka 5.2: Statistiky směrovačů

Typ IPv6 adresy	Počet IPv6 adres
SLAAC	78%
IPv4-based	10%
DHCPv6	8%
Teredo	2%
Low-byte	2%

Tabulka 5.3: Navržený adresní prostor systému LaBrea6

konfiguraci generátoru konfigurace systému *LaBrea6* a DNS serveru *Bind9* (viz kapitola 6.1).

5.2 Návrh DNS zóny

Jednou z možných technik pro skenování cílové IPv6 sítě je prohledávání veřejně dostupných DNS záznamů cílové zóny. Z nalezených IPv6 adres pak

může útočník vyčíst používané vzory IPv6 adres, které použije pro redukci adresního prostoru cílové IPv6 sítě. Útočník pak tento redukovaný adresní prostor prohledává s úmyslem najít všechny možné stanice ve zkoumané IPv6 síti.

Samotný návrh DNS zóny vychází z navrženého adresního prostoru z předchozí kapitoly. Pro zvýšení pravděpodobnosti přilákání útočníka je každé IPv6 adrese z navrženého adresního prostoru přiřazeno unikátní doménové jméno. Pokud však uvážíme předpoklad vytvoření adresního prostoru o velikosti 10 000 IPv6 adres, mohl by být požadavek na unikátní doménová jména neproveditelný ba navíc výsledný zónový soubor by mohl útočníka odradit svou nesmyslnou strukturou.

Z těchto důvodů byly při návrhu DNS zóny využity shluky IPV6 adres, které spadají pod jedno doménové jméno doplněné o číslici, která reprezentuje pořadí sub-domény. Tím se zachovává unikátnost doménových jmen pro jednotlivé adresy a zároveň není potřeba nesmyslně veliká množina unikátních jmen pro jednotlivé sub-domény. Pro představu si uvedeme část výsledného zónového souboru:

```
doctor.ciz.zcu.cz       IN      AAAA    <IPv6 adresa 1>
doctor2.ciz.zcu.cz      IN      AAAA    <IPv6 adresa 2>
doctor3.ciz.zcu.cz      IN      AAAA    <IPv6 adresa 3>
moon.ciz.zcu.cz         IN      AAAA    <IPv6 adresa 4>
moon2.ciz.zcu.cz        IN      AAAA    <IPv6 adresa 5>
moon3.ciz.zcu.cz        IN      AAAA    <IPv6 adresa 6>
moon4.ciz.zcu.cz        IN      AAAA    <IPv6 adresa 7>
atd...
```

Výsledná DNS zóna obsahuje kromě `AAAA` záznamů pro adresní prostor systému *LaBrea6* navíc záznamy dalších nasazených systémů a aplikací nasazených v navržené IPv6 síti (viz kapitola 5.3):

- `MX` záznamy pro SMTP honeypot
- `AAAA` záznamy pro webový honeypot

5.3 Návrh honeypotů pro nasazení

Byl vytvořen seznam honeypotů, které budou následně nasazeny do navržené IPv6 sítě včetně systému *LaBrea6*:

- Webový honeypot - s vybranými aplikacemi
- SSH honeypot - *Kippo*
- SMTP honeypot - *Postfix*

5.3.1 Webový honeypot

Jedním z úkolů bylo vytvoření webového honeypotu složeného z navzájem propojených webových aplikací, které poskytnou útočníkovi či skenovacímu „robotovi“ zajímavý obsah pro jeho přilákání. Webový honeypot zároveň bude ukládat veškeré informace o přístupech k jednotlivým webovým stránkám pro následnou analýzu.

Návrh webového honeypotu se skládá z tří navzájem propojených webových aplikací: *Joomla*, *Media Wiki*, *Flyspray*. Každá webová aplikace ukládá informace o přístupech. Jsou ukládány data typu:

- Zdrojová IPv6 adresa
- GET parametry
- POST parametry
- SERVER parametry

Nasazené webové stránky mají následující vlastnosti, které mají podpořit přilákání IPv6 útočníků:

- Webové stránky jsou přístupné jen pod IPv6.
- Každá webová stránka je zaregistrována do indexu vyhledávače *Google*.
- Na jiných fungujících webových stránkách byly vytvořeny zpětné odkazy odkazující na navržené webové stránky.
- Webové stránky obsahují smysluplný obsah.

5.3.2 SSH honeypot - Kippo

Kippo je SSH honeypot střední interakce navržený pro zaznamenávání a zpětnou analýzu SSH útoků. *Kippo* vytváří virtuální SSH přístupový bod s vlastnostmi běžného SSH přístupu. Následuje výpis hlavních vlastností *Kippo*:

- Vytváří falešný souborový systém s možností přidávat a odebírat soubory.
- Poskytuje zaznamenávání a znovu přehrátí kompletní interakce útočníka včetně používaných příkazů v originálním načasování.
- Implementuje příkaz `wget` pro ukládání obsahu pro pozdější analýzu stažených souborů.
- *Kippo* přestírá odhlášení útočníka po použití příkazů `logout` nebo `exit`. Ve skutečnosti je útočník stále připojen ke *Kippo*.

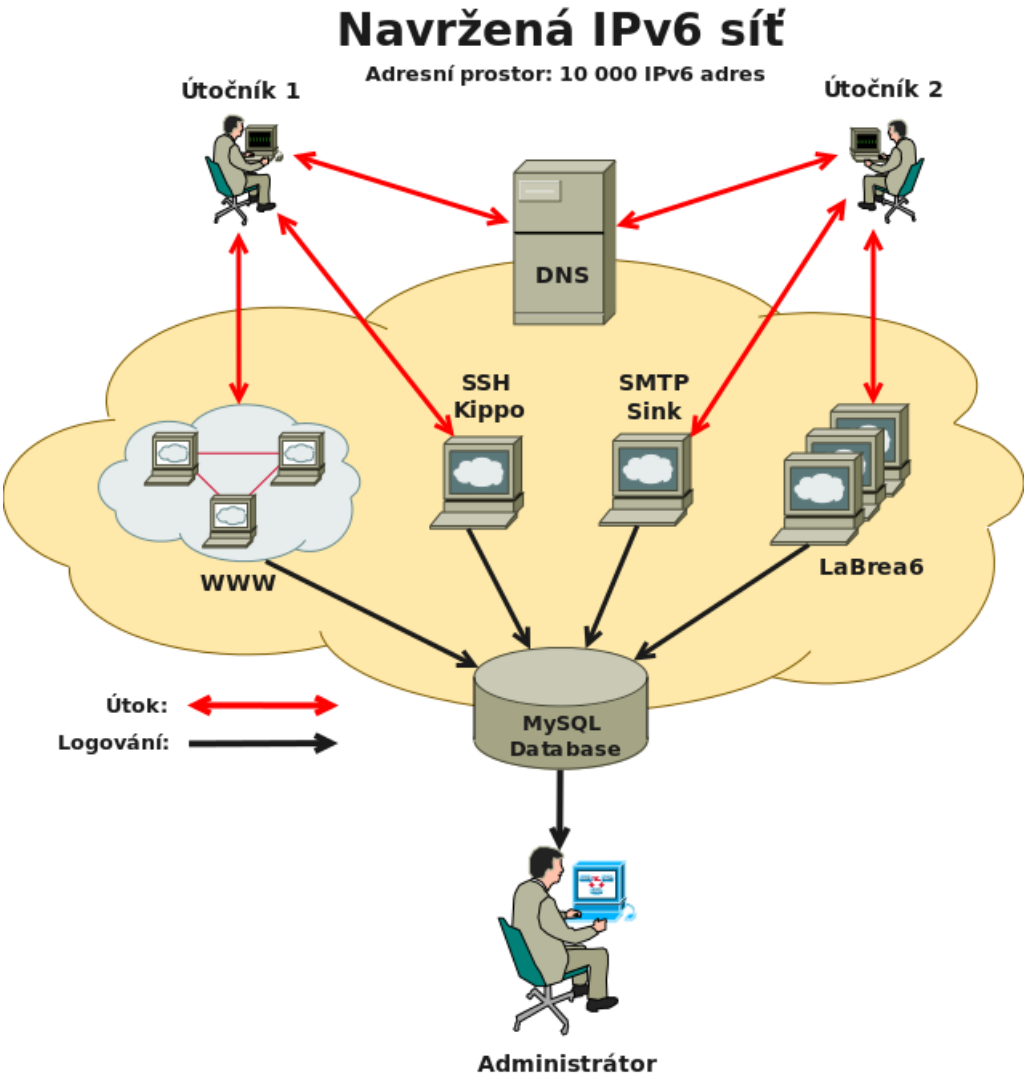
5.3.3 SMTP honeypot - Postfix

Cílem SMTP honeypotu je vytvořit službu, která přijímá emailové zprávy na adresách propagovaných na webovém honepotu (viz kapitola 5.3.1).

K vytvoření SMTP honeypotu je použit mailový server *Postfix*. Pro účel ukládání přijatých SMTP zpráv je využita transportní mapa, pomocí které jsou příchozí SMTP zprávy směrovány do speciální služby, která provede zaznamenání informací o příchozí zprávě do databáze pro následné zpracování (viz kapitola 6.5).

5.4 Návrh výsledného propojení

Návrh propojení všech zvolených systémů znázorňuje obrázek 5.1.



Obrázek 5.1: Navržená IPv6 síť

6 Realizace

Na základě návrhu z kapitoly 5 byla provedena implementace navržených systémů.

6.1 Generátor konfigurace

Generátor konfigurace je program vytvořený v rámci této práce, který podle vstupních parametrů vygeneruje:

- **Konfigurační soubor pro systém LaBrea6** - soubor obsahující seznam IPv6 adres určující adresní prostor navržený v kapitole 5.1, který systém *LaBrea6* pokrývá.
- **Konfigurační soubor pro doménový server Bind9** - jedná se o zónový soubor pro vybranou doménu vytvořený podle navrženého adresního prostoru z kapitoly 5.2.

Generátor konfigurace se skládá z následujících modulů:

- ***main.c*** - hlavní modul generátoru. Slouží k získání vstupních parametrů z příkazové řádky a k jejich kontrole. Pokud jsou parametry správné, je zavolána funkce z modulu *generator.c* pro vygenerování příslušných konfiguračních souborů. V opačném případě je vypsána nápověda k programu.
- ***generator.c*** - modul sloužící, k inicializaci výstupních souborů a jejich správnému vyplnění podle již zkontrolovaných parametrů předaných z příkazové řádky.
- ***genfunc.c*** - modul obsahující potřebné funkce sloužící pro generování příslušných druhů IPv6 adres. Každá funkce vrací jednu nově vygenerovanou IPv6 adresu.
- ***list.c*** - pomocný modul, který reprezentuje spojový seznam. Je použit pro načtení seznamu podstatných jmen reprezentujících jednotlivá doménová jména ve výsledném konfiguračním zónovém souboru pro DNS server *Bind9.*

Použití a vstupní parametry generátoru konfigurace popisuje následující výpis.

```
============================================================
pouziti: labrea6-config  -n -r [-h] [-o] [-b] [-S] [-I] [-T] [-W] [-L] [-s]
[-i] [-t] [-w] [-l]
-h = vytiskne napovedu
--------------------------
-r = *segment IPv6 adres, pro ktery se maji vytvorit konfiguracni soubory.
-n = *vstupni textovy soubor obsahujici podstatna jmena pro tvoreni clusteru
-o = vystupni konfiguracni soubor pro system LaBrea6 typu .txt
-b = vystupni konfiguracni soubor pro DNS server Bind9
--------------------------
-S = pocet IPv6 adres typu SLAAC
-I = pocet IPv6 adres typu IPv4-BASED
-T = pocet IPv6 adres typu TEREDO
-W = pocet IPv6 adres typu WORDY
-L = pocet IPv6 adres typu LOW-BYTE
-D = pocet IPv6 adres typu DHCPv6
-s = velikost clusteru pro DNS server Bind9 typu SLAAC
-i = velikost clusteru pro DNS server Bind9 typu IPv4-BASED
-t = velikost clusteru pro DNS server Bind9 typu TEREDO
-w = velikost clusteru pro DNS server Bind9 typu WORDY
-l = velikost clusteru pro DNS server Bind9 typu LOW-BYTE
-d = velikost clusteru pro DNS server Bind9 typu DHCPv6
--------------------------
* = povinny parametr
============================================================
```

Generátor konfigurace je využíván externím skriptem `labrea6.reconf`, který se stará o správné překonfigurování systému *LaBrea6* a vygeneruje příslušné konfigurační soubory podle navrženého adresního prostoru (viz kapitola 5.1) po zavolání příkazu `/etc/init.d/labrea6 reconfigure`.

6.2 LaBrea6 - IDS system

Pro nasazení systému *LaBrea6* do navržené IPv6 sítě z kapitoly 5 bylo potřeba rozšířit původní systém *LaBrea6* a vytvořit konfiguraci adresního prostoru pomocí generátoru konfigurace (viz kapitola 6.1) podle navrženého prostoru v kapitole 5.1.

6.2.1 Rozšíření původního systému LaBrea6

Úprava původního systému *LaBrea6* spočívala v implementaci ukládání komunikace ve formátu `pcap` a doplnění o konfiguraci nástroje *logrotate*.

Ukládání v pcap formátu

Byl implementován mechanismus, který ukládá veškerou komunikaci systému *LaBrea6* do souboru v `pcap` formátu. Výsledný `pcap` soubor s uloženou komunikací systému *LaBrea6* je možné otevřít v programu *Wireshark*. Toto vylepšení výrazně zlepší možnosti analýzy případných útoků v analyzované síti IPv6 (viz kapitola 7.2).

Knihovna *libpcap* již obsahuje funkce pro ukládání komunikace ve formátu `pcap`, které lze k tomuto účelu použít. V prvé řadě je nutné otevřít výstupní soubor pro ukládání. K tomu je použita knihovní funkce `pcap_dump_open()`, která navrací ukazatel na strukturu `pcap_dumper_t`. Tento ukazatel je pak použit pro zapisování jednotlivých paketů (přijatých i odeslaných) pomocí funkce `pcap_dump()`.

Doplnění o konfiguraci *logrotate*

Program *logrotate* se v Unixových systémech používá pro správu logů od jednotlivých programů. Soubory obsahující logy jednotlivých programů mají časem tendenci růst a je potřeba tyto soubory takzvaně "rotovat" (tj. smazat staré a vytvořit nové).

Systém *LaBrea6* byl doplněn o konfigurační soubor programu *logrotate*, který se stará o denní rotaci příslušných log souborů včetně komprese. Soubory starší jak 1 rok jsou mazány. V průběhu samotné rotace je systém *LaBrea6* pozastaven, aby se předešlo k chybám při rotaci.

6.2.2 Konfigurace adresního prostoru

Adresní prostor je konfigurován pomocí generátoru konfigurace s nastavením, které vychází z návrhu adresního prostoru v kapitole 5.1. K vytvoření konfiguračních souborů pro systém *LaBrea6* a DNS server *Bind9* byl vytvořen skript `labrea6.reconf`, kdy se provede pozastavení systému *LaBrea6*, vygenerování konfiguračních souborů pro oba zmíněné systémy a znovu obnovení funkce systému *LaBrea6*.

Pro účely vytvoření IPv6 adresního prostoru pro systém *LaBrea6* byl stroji přidělen segment IPv6 adres definovaný prefixem sítě:

`2001:718:1801:1077::/64`

Tento prefix IPv6 sítě je předáván jako vstupní parametr generátoru konfigurace a tím definuje pro jakou IPv6 síť jsou výsledné IPv6 adresy generovány.

Spolu s prefixem sítě je potřeba jako vstupní parametr generátoru definovat i počty jednotlivých druhů generovaných IPv6 adres a velikosti příslušných DNS shluků (viz kapitola 5). Zvolení těchto parametrů zobrazuje tabulka 6.1.

Typ IPv6 adresy	Počet IPv6 adres	Počet DNS shluků
SLAAC	7 800 (78%)	100
IPv4-based	1 000 (10%)	400
DHCPv6	800 (8%)	46
Teredo	200 (2%)	25
Low-byte	200 (2%)	20
Celkem:	**10 000 (100%)**	-

Tabulka 6.1: Vstupní parametry generátoru konfigurace

Posledním vyžadovaným parametrem je cesta k souboru, který obsahuje seznam unikátních anglických podstatných jmen, které slouží generátoru pro vytváření smysluplných doménových názvů pro výslednou DNS zónu domény `ciz.zcu.cz`. Soubor s těmito unikátními podstatnými jmény je k nalezení v projektu generátoru konfigurace pod umístěním `/opt/labrea6/src/labrea6-config/config/nouns.txt`.

Následuje příklad spuštění generátoru s navrženými parametry a ukázka výstupních konfiguračních souborů:

```
./labrea6-config -r "2001:718:1801:1077::" -S 7800 -s 100 -I 1000 -i40
 -T 200 -t 25 -L 200 -l 20 -D800 -d 46 -n ./src/labrea6-config/nouns.txt
```

Konfigurační soubor systému *LaBrea6* `ip_list.txt`:

```
2001:0718:1801:1077:baac:6fff:fec1:56de
2001:0718:1801:1077:baac:6fff:fe68:5503
2001:0718:1801:1077:baac:6fff:fe3f:59e9
2001:0718:1801:1077:0000:0000:192.168.206.239
2001:0718:1801:1077:0000:0000:192.168.172.119
2001:0718:1801:1077:0000:0000:192.168.174.239
2001:0000:53aa:064c:8000:63bf:3f57:5bec
2001:0000:53aa:064c:8000:63bf:3f57:3733
2001:0000:53aa:064c:8000:63bf:3f57:c683
2001:0718:1801:1077:0000:0000:0000:0031
2001:0718:1801:1077:0000:0000:0000:0071
```

Konfigurační soubor DNS zóny `db.ciz.zcu.cz` pro server *Bind9*:

```
;
; BIND data file for ciz.zcu.cz
;
$TTL    604800
$ORIGIN ciz.zcu.cz.

@ 604800        IN      SOA     ns.ciz.zcu.cz. webmaster.ciz.zcu.cz. (
                     2013042801         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@                               IN      NS      cizam.civ.zcu.cz.
@                               IN      MX      10      mail.ciz.zcu.cz.
@                               IN      MX      20      mail2.ciz.zcu.cz.
mail                            IN      AAAA     2001:0718:1801:1077::1:4
mail2                           IN      AAAA     2001:0718:1801:1077::1:5
wiki.gabless                    IN      AAAA     2001:0718:1801:1077::1:12
fly.gabless                     IN      AAAA     2001:0718:1801:1077::1:11
gabless                         IN      AAAA     2001:0718:1801:1077::1:10
www.wiki.gabless                IN      CNAME     wiki.gabless
www.fly.gabless                 IN      CNAME     fly.gabless
www.gabless                     IN      CNAME     gabless
furniture       IN      AAAA    2001:0718:1801:1077:baac:6fff:fec1:56de
furniture1      IN      AAAA    2001:0718:1801:1077:baac:6fff:fe68:5503
furniture2      IN      AAAA    2001:0718:1801:1077:baac:6fff:fee6:5f8d
furniture3      IN      AAAA    2001:0718:1801:1077:baac:6fff:fe5b:9111
dolls           IN      AAAA    2001:0718:1801:1077:0000:0000:0000:005d
dolls1          IN      AAAA    2001:0718:1801:1077:0000:0000:0000:005e
apple           IN      AAAA    2001:0718:1801:1077:0000:0000:192.168.80.106
apple1          IN      AAAA    2001:0718:1801:1077:0000:0000:192.168.45.58
apple2          IN      AAAA    2001:0718:1801:1077:0000:0000:192.168.18.160
```

6.3 Bind9 - DNS server

Na produkčním stroji byl nainstalován a nakonfigurován DNS server *Bind9*, který slouží pro spravování DNS zóny `ciz.zcu.cz`.

Prvním krokem bylo zavedení konfigurace zóny `db.ciz.zcu.cz` vygenerované generátorem konfigurace (viz kapitola 6.1) do správy DNS serveru v souboru `named.conf.local`:

```
zone "ciz.zcu.cz"
{
    type master;
        file "/etc/bind/db.ciz.zcu.cz";
};
```

Dále bylo potřeba nastavit samotný DNS server*Bind9* pro poslech na

všech rozhraních upravením konfiguračního souboru `named.conf.options`:

```
options {
        directory "/var/cache/bind";

        auth-nxdomain no;

        # Poslech na vsech IP a rozhranich
        listen-on-v6 { any; };
        listen-on { any; };
};
```

6.4 Kippo - SSH honeypot

Pro nasazení *Kippo* byly vytvořeny tři skripty: `kippo.start`, `kippo.stop`, `kippo.loadup`. Tyto tři skripty jsou volány z init skriptu `/etc/init.d/kippo`.

Spuštění systému *Kippo* pomocí příkazu `start` vyvolá provedení vytvořeného skriptu `kippo.start`, který provede následující kroky:

- Pozastavení systému *LaBrea6*.
- Skript vybere 100 náhodně vybraných IPv6 adres ze souboru ip_list.txt, který reprezentuje adresní prostor navržené IPv6 sítě systému *LaBrea6*.
- Vybrané adresy jsou následně přidány na síťové rozhraní produkčního stroje.
- Spuštění systému *LaBrea6* s upravenou konfigurací.

Těmito kroky se zajistí, že systém *LaBrea6* kompletně přenechá systému *Kippo* veškeré odpovědi vztažené k vybraným IPv6 adresám. Toto opatření je nutné, aby se zabránilo chybným odpovědím při vytváření SSH spojení se systémem *Kippo*. Pro ukončení systému *Kippo* je zavolán skript `kippo.stop`, který provede návrat původního nastavením zpětným procházením výše zmíněných kroků. V době běhu jsou ukládány informace o přístupech ke *Kippo* do databáze pomocí skriptu `kippo.loadup`.

Útočník obecně očekává SSH přístupové body pod standardním TCP portem 22, proto byl systémový SSH deamon přesunut na vybraný neprivilegovaný port a tím uvolněn port 22 pro *Kippo*. Problém však nastal v momentě, kdy bylo potřeba na tento port připojit systém *Kippo*. Obecně není

vhodné pouštět honeypoty s právy administrátora a proto *Kippo* obsahuje bezpečnostní pojistku, kdy nelze spouštět tento systém v privilegovaném režimu. To však představuje překážku, kdy bez administrátorských práv není možné přistupovat k privilegovaným portům nižším než je port 1024. Protože je systém *Kippo* napsán v jazyce *Python*, jednou z možností je nastavení administrátorských práv interpretu jazyka *Python* pro poslech na portech nižších než 1024 nastavením tzv. *capabilities* typu „`CAP_NET_BIND_SERVICE`“ pro interpret jazyka *Python* pomocí příkazu:

```
setcap 'cap_net_bind_service=+ep' /usr/local/bin/python
```

6.5 Postfix - SMTP honeypot

SMTP honeypot je realizován pomocí vhodného nastavení již existujícího SMTP serveru Postfix. Prvním krokem bylo nastavení transportní mapy, která slouží pro předávání pošty na jakéhokoliv jiného hostitele bez ohledu na nastavení záznamu `MX` v DNS. Jako hostitel může být použita i systémová pojmenovaná „roura“, která předá zvolené informace o příchozí SMTP zprávě například skriptu pro následné zpracování. Skript přijme tyto informace a uloží do databáze.

Definování pojmenované „roury“ (v tomto případě `pspam_route`) a informací, které se mají předávat vytvořenému skriptu `sink.loadup.php` bylo nutno zaregistrovat v konfiguračním souboru `master.cfg` (viz příloha A.3).

Postfix server je používán také pro zasílání denního reportingu o útocích a je tedy potřeba vymezit, které SMTP zprávy má zpracovat skript, a které se mají standardně doručit příjemci. Toto vymezení je nastaveno konfiguračním souborem transportní mapy, kde jsou včetně přesměrování zpráv do skriptu uvedeny i emailové adresy uživatelů, kteří odebírají denní emaily:

```
cizam@students.zcu.cz :
* pspam_route         :
```

Toto nastavení tedy zařídí, aby veškerá pošta směrovaná na adresáta *cizam@students.zcu.cz* byla doručena normálním způsobem (adresát pro denní reporting) a všechna ostatní pošta směrovaná na tento SMTP server byla předána pojmenované „rouře“ s názvem `pspam_route` pro další zpracování v definovaném skriptu.

Dalším nastavením vytvořeného *SMTP Sink* serveru je definování, z jakých IPv6 adres bude server přijímat SMTP zprávy. Toto nastavení se provede v konfiguračním souboru `main.cfg` nastavení parametru `myneworks` na hodnotu:

```
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::]/0
```

6.6 Webový honeypot

Na produkčním stroji byl nakonfigurován webový server *Apache* pro webový honeypot složený z vybraných webových aplikací z kapitoly 5.3.1.

Na stránky webových aplikací byly umístěny jak odkazy na jednotlivé dílčí stránky webového honeypotu, tak smyšlené emailové adresy v doméně `ciz.zcu.cz` odkazující na SMTP honeypot (viz kapitola 6.5), aby byla zajištěna provázanost nasazených honeypotů.

Byly také vytvořeny transparentní zpětné odkazy na jednotlivé aplikace z jiných již existujících stránek pro přilákání prohledávacích pavouků:

```
<a href="http://gabless.ciz.zcu.cz/">      <font face="verdena"> </font> </a>
<a href="http://fly.gabless.ciz.zcu.cz/">  <font face="verdena"> </font> </a>
<a href="http://wiki.gabless.ciz.zcu.cz/"> <font face="verdena"> </font> </a>
```

Webový honeypot ukládá do databáze veškerý přístup na webové stránky stránky pomocí skriptu `request_log.php`. Tomuto skriptu jsou předávány veškeré informace o jednotlivých přístupech pomocí funkce webového serveru *Apache* `auto_prepend_file`, která volá skript před každým načtením php souborů webových aplikací. Nastavení je provedeno upravením následující řádky v konfiguračním souboru webového serveru `php.ini`:

```
auto_prepend_file = /var/www/request_log.php
```

6.7 Webové administrační prostředí

Informace ze všech nasazených systémů jsou ukládány do lokální databáze. Aby měl administrátor analyzované IPv6 sítě možnost provádění analýzy nad uloženými daty, jsou data prezentována v smysluplné formě ve statistikách na webovém administračním prostředí.

6.7.1 Oprava chyb

Původní administrační webové rozhraní obsahovalo několik chyb v zobrazování statistik o útocích:

- Graf zobrazující statistiku útoků v čase podával špatné informace.
- Chybné zobrazování portů.

Graf byl zpočátku lineární, tedy prokládal jednotlivé počty útoků přímkou. To způsobovalo nepřesnosti v zobrazení statistik o útocích. Oprava grafu spočívala v nahrazení lineárního grafu grafem sloupcovým, který zobrazuje příslušný počet útoků pomocí jednotlivých sloupců.

Oprava chyby v zobrazování portů spočívala v použití překladu tohoto vnitřního formátu pomocí funkce `ntohs()` na řetězec znaků, který správně reprezentuje uložený port.

6.7.2 Struktura

Hlavní navigace webového rozhraní byla rozšířena o záložky:

- Summary
- Kippo - SSH honeypot
- WWW
- SMTP Sink
- Graphs
- DNS queries
- PhpMyAdmin

Summary

Úvodní stránka webového rozhraní „*Summary*“ obsahuje souhrnné informace o útocích nasbírané ze všech nasazených systémů a aplikací, které byly do navržené IPv6 sítě zapojeny. Jedná se o statistiky typu:

- Útočníci z lokální sítě za posledních 30 dní
- Statistiky používaných portů
- Top 10 útočníci za poslední 3 dny
- Top 10 prohledávači za poslední 3 dny
- SSH útočníci za posledních 24 hodin
- Návštěvníci webových stránek za posledních 24 hodin
- SMTP zprávy za posledních 24 hodin

Statistiky z této úvodní stránky jsou dále použity pro zasílání denních statistik (viz kapitola 6.8).

Kippo - SSH honeypot

V této záložce jsou k nalezení informace o IPv6 adresách, které prováděly interakci s nasazeným SSH honeypotem *Kippo*. Zobrazovány jsou jen informace základní jako jsou čas posledního útoku, počet připojení dané adresy či odhad provedených příkazů. Pro získání všech informací z SSH honeypotu *Kippo* o dané IPv6 adrese je možné kliknout na odkaz příslušné IPv6 adresy (detailnější informace jsou dostupné v záložce 6.7.3).

WWW

Záložka „*WWW*“ poskytuje základní informace o návštěvnících webových aplikací nasazené do navržené IPv6 sítě. Z těchto statistik lze zjisti například počet provedených přístupů na webové stránky nebo čas posledního přístupu. Pro zobrazení detailnějších informací (detailnější informace jsou dostupné v záložce 6.7.3).

LaBrea6

Summary | Kippo - SSH Honeypot | WWW | SMTP Sink | Graphs | DNS queries | PhpMyAdmin

Overall

Time frame: 2013-02-19 11:06:47 - 2013-04-28 09:47:34
Last tarpit at: 2013-04-28 09:47:34 (**11994m ago**) from **2002:36f5:2be2:0000:0fe2:ef12:d21d:b001** (
2002:36f5:2be2:0000:0fe2:ef12:d21d:b001)
Legend: Port rise Port fall Well-known port Registered port Dynamic/private/local port

Ownnet attackers for 30 days

time	held	ips	ipd	psrc	pdst	count	hostname
2013-04-25 15:34:48	1	2001:0718:1801:10f4:0216:3eff:fe23:8202	2001:0718:1801:1077:baac:6fff:fec2:9334	59547	22	6	2001:0718:1801:10f4:0216:3eff:fe23:8202
2013-04-16 21:21:23	0	2001:0718:1801:10f4:0216:3eff:fe23:8202	2001:0718:1801:1077:baac:6fff:fe8d:fc91	52390	2444	2	2001:0718:1801:10f4:0216:3eff:fe23:8202
2013-04-16 11:23:04	2	2001:0718:1801:1001:0000:0000:0001:b0d0	2001:0718:1801:1077:baac:6fff:feb1:aeb6	44576	456	11	2001:0718:1801:1001:0000:0000:0001:b0d0
2013-04-08 21:19:46	0	2001:0718:1801:10f4:0216:3eff:fe23:8202	2001:0718:1801:1077:baac:6fff:fe41:ea66	58140	2222	2	2001:0718:1801:10f4:0216:3eff:fe23:8202

Port trends

trend = count(dpst) - avg(last 72 hours)

pdst/name	count	count72	trendRatio
22/ssh	12	0	12
456/	11	0	11
80/www	8	0	8
443/https	8	0	8
6106/	6	0	6
9220/	6	0	6
5030/	6	0	6
2223/	0	4	0.25
23/telnet	0	4	0.25
123/ntp	0	3	0.333
2555/	0	1	1
12345/	0	1	1
2222/	0	1	1

Top 10 attackers for last 72 hours

ips	count	first	last	hostname
2002:36f5:2be2:0000:0fe2:ef12:d21d:b001	4128	2013-04-28 09:46:41	2013-04-28 09:47:34	2002:36f5:2be2:0000:0fe2:ef12:d21d:b001
2001:0718:1801:1001:0000:0000:0001:b0d0	11	2013-04-16 11:23:04	2013-04-16 11:24:55	2001:0718:1801:1001:0000:0000:0001:b0d0
2001:0718:1801:10f4:0216:3eff:fe23:8202	10	2013-04-08 21:19:46	2013-04-25 15:35:40	2001:0718:1801:10f4:0216:3eff:fe23:8202
2a01:0430:0046:0000:0000:0000:0000:0002	2	2013-04-16 06:30:48	2013-04-16 06:30:48	2a01:0430:0046:0000:0000:0000:0000:0002

Top 10 ICMPv6 attackers for last 72 hours

ips	count	first	last	hostname
2a01:0430:0046:0000:0000:0000:0000:0002	110	2013-05-02 08:46:14	2013-05-02 08:48:13	2a01:0430:0046:0000:0000:0000:0000:0002
2001:0718:1801:10f4:0216:3eff:fe23:8202	57	2013-04-08 21:23:38	2013-04-28 20:51:02	2001:0718:1801:10f4:0216:3eff:fe23:8202
fe80:0000:0000:0000:0208:e3ff:feff:fc2c	31	2013-04-08 21:19:46	2013-05-02 08:47:40	fe80:0000:0000:0000:0208:e3ff:feff:fc2c

SSH attackers for last 24 hours

time	ips	session_count	hostname
2013-04-16 11:25:54	2a01:0028:00ca:0110:0088:0086:0102:0005	1	2a01:0028:00ca:0110:0088:0086:0102:0005
2013-04-16 11:25:11	2001:0718:1801:1001:0000:0000:0001:b0d0	2	2001:0718:1801:1001:0000:0000:0001:b0d0
2013-04-16 11:11:36	0000:0000:0000:0000:0000:ffff:93e4:0185	5	bodik.zcu.cz
2013-04-15 16:23:00	0000:0000:0000:0000:0000:ffff:7f00:0001	1	localhost.localdomain
2013-04-15 16:08:53	0000:0000:0000:0000:0000:ffff:93e4:d1ae	4	kolej-bk-70.zcu.cz
2013-04-08 21:27:36	2001:0718:1801:10f4:0216:3eff:fe23:8202	4	2001:0718:1801:10f4:0216:3eff:fe23:8202

WWW visitors for last 24 hours

time	ips	hostname
2013-04-28 09:14:40	2002:36f5:2be2:0000:dead:beef:dead:beef	2002:36f5:2be2:0000:dead:beef:dead:beef
2013-04-16 21:09:05	2001:1528:0123:0044:046b:c1ff:fefd:d191	2001:1528:0123:0044:046b:c1ff:fefd:d191
2013-04-16 16:24:10	2001:1528:0123:0044:689a:6bff:fe94:1017	2001:1528:0123:0044:689a:6bff:fe94:1017
2013-04-16 15:46:55	2001:1528:0123:0044:7059:0fff:feb3:045c	2001:1528:0123:0044:7059:0fff:feb3:045c
2013-04-13 20:25:50	2002:253b:d5d1:0000:0000:0000:253b:d5d1	2002:253b:d5d1:0000:0000:0000:253b:d5d1
2013-04-09 08:41:17	2001:1528:0123:0044:6c0a:62ff:fe9c:7974	2001:1528:0123:0044:6c0a:62ff:fe9c:7974

Obrázek 6.1: Ukázka záložky *Summary*

LaBrea6

Summary | Kippo - SSH Honeypot | WWW | SMTP Sink | Graphs | DNS queries | PhpMyAdmin

Last 1000 SSH attackers

last_time	ips	count	session_count	hostname
2013-04-30 10:11:06	0000:0000:0000:0000:0000:ffff:93e4:0185	84	5	bodik.zcu.cz
2013-04-29 09:43:20	0000:0000:0000:0000:0000:ffff:93e4:d1ae	81	4	kolej-bk-70.zcu.cz
2013-04-28 14:58:47	2001:0718:1801:10f4:0216:3eff:fe23:8202	252	4	2001:0718:1801:10f4:0216:3eff:fe23:8202
2013-04-16 11:26:00	2a01:0028:00ca:0110:0088:0086:0102:0005	23	1	2a01:0028:00ca:0110:0088:0086:0102:0005
2013-04-16 11:25:39	2001:0718:1801:1001:0000:0000:0001:b0d0	31	2	2001:0718:1801:1001:0000:0000:0001:b0d0
2013-04-15 16:23:00	0000:0000:0000:0000:0000:ffff:7f00:0001	5	1	localhost.localdomain
2013-04-02 23:19:06	2a01:0430:0046:0000:0000:0000:0000:0002	5	1	2a01:0430:0046:0000:0000:0000:0000:0002

Obrázek 6.2: Ukázka záložky *Kippo - SSH honeypot*

LaBrea6

Summary | Kippo - SSH Honeypot | WWW | SMTP Sink | Graphs | DNS queries | PhpMyAdmin

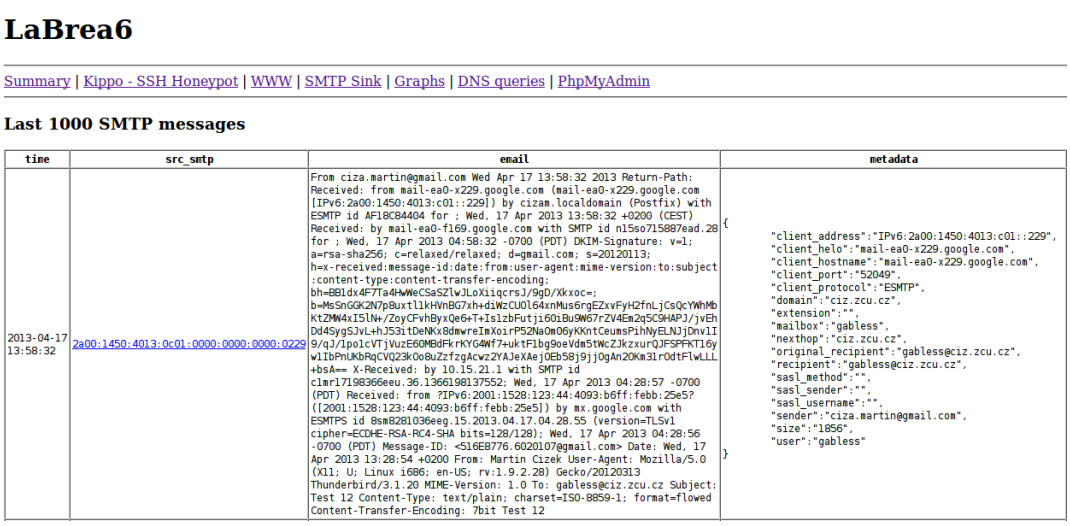
Last 1000 WWW visitors

time	ips	count	hostname
2013-05-02 08:33:40	2001:4860:4801:2003:0000:6006:1300:b075	1342	crawl-2001-4860-4801-2003-0000-6006-1300-b075.googlebot.com
2013-05-02 08:14:28	2001:4860:4801:2004:0000:6006:1300:b075	1410	crawl-2001-4860-4801-2004-0000-6006-1300-b075.googlebot.com
2013-05-02 07:36:25	2001:4860:4801:2001:0000:6006:1300:b075	1408	crawl-2001-4860-4801-2001-0000-6006-1300-b075.googlebot.com
2013-05-02 06:39:03	2001:4860:4801:2002:0000:6006:1300:b075	1403	crawl-2001-4860-4801-2002-0000-6006-1300-b075.googlebot.com
2013-04-28 09:18:09	2002:36f5:2be2:0000:dead:beef:dead:beef	43	2002:36f5:2be2:0000:dead:beef:dead:beef
2013-04-16 21:09:09	2001:1528:0123:0044:046b:c1ff:fefd:d191	3	2001:1528:0123:0044:046b:c1ff:fefd:d191
2013-04-16 16:24:10	2001:1528:0123:0044:689a:6bff:fe94:1017	1	2001:1528:0123:0044:689a:6bff:fe94:1017
2013-04-16 15:46:57	2001:1528:0123:0044:7059:0fff:feb3:045c	2	2001:1528:0123:0044:7059:0fff:feb3:045c
2013-04-13 20:25:54	2002:253b:d5d1:0000:0000:0000:253b:d5d1	2	2002:253b:d5d1:0000:0000:0000:253b:d5d1
2013-04-09 08:43:13	2001:1528:0123:0044:6c0a:62ff:fe9c:7974	5	2001:1528:0123:0044:6c0a:62ff:fe9c:7974
2013-04-05 16:21:06	2001:0718:1801:1001:0000:0000:0001:b0d0	9	2001:0718:1801:1001:0000:0000:0001:b0d0
2013-04-03 15:10:13	2001:1528:0123:0044:e84e:c4ff:fe7a:95d9	3	2001:1528:0123:0044:e84e:c4ff:fe7a:95d9
2013-04-03 15:05:58	2001:1528:0123:0044:1821:c1ff:fec3:d808	33	2001:1528:0123:0044:1821:c1ff:fec3:d808

Obrázek 6.3: Ukázka záložky *WWW*

SMTP Sink

Pro zobrazení souhrnných informací o příchozích SMTP zprávách z nasazeného SMTP honeypotu slouží záložka „*SMTP Sink*“. Je zde uveden seznam posledních příchozích SMTP zpráv včetně jejích hlaviček a příslušných metadat.



LaBrea6

Summary | Kippo - SSH Honeypot | WWW | SMTP Sink | Graphs | DNS queries | PhpMyAdmin

Last 1000 SMTP messages

time	src_smtp	email	metadata
2013-04-17 13:58:32	2a00:1450:4013:0c01:0000:0000:0000:0229	From ciza.martin@gmail.com Wed Apr 17 13:58:32 2013 Return-Path: Received: from mail-ea0-x229.google.com (mail-ea0-x229.google.com [IPv6:2a00:1450:4013:c01::229]) by cizam.localdomain (Postfix) with ESMTP id AF18C84404 for ; Wed, 17 Apr 2013 13:58:32 +0200 (CEST) Received: by mail-ea0-f169.google.com with SMTP id n15so715887ead.28 for ; Wed, 17 Apr 2013 04:58:32 -0700 (PDT) DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=gmail.com; s=20120113; h=x-received:message-id:date:from:user-agent:mime-version:to:subject :content-type:content-transfer-encoding: bh=BBldx4F7Ta4HWWeCSaSZlwJLoXiiqcrsJ/9gD/Xkxoc=; b=MsSnGGK2N7p8uxtlUkHvnBG7xh+diWzCUOl64xnMus8rgEZxvFyH2fnLjCsQcYWhMb KtZMW4xI5lN+/ZoyCFvhByxQe6+T+Is1zbFutji6OiBu8W67rZV4Em2q5C9HAPJ/jvEh Dd4SygSJvL+hJ53itDeNKx8dmwreImXoirP52NaOm06yKKntCeumsPihNyELNJjDnv1I 9/qJ/lpolcVTjVuzE6OMBdFkrkYG4Wf7+uktF1bg9oeVdm5tWcZJkzxurQJFSPFKTl6y w1IbPnUKbRqCVQ23kOo8uZzfzgAcwz2YAJeXAejOEb58j9jjOgAn2OKm3lrOdtFlwLLL +bsA== X-Received: by 10.15.21.1 with SMTP id clmrl7198366eeu.36.1366198137552; Wed, 17 Apr 2013 04:28:57 -0700 (PDT) Received: from ?IPv6:2001:1528:123:44:4093:b6ff:febb:25e5? ([2001:1528:123:44:4093:b6ff:febb:25e5]) by mx.google.com with ESMTPS id 8sm8281036eeg.15.2013.04.17.04.28.55 (version=TLSv1 cipher=ECDHE-RSA-RC4-SHA bits=128/128); Wed, 17 Apr 2013 04:28:56 -0700 (PDT) Message-ID: <516E8776.6020107@gmail.com> Date: Wed, 17 Apr 2013 13:28:54 +0200 From: Martin Cizek User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.28) Gecko/20120313 Thunderbird/3.1.20 MIME-Version: 1.0 To: gabless@ciz.zcu.cz Subject: Test 12 Content-Type: text/plain; charset=ISO-8859-1; format=flowed Content-Transfer-Encoding: 7bit Test 12	{ "client_address":"IPv6:2a00:1450:4013:c01::229", "client_helo":"mail-ea0-x229.google.com", "client_hostname":"mail-ea0-x229.google.com", "client_port":"52049", "client_protocol":"ESMTP", "domain":"ciz.zcu.cz", "extension":"", "mailbox":"gabless", "nexthop":"ciz.zcu.cz", "original_recipient":"gabless@ciz.zcu.cz", "recipient":"gabless@ciz.zcu.cz", "sasl_method":"", "sasl_sender":"", "sasl_username":"", "sender":"ciza.martin@gmail.com", "size":"1856", "user":"gabless" }

Obrázek 6.4: Ukázka záložky *SMTP Sink*

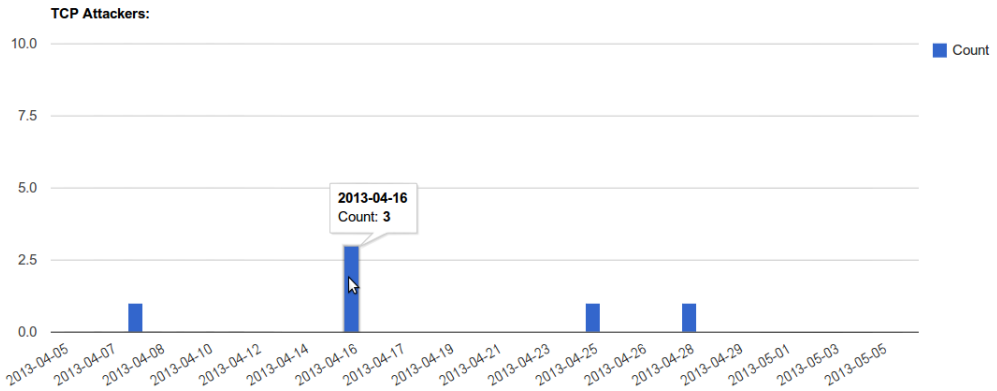
Graphs

Záložka „*Graphs*“ slouží pro grafické znázornění proběhlých útoků na systém *LaBrea6* za posledních 31 dní v podobě grafů. Grafy slouží pro přehlednější představu o útocích, jak probíhaly v čase. Uvedeny jsou zde dva typy grafů:

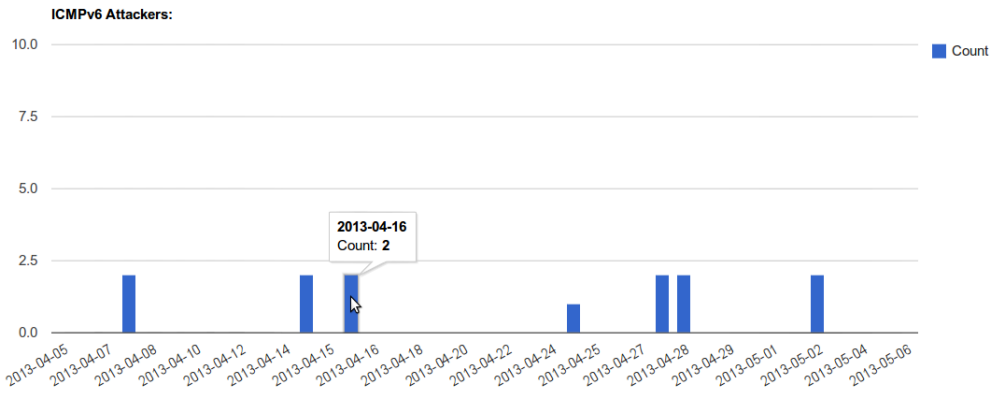
- Útočníci za posledních 31 dní
- ICMPv6 útočníci za posledních 31 dní

DNS queries

Záložka „*DNS queries*“ se používá pro zobrazení posledních 200 dotazů na nakonfigurovaný DNS server Bind9. Informace o DNS příchozích DNS dotazech by mohli pomoci odkrýt skenovaní DNS zóny a tím identifikovat útočníka, který se snaží z DNS dotazů získat vzory používaných IPv6 adres z cílové sítě pro následné skenování cílové IPv6 sítě (viz kapitola 4.4).



Obrázek 6.5: Ukázka záložky *Grahps* - 1

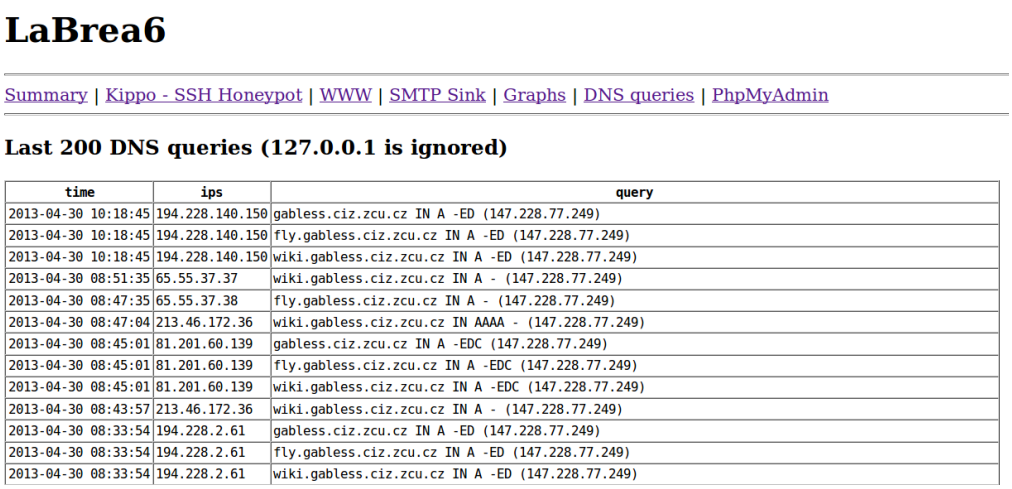


Obrázek 6.6: Ukázka záložky *Grahps* - 2

6.7.3 Souhrnné informace o IP

Jako dalším rozšířením webového administračního rozhraní je zobrazení podrobných informací o IPv6 adrese za všech systémů a aplikací nasazených do navržené IPv6 sítě. Pokud si administrátor přeje zobrazit tyto detailní informace o IPv6 adrese, stačí kliknout na odkaz vybrané IPv6 adresy. Detailní výpis informací obsahuje:

- Souhrnné informace o IPv6 adrese
- Výpis posledních 30 ICMPv6 paketů
- Výpis posledních 30 TCP paketů



LaBrea6

Summary | Kippo - SSH Honeypot | WWW | SMTP Sink | Graphs | DNS queries | PhpMyAdmin

Last 200 DNS queries (127.0.0.1 is ignored)

time	ips	query
2013-04-30 10:18:45	194.228.140.150	gabless.ciz.zcu.cz IN A -ED (147.228.77.249)
2013-04-30 10:18:45	194.228.140.150	fly.gabless.ciz.zcu.cz IN A -ED (147.228.77.249)
2013-04-30 10:18:45	194.228.140.150	wiki.gabless.ciz.zcu.cz IN A -ED (147.228.77.249)
2013-04-30 08:51:35	65.55.37.37	wiki.gabless.ciz.zcu.cz IN A - (147.228.77.249)
2013-04-30 08:47:35	65.55.37.38	fly.gabless.ciz.zcu.cz IN A - (147.228.77.249)
2013-04-30 08:47:04	213.46.172.36	wiki.gabless.ciz.zcu.cz IN AAAA - (147.228.77.249)
2013-04-30 08:45:01	81.201.60.139	gabless.ciz.zcu.cz IN A -EDC (147.228.77.249)
2013-04-30 08:45:01	81.201.60.139	fly.gabless.ciz.zcu.cz IN A -EDC (147.228.77.249)
2013-04-30 08:45:01	81.201.60.139	wiki.gabless.ciz.zcu.cz IN A -EDC (147.228.77.249)
2013-04-30 08:43:57	213.46.172.36	wiki.gabless.ciz.zcu.cz IN A - (147.228.77.249)
2013-04-30 08:33:54	194.228.2.61	gabless.ciz.zcu.cz IN A -ED (147.228.77.249)
2013-04-30 08:33:54	194.228.2.61	fly.gabless.ciz.zcu.cz IN A -ED (147.228.77.249)
2013-04-30 08:33:54	194.228.2.61	wiki.gabless.ciz.zcu.cz IN A -ED (147.228.77.249)

Obrázek 6.7: Ukázka grafu *DNS queries*

- Výpis logu pro SSH honeypot *Kippo*
- Výpis logu z přístupů na webové stránky
- Výpis všech SMTP zpráv přijaté z této IPv6 adresy

6.8 Denní reporting

V rámci rozšíření webového administračního rozhraní byl nasazen denní reporting ve formě zasílání emailových zpráv. Byl vytvořen PHP skript `reporting.php`, který obsahuje kód pro načtení souhrnných informací o útocích z databáze a ty přepošle ve formě HTML (viz příloha A.4) na emailové adresy odběratelů denního reportingu. Pro denní spouštění tohoto skriptu byl využit plánovač úloh *Cron*.

```
0 6 * * * root /usr/bin/php /var/www/labrea6/reporting.php
```

6.9 Mobilní klient pro Android

V rámci předmětu FAV/KIV/MKZ byl vytvořen mobilní klient pro platformu *Android*. V klientu jsou zobrazovány souhrnné informace o útocích na

LaBrea6

Summary | Kippo - SSH Honeypot | WWW | SMTP Sink | Graphs | DNS queries | PhpMyAdmin

IP summary of 2a01:0430:0046:0000:0000:0000:0000:0002 (2a01:0430:0046:0000:0000:0000:0000:0002):

TOTAL tarpit count: **16**
Last tarpit at: 2013-04-16 06:30:48 (**29504m ago**)
TOTAL ICMPv6 count: **113**
Last ICMPv6 at: 2013-05-02 08:48:13 (**6326m ago**)
Is ownnet?: **NO**
Tartpit ports:

Last 30 ICMPv6 packets:

time	ipsrc	ipdst	type	hostname
2013-05-02 08:48:13	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3
2013-05-02 08:48:12	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3
2013-05-02 08:48:11	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3
2013-05-02 08:48:10	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3
2013-05-02 08:48:09	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3
2013-05-02 08:48:08	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3
2013-05-02 08:48:07	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3
2013-05-02 08:48:06	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3
2013-05-02 08:48:05	2a01:0430:0046:0000:0000:0000:0000:0002	2001:0718:1801:1077:baac:6fff:fe56:33a3	ECHO_REQUEST	2001:0718:1801:1077:baac:6fff:fe56:33a3

Last 30 tarpits:

time	ipsrc	psrc	ipdst	pdst	type	hostname
2013-04-16 06:30:48	2a01:0430:0046:0000:0000:0000:0000:0002	45089	2001:0718:1801:1077:baac:6fff:fe62:7618	22	TCP_ACK	2001:0718:1801:1077:baac:6fff:fe62:7618
2013-04-16 06:30:48	2a01:0430:0046:0000:0000:0000:0000:0002	45089	2001:0718:1801:1077:baac:6fff:fe62:7618	22	TCP_SYN	2001:0718:1801:1077:baac:6fff:fe62:7618
2013-04-02 10:34:10	2a01:0430:0046:0000:0000:0000:0000:0002	41970	2001:0718:1801:1077:baac:6fff:fe75:afa1	2555	TCP_ACK	2001:0718:1801:1077:baac:6fff:fe75:afa1
2013-04-02 10:34:10	2a01:0430:0046:0000:0000:0000:0000:0002	41970	2001:0718:1801:1077:baac:6fff:fe75:afa1	2555	TCP_SYN	2001:0718:1801:1077:baac:6fff:fe75:afa1
2013-03-12 17:50:18	2a01:0430:0046:0000:0000:0000:0000:0002	51643	2001:0718:1801:1077:baac:6fff:fe29:6a23	23	TCP_ACK	2001:0718:1801:1077:baac:6fff:fe29:6a23
2013-03-12 17:49:24	2a01:0430:0046:0000:0000:0000:0000:0002	51643	2001:0718:1801:1077:baac:6fff:fe29:6a23	23	TCP_ACK	2001:0718:1801:1077:baac:6fff:fe29:6a23
2013-03-12 17:48:57	2a01:0430:0046:0000:0000:0000:0000:0002	51643	2001:0718:1801:1077:baac:6fff:fe29:6a23	23	TCP_ACK	2001:0718:1801:1077:baac:6fff:fe29:6a23
2013-03-12 17:48:43	2a01:0430:0046:0000:0000:0000:0000:0002	51643	2001:0718:1801:1077:baac:6fff:fe29:6a23	23	TCP_ACK	2001:0718:1801:1077:baac:6fff:fe29:6a23
2013-03-12 17:48:37	2a01:0430:0046:0000:0000:0000:0000:0002	51643	2001:0718:1801:1077:baac:6fff:fe29:6a23	23	TCP_ACK	2001:0718:1801:1077:baac:6fff:fe29:6a23
2013-03-12 17:48:30	2a01:0430:0046:0000:0000:0000:0000:0002	51643	2001:0718:1801:1077:baac:6fff:fe29:6a23	23	TCP_ACK	2001:0718:1801:1077:baac:6fff:fe29:6a23
2013-03-12 17:48:27	2a01:0430:0046:0000:0000:0000:0000:0002	51643	2001:0718:1801:1077:baac:6fff:fe29:6a23	23	TCP_SYN	2001:0718:1801:1077:baac:6fff:fe29:6a23

Kippo - SSH honeypot log:

time	kippo_modul	session_id	ip_from	log_text
2013-04-02 23:19:06	HoneyPotTransport	1	2a01:0430:0046:0000:0000:0000:0000:0002	connection lost
2013-04-02 23:19:05	HoneyPotTransport	1	2a01:0430:0046:0000:0000:0000:0000:0002	Remote SSH version: SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7
2013-04-02 23:19:05	HoneyPotTransport	1	2a01:0430:0046:0000:0000:0000:0000:0002	kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
2013-04-02 23:19:05	HoneyPotTransport	1	2a01:0430:0046:0000:0000:0000:0000:0002	outgoing: aes128-ctr hmac-md5 none
2013-04-02 23:19:05	HoneyPotTransport	1	2a01:0430:0046:0000:0000:0000:0000:0002	incoming: aes128-ctr hmac-md5 none

WWW Application log:

Sink SMTP server log:

Obrázek 6.8: Ukázka souhrnných informací o IP

vytvořené IPv6 síti. Má poskytnout další variantu pro získávání aktuálních informací o útocích a snížit dobu reakce na případný útok.

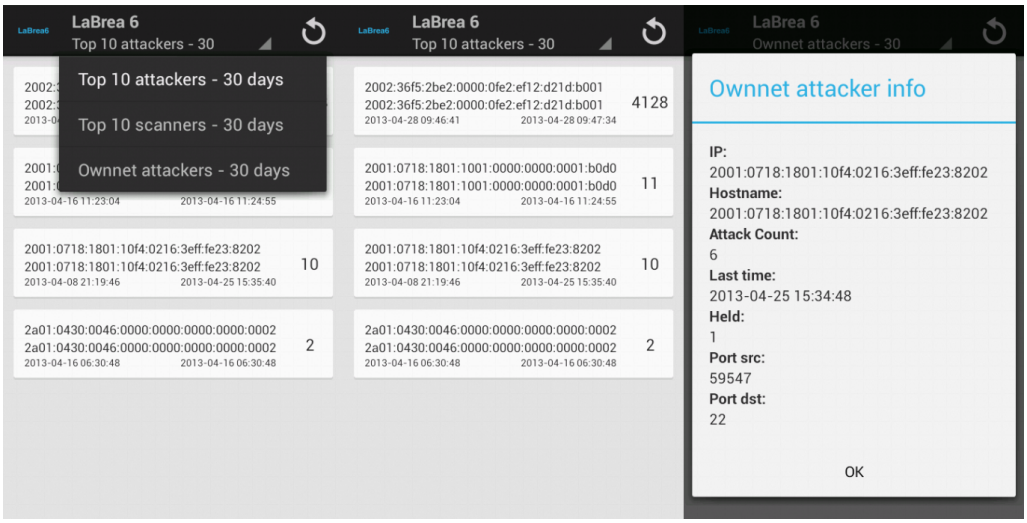
Aplikace obsahuje 3 základní záložky (viz obrázek 6.9):

- Top 10 attackers - 30 days

- Top 10 scanners - 30 days
- Ownnet attackers - 30 days

Každá záložka obsahuje seznam IPv6 adres spadající do příslušné kategorie a seřazené podle počtu útoků. V seznamu jsou uvedeny včetně IPv6 adresy také základní informace jako je například hostname, čas útoku nebo počet útoků provedené z konkrétní IPv6 adresy (viz obrázek 6.9).

Pro získání detailnějších informací o IPv6 adrese stačí otevřít dialogové okno dotykem na vybraný záznam. Okno obsahuje všechny dostupné informace o IPv6 adrese (viz obrázek 6.9).



Obrázek 6.9: Mobilní klient

6.10 Testování

U jednotlivých systémů se při testování kontrolovala správná interakce systému s útočníkem, ukládání korektních informací do lokální databáze a správné zobrazování statistik na webovém administračním prostředí.

Pro simulaci útočníka byly použity testovací stroj a služba *UnblockVPN*, která poskytuje VPN připojení na servery s IPv6 konektivitou rozmístěné na mnoha místech po celém světě.

LaBrea6

Systém *LaBrea6* bylo potřeba otestovat na 2 úrovních:

- Odpovědi na ICMPv6 pakety.
- Odpovědi na TCP pakety.

Pro testování odpovědí na ICMPv6 pakety typu ECHO REQUEST paketem ECHO:REPLY a NEIGHBOR SOLICITATION paketem NEIGHBOR ADVERTISEMENT byl použit síťový nástroj *ping6*, který se používá pro zjišťování dostupnosti testovaného IPv6 stroje pomocí ICMPv6 paketů ECHO REQUEST. Při použití tohoto nástroje byly otestovány oba typy ICMPv6 paketů díky vlastnosti IPv6 sítí, kdy pro doručení prvního paketu ECHO REPLY předchází zaslání paketu NEIGHBOR SOLICITATION.

V případě testování TCP odpovědí bylo potřeba vyvolat pakety typu TCP SYN, TCP ACK a TCP SYN/ACK. Tyto typy paketů jsou vyvolávány při tzv. *3-way handshaku* v rámci žádosti o SSH připojení. Proto byl pro otestování těchto typů paketů použit nástroj `ssh`.

Po rozšíření systému *LaBrea6* o ukládání kompletní komunikace systému *LaBrea6* s útočníkem ve formátu `pcap` (viz kapitola 6.2.1) bylo možné k ověření správnosti odesílaných paketů použit nástroj *Wireshark*.

Kippo - SSH honeypot

Testování SSH honeypotu probíhalo podobným způsobem jako u testování TCP odpovědí systému *LaBrea6*. Byl použit nástroj SSH ze vzdáleného IPv6 stroje. Byla provedena interakce se systémem *Kippo* a následná kontrola výstupního logu systému přes webové administrační prostředí.

SMTP Sink server

Pro testování *SMTP Sink* serveru byla na IPv6 stroji vytvořena skupina SMTP zpráv, které byly zaslány na emailové adresy pod doménou `ciz.zcu.cz`. Následovala kontrola detekce a právnosti informací SMTP zpráv přes webové administrační prostředí.

Webové aplikace

Pro simulování návštěvníka nasazených webových aplikací byla využita služba VPN od společnosti *UnblockVPN*, která poskytuje přístup na vzdálené servery s IPv6 konektivitou. Po připojení na takovýto server bylo možné přistupovat na stránky přes webový prohlížeč a vyvolat tak záznamy o přístupu na webové stránky, které byly kontrolovány na webovém administračním prostředí.

7 Nasazení

7.1 Instalace

Postup instalace softwarového balíku systému *LaBrea6* je detailně popsán v souboru `install.txt`, který je k nalezení na přiloženém CD v umístění `/labrea6/labrea6/install.txt` (viz příloha A.1).

7.2 Analýza útoků

Kompletní navržené prostředí se systémem *LaBrea6* pokrývající adresní prostor o velikosti 10 000 IPv6 adres bylo po otestování nasazeno včetně ostatních aplikací do reálného měsíčního provozu. Po dobu nasazení systém zaznamenal celkem 2 útoky.

7.2.1 Útok č.1

Webový honeypot zaznamenal návštěvníka webových stránek, který provedl dva přístupy na webové stránky aplikace *Joomla*.

Po důslednější analýze, bylo zjištěno, že se jedná o SPAM robota, který automaticky prohledává webové stránky a pomocí metody POST se snaží nalezeným stránkám předat náhodné parametry, které by mohla nalezená stránka akceptovat. SPAM robot maskovaný například za vyhledávač *Google* většinou útočí na blogy a stránky s komentáři. V našem případě se jednalo o tyto parametry:

```
{
        "blog_name": "Google",
        "url": "http: //www.google.com/tdtstols",
        "title": "Google",
        "excerpt": "here are some links to websites that we link to since
        we consider they are worth visiting"
}
```

7.2.2 Útok č.2

Druhý zaznamenaný útok byl proveden ve dvou po sobě jdoucích fázích. V první fázi útočník napadl webový honyepot a v druhé adresní prostor systému *LaBrea6* (viz dále). Útoky byly provedeny z rozdílných IPv6 adres:

```
2002:36f5:2be2:0000:dead:beef:dead:beef
2002:36f5:2be2:0000:0fe2:ef12:d21d:b001
```

Po provedené analýze těchto adres bylo zjištěno, že se jedná o technologii *6TO4*. Ta dovoluje komunikovat přes IPv4 sítě standardním protokolem IPv6. Používá k tomu techniku, kdy dojde k zabalení IPv6 hlavičky hlavičkou protokolu IPv4.

Tyto adresy jsou sice rozdílné, ale protože byla použita stejná překladová technologie pro jejich vytvoření a tato technologie má přesně definované schéma pro vytváření výsledné IPv6 adresy z původní IPv4 adresy, lze pomocí tohoto schéma lehce zjistit původní IPv4 adresu, ze které byl útok proveden.

První a druhý bajt adresy představuje prefix adres používaný pro identifikaci technologie *6TO4*. Třetí až šestý bajt adresy reprezentují onu hledanou IPv4 adresu:

```
36f5:2be2
```

```
54.245.43.226
```

Informace o nalezené IPv4 adrese popisuje tabulka A.5. Z informací lze vyčíst, že útok byl roveden z IP adresy patřící *Amazon Web Services*. Jedná se o amerického poskytovatele tzv. *Cloud* služeb (*PaaS*). Útočník zde nejspíše vlastní server, ze kterého byly útoky provedeny.

Webový honeypot

V první fázi útoku se útočník snažil napadnout webový honeypot. Pomocí metod GET, HEAD, DEBUG, INDEX, OPTION, PROPFIND, TRACK z nichž nejvyužívanější byla metoda GET s proměnnými parametry. Útočník

Popis	Hodnota
IP	`54.245.43.226`
Hostname	`ec2-54-245-43-226.us-west-2.compute.amazonaws.com`
ISP	`Amazon Technologies`
Stát	`USA`
Region	`Oregon`
Město	`Boardman`

Tabulka 7.1: Informace o nalezené IPv4 adrese

se snažil pomocí tohoto útoku zjistit slabiny webového serveru a získat potřebné informace k následnému napadnutí.

Informace o útoku:

Popis	Hodnota
Zdrojová IPv6	`2002:36f5:2be2:0000:dead:beef:dead:beef`
Zdrojová IPv4	`54.245.43.226`
Začátek útoku	`28.4.2013 9:14:40`
Konec útoku	`28.4.2013 9:18:09`
Doba trvání útoku	`3 minuty, 11 sekund`
Počet dotazů	`43`
URL napadané stránky	`http://gabless.ciz.zcu.cz/index.php`

Tabulka 7.2: Informace útoku na webový honeypot

Následuje příklad použitých parametrů v případě použití metody GET:

```
"searchword":   "<script>alert(document.cookie);</script>"
"option":       "search"
"page":         "../../../../../../../../etc/passwd"
"page":         "../../../../../../../../boot.ini"
"l":            "forum/view.php"
"topic":        "../../../../../../../../etc/passwd"
"download":     "/windows/win.ini"
"download":     "/etc/passwd"
"|":            "../../../../../../../../etc/passwd"
"download":     "/winnt/win.ini"
"mod":          "node"
"nid":          "some_thing"
"op":           "view"
"chemin":       "../../../../../../..//etc"
```

Podle parametru `HTTP_USER_AGENT`, který určuje z jakého softwaru návštěvník na webový server přichází, útočník prováděl útoky pomocí nástroje pro skenování webových serverů *Nikto*.

```
"HTTP_USER_AGENT":"Mozilla/4.75 (Nikto/2.1.4) (Evasions:None)
                  (Test:000703)"
```

Nikto je open source nástroj, který se používá pro testování zranitelnosti zvoleného webového serveru. Objevuje zranitelnosti pomocí následujících technik:

- Remote File Inclusion (RFI)
- Directory Traversal
- Cross-Site Scripting (XSS)
- SQL injection

Tento útok poukázal na bezpečnostní chybu webového administračního rozhraní. Webové administrační rozhraní nebylo ošetřeno před tzv. *Cross-Site Scripting* (*XSS*). Tato chyba se objevila při výpisu informací o IP adrese útočníka, kdy byl spuštěn skript, který vyvolal otevření tzv. *alert dialogu* s informacemi o *cookies*. Tento dialog vyvolal *javascript* „`<script> alert(document.cookie); </script>`“ vložený na stránky při zpětném načítání dat z databáze. Protože uložený řetězec nebyl ošetřen od speciálních znaků před zobrazením na cílové stránce, skript byl po načtení spuštěn. Chyba byla opravena ošetřením speciálních znaků před zobrazením pomocí PHP funkce `htmlspecialchars()`.

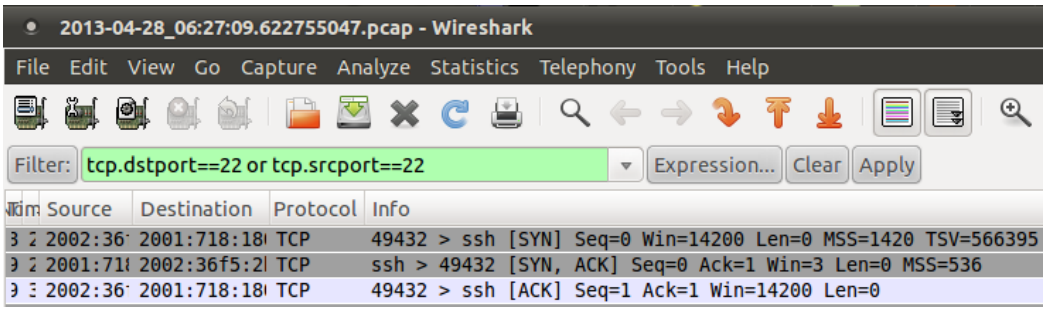
LaBrea6

V druhé fázi útoku se útočník zaměřil na dvě vybrané IPv6 adresy z navrženého adresního prostoru systému LaBrea6:

```
2001:0718:1801:1077:0000:0000:c0a8:2d92
2001:0718:1801:1077:baac:6fff:fe02:8b22
```

Protože z útočníkovi IPv6 adresy nebyl systémem *LaBrea6* zaznamenán žádný pokus o skenování navržené IPv6 sítě, útočník s velkou pravděpodobností použil pro získání těchto dvou IPv6 adres techniku skenovaní DNS záznamů (viz kapitola 4.4) domény `ciz.zcu.cz`, kterou znal z předchozí fáze útoku na webový honeypot.

K analýze útoků byl použit program *Wireshark*, v kterém byla zobrazena uložená komunikace systému *LaBrea6* s útočníkem v `pcap` formátu. Jak je vidět na obrázku 7.1, útočník prováděl na vybrané IPv6 adresy pokusy o TCP připojení tzv. *3-way handshake*. Útoky byly prováděny z náhodných neprivilegovaných portů v rozmezí 32805 - 60967 na celý rozsah portů cílové IPv6 adresy. Po dokončení TCP připojení komunikace nepokračovala.



Obrázek 7.1: Ukázka záložky *Ukázka analýzy útoku v programu* Wireshark

Informace o útoku:

Popis	Hodnota
Zdrojová IPv6	`2002:36f5:2be2:0000:0fe2:ef12:d21d:b001`
Zdrojová IPv4	`54.245.43.226`
Začátek útoku	`28.4.2013 9:46:41`
Konec útoku	`28.4.2013 9:47:34`
Doba trvání útoku	`53 sekund`
Počet dotazů	`4128`
Typ TCP paketů	`SYN, ACK`

Tabulka 7.3: Informace útoku na systém LaBrea6

7.3 Nevýhody použitých honeypotů

Při realizaci této práce bylo také zjištěno pár nevýhod či bezpečnostních rizik vybraných honeypotů.

7.3.1 LaBrea6

V průběhu realizace této práce byla zjištěna jedna z hlavních nevýhod IDS systému *LaBrea6*. Díky své povaze odpovídat na všechny příchozí pakety, se tento IDS systém stává lehce zneužitelným pro tzv. *bounce* útoky. Jedná se o typ útoku, kdy útočník využívá k provedení útoků i množinu jiných vzdálený strojů.

Příkladem může být nedávná kauza *DDoS* útoků na české webové servery jako jsou například *www.seznam.cz*, *www.idnes.cz* a bankovní servery jako např. *www.csob.cz*. Při těchto útocích dochází k vytvoření velké množiny požadavků na vybraný webový server. Takovýto webový server se snaží obsloužit všechny příchozí dotazy, stává se vytížen a pro normálního návštěvníka se stává nedostupným.

Útočník k vytvoření těchto požadavků využívá jiné stroje. Můžou to být stroje napadené útočníkem již v minulosti a čekající na jeho povel k útoku. Nebo útočník používá techniku, kdy posílá dotazy na vzdálený stroj se zdrojovou adresou cílového stroje, který chce útočník napadnout. V tomto případě pak zneužitý vzdálený stroj odpovídá na dotazy směrem k napadanému cíli a nevědomě se podílí na *DDoS* útoku. Toto je případ, kdy je systém *LaBrea6* vhodným zneužitelným cílem k provádění těchto útoků.

Vhodným řešením pro omezení možnosti zneužití IDS systému *LaBrea6* by mohlo být použití tzv. *packet rate limitation* neboli omezení počtu odpovídaných paketů za jednotku času na úrovni implementace systému *LaBrea6*. Zjednodušeně řečeno by si systém *LaBrea6* sám kontroloval počet odeslaných paketů za jednotku času.

7.3.2 Kippo

Chyba Kippo

Jedná se o chybu, která se projeví ve chvíli, kdy se útočník pokusí k SSH honeypotu *Kippo* připojit pomocí nástroje `telnet` místo `ssh`. V normálním případě se pokus o připojení k SSH přístupovému bodu pomocí nástroje `telnet` projeví následujícím výpisem:

```
root@slezids2:~# telnet 2001:0718:1801:1077:baac:6fff:fe38:8091 22
Trying 2001:718:1801:1077:baac:6fff:fe38:8091...
Connected to 2001:0718:1801:1077:baac:6fff:fe38:8091
(2001:718:1801:1077:baac:6fff:fe38:8091).
Escape character is '^]'.
SSH-2.0-OpenSSH_5.1p1 Debian-5
```

Což je standardní uvodní zpráva protokolu SSH. V případe *Kippo* je však situace jiná:

```
root@slezids2:~# telnet 2001:0718:1801:1077:baac:6fff:fec2:9334 22
Trying 2001:718:1801:1077:baac:6fff:fec2:9334...
Connected to 2001:0718:1801:1077:baac:6fff:fec2:9334
(2001:718:1801:1077:baac:6fff:fec2:9334).
Escape character is '^]'.
SSH-2.0-OpenSSH_5.1p1 Debian-5
dsafockdvxcvkmrowewfdiffie-hellman-group1-sha1ssh-rsaaes256-ctr,
aes256-cbc,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,cast128-ctr,
cast128-cbc,blowfish-ctr,blowfish-cbc,3des-ctr,3des-cbcaes256-ctr,
aes256-cbc,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,cast128-ctr,
cast128-cbc,blowfish-ctr,blowfish-cbc,3des-ctr,3des-cbchmac-sha1,
hmac-md5hmac-sha1,hmac-md5      none,zlib       none,zlib
```

V případě provedení spojení s *Kippo* se na obrazovce objeví nestandardní zpráva, který se přenáší při žádosti o spojení. Tato chyba by mohla útočníkovi prozradit, že se nejedná o reálný SSH server a tím ho odradit od provádění dalších útoků.

Jedná se o chybu ve frameworku *Twisted*, nad kterým *Kippo* běží. Chyba je způsobena vyvoláním předčasného *3-way handshaku*, což způsobuje předání těchto nesmyslných informací. Oprava této chyby spočívala v úpravě kódu frameworku *Twisted*. Upravený zdrojový soubor `transport.py` je k nalezení na přiloženém CD s umístěním:

`/labrea6/labrea6/config/backup/twisted/transport.py`

Zneužití příkazu `wget`

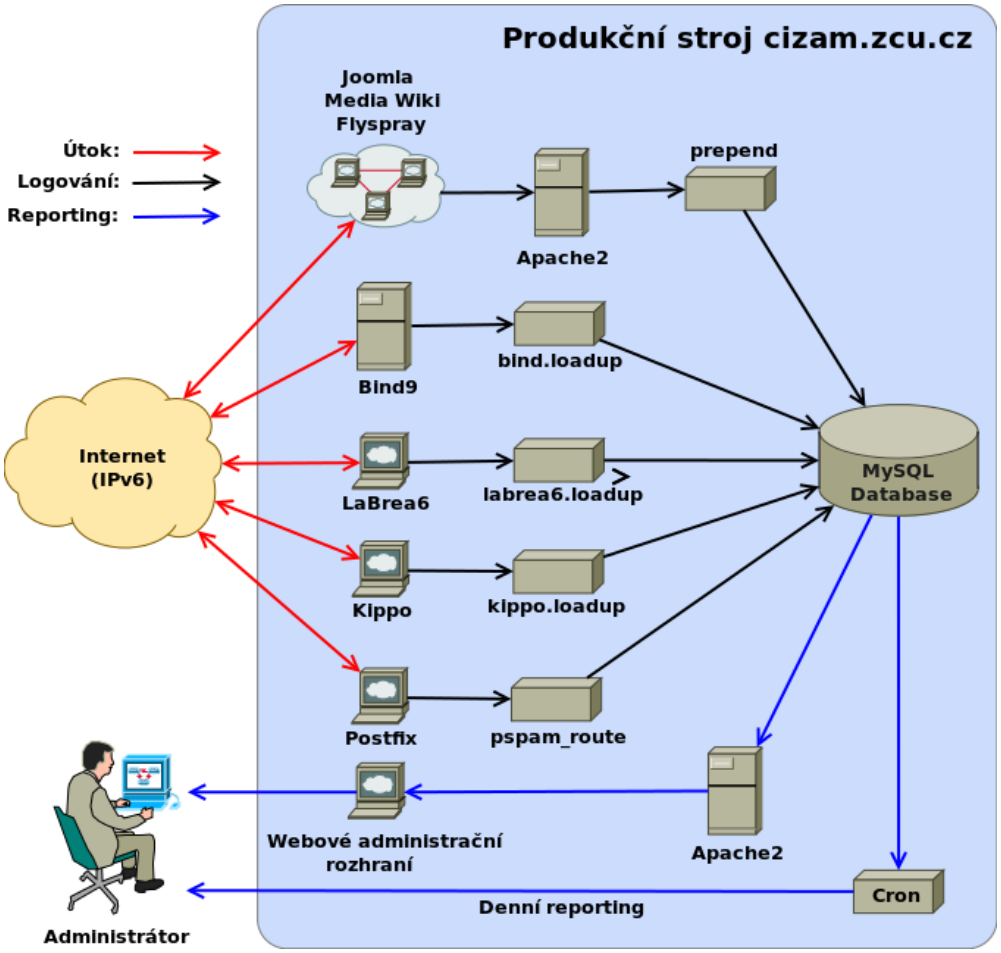
Příkaz `wget` je v *Kippo* použit pro analýzu útočníkem stažených souborů. Implementace tohoto příkazu však může představovat bezpečnostní rizika.

Jedním z takovýchto rizik je případ, kdy útočník začne stahovat velké množství dat za účelem zaplnit disk systému a tím ho vyřadit z provozu. Útočník by tímto ohrozil správnou funkčnost hostujícího stroje. Z tohoto důvodu je vhodné použít monitoring, který při nedostatku volného místa na disku vypne *Kippo*.

Jako další riziko může být případ, kdy je příkaz `wget` použit pro realizaci tzv. *DoS* útoku (Denial of Service). Útočník vytvoří velkou množinu `wget` dotazů na cílový server. Cílový server se snaží všechny tyto dotazy obsloužit a stává se nedostupným pro normálního uživatele.

7.4 Kompletní propojení nasazených systémů

Následující obrázek popisuje kompletní propojení všech navržených systémů na produkčním stroji.



Obrázek 7.2: Kompletní propojení nasazených systémů

8 Závěr

Cílem této práce bylo nasazení systému *LaBrea6* vytvořeného v rámci bakalářské práce FAV/KIV/INIB do reálného provozu v síti IPv6 a následný sběr informací o útocích, které by pomohly s preventivní obranou v tomto druhu sítí.

Na začátku práce byla provedena analýza původního stavu systému *LaBrea6* a nutných úprav pro nasazení do reálného provozu. Následovala analýza známých způsobů skenování adresního prostoru sítí IPv6. Ze získaných informací z provedené analýzy a zkušeností z bakalářské práce bylo navrženo IPv6 prostředí pro nasazení systému *LaBrea6*. Toto prostředí se skládá z navrženého adresního prostoru, na kterém systém *LaBrea6* poslouchá, z DNS serveru Bind9 pro mapování doménových jména na vytvořený adresní prostor a skupiny honeypotů sloužící pro doplňkový sběr informací o útocích. Takto navržené prostředí mělo za úkol navodit pocit, že se jedná o reálnou (napadnutelnou) IPv6 síť a přilákat útočníky.

V realizaci byl vytvořen generátor konfigurace, který má za úkol vytvářet konfigurace pro systémy *LaBrea6* a DNS server *Bind9* podle navrženého adresního prostoru. Dále byly provedeny všechny potřebné úpravy zvolených honeypotů k nasazení.

Nasazení navrženého IPv6 prostředí včetně systému *LaBrea6* do reálného provozu probíhalo v průběhu jednoho měsíce na adresním prostoru o velikosti 10 000 IPv6 adres. Během tohoto nasazení došlo k několika útokům do navržené sítě, které byly následně analyzovány.

Analýza ukázala, že se sice jednalo o útoky používané k vyhledávání napadnutelných míst koncových stanic, ale kvantita těchto útoků byla stále nízká (řádově se jednalo o jednotky) s porovnáním na sítích IPv4, kde je přítomnost těchto útoků běžná. Je třeba brát v úvahu, že IPv6 sítě stále rozšiřují své pole působnosti a je jen otázkou času, kdy útočníci přejdou ze sítí IPv4 na sítě IPv6.

Seznam obrázků

Seznam tabulek

Seznam zkratek

- **IPv4** - Internet protokol verze 4.
- **IPv6** - Internet protokol verze 6.
- **ZČU** - Západočeská univerzita v Plzni.
- **IDS** - Systém pro odhalení průniku (Intrusion Detection System) je v informatice obranný systém, který monitoruje síťový provoz a snaží se odhalit podezřelé aktivity.
- **CIV** - Centrum informatizace a výpočetní techniky Západočeské univerzity v Plzni.
- **WEBnet** - Počítačová síť Západočeské univerzity v Plzni.
- **DNS** - *Domain Name System* je hierarchický systém doménových jmen.
- **ICMPv6** - *Internet Control Message Protocol Version 6* je protokol pro zasílání kontrolních zpráv v IPv6 sítích.
- **DHCPv6** - *Dynamic Host Configuration Protocol Version 6*. Používá se pro automatické konfigurace hostů v IPv6 sítích.
- **MAC** - *Media Access Control* je jedinečný identifikátor síťového zařízení - Linková adresa.
- **VPN** - Virtuální privátní sítě (Virtual private network) - slouží k virtuálnímu propojení počítačů v Internetu. Komunikace přes VPN probíhá šifrovaně.
- **IPSec** - *IP security* je bezpečnostní rozšíření IP protokolu založené na autentizaci a šifrování každého IP datagramu.

- **ARP** - *Address Resolution Protocol* se v počítačových sítích s IPv4 protokolem používá k získání ethernetové MAC adresy sousedního stroje z jeho IP adresy.
- **MySQL** - Databázový systém.
- **TCP** - *Transmission Control Protocol* je jedním ze základních protokolů sady protokolů Internetu, konkrétně představuje transportní vrstvu.
- **UDP** - *User Datagram Protocol* je protokol transportní vrstvy bez potvrzování.
- **IID** - *Interface Identificator* je identifikátor rozhran9 síťové karty.
- **OUI** - *Organizationally unique identifier* je identifikátor výrobce použitý pro konfiguraci MAC adresy síťové karty.
- **SEND** - *Secure Neighbor Discovery Protocol* je nutným rozšířením protokolu ICMPv6 jako zabezpečení proti tzv. *arp cache poisoning* útokům.
- **SLACC** - *StateLess Address Auto Configuration* je bezstavová konfigurace IPv6 adres.
- **SSH** - *Secure Shell* je protokol používaný pro přístup k vzdáleným strojům.
- **SMTP** - *Simple Mail Transfer Protocol* slouží pro zasílání emailových zpráv.
- **DoS** - *Denial of Service* je útok, při kterém útočník na cílový server vyvolá tisíce dotazů za vteřinu z jedné IP adresy a tím zaneprázdní server. Server se stává nedostupným pro normálního uživatele.
- **DDoS** - *Dynamic Denial of Service* je vylepšená verze DoS útoku, kdy dochází k útokům z množiny jiných strojů vlastněných útočníkem, tedy né z jedné IP, jak je to v případě DoS.
- **6TO4** - překladová technologie pro možnost komunikace standardním protokolem IPv6 přes sítě IPv4
- **PaaS** - *Platform as a Service* je jedna z kategorií *Cloud* technologií.
- **XSS** - *Cross-Site scripting* je styl útoku, který využívá spouštění vlastních skriptů na cizích stránkách.

Literatura

[1] SATRAPA Pavel. *Protokol IPV6.* Praha : CZ.NIC, z. s. p. o. , 2008. ISBN 978-80-904248-0-7

[2] DOSTÁLEK Libor. *Velký průvodce protokoly TCP/IP a DNS.* ISBN 80-7226-323-4

[3] HEROUT Pavel. *Učebnice jazyka C 1.díl.* ISBN: 80-80828-21-9

[4] HEROUT Pavel. *Učebnice jazyka C 2.díl.* ISBN: 80-85828-50-2

[5] JIŘIČKA Martin. *Síť ové útoky* DP ZČU 2004/2005

[6] TICHA Ondřej. *Síť ové útoky* DP ZČU 2005/2006

[7] LUDVÍK Libor. *Systém pro automatickou bezpečnostní analýzu NetFlow informací* DP ZČU 2005/2006

[8] SLEZÁČEK Antonín. *IDS systém typu Tarpit pro IPv6* BP ZČU 2009/2010

[9] PROVOS N., HOLZ T. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection* ISBN: 978-0321336323

[10] RUVALCABA Cristian. *SMART IDS – HYBRID LABREA TARPIT* Prosinec 2009 [cit. 8.5.2011].

[11] BODÓ Radoslav, PADRTA Aleš: *Rozvoj systémů pro detekci průniků v síti WEBnet* Závěrečná zpráva FR Cesnet projektu 230R2/2007 Dostupné z `http://fondrozvoje.cesnet.cz/projekt.aspx?ID=230`

[12] VACHEK Pavel. *CESNET Intrusion Detection systém* Technická zpráva CESNETu číslo 5/2006
Dostupné z `http://www.cesnet.cz/doc/techzpravy/2006/ids/`

[13] GONT Fernando, CHOWN Tim: *Network Reconnaissance in IPv6 Networks* draft-ietf-opsec-ipv6-host-scanning-00 12/2012 [cit. 15.4.2013].
Dostupné z `http://tools.ietf.org/html/draft-ietf-opsec-ipv6-host-scanning-00`

[14] *Scapy* – packet generator [cit. 23.3.2011].
Dostupné z `http://www.secdev.org/projects/scapy/`

[15] *IP Version 6 Addressing Architecture* Únor 2006 [cit. 5.5.2011].
Dostupné z `http://tools.ietf.org/html/rfc4291`

A Přílohy

A.1 Uživatelská dokumentace

```
*********************************************************
*   Uzivatelska dokumentace *
*********************************************************
Poznamka:
---
LaBrea6 pracuje jen v˜operacnim systemu GNU/LINUX!
---



>Instalace LaBrea6:
=========================================================

1) Nainstalujte si vsechny potrebne baliky prikazem:
---
$ apt-get install libpcap0.8-dev php5 dnsutils
libnet-cidr-perl libnet-subnets-perl libbit-vector-perl
libcarp-clan-perl bind9 git
---

2) Pro spravny chod celeho systemu LaBrea6 je nutne nain-
stalovat MySQL databazi s˜administraci phpMyadmin a Web
server Apache:
---
$ apt-get install  mysql-server phpmyadmin apache2
---

3) Je nutne pouzivat knihovnu libpcap zkompilovanou pro
IPv6. Pokud kompilujete knihovnu libpcap manuálně, nastavte
konfiguracni soubor knihovny libpcap pred kompilaci prikazem:
---
./configure --enable-ipv6
---

4) Vytvorte adresar /opt/labrea6/ do ktereho zkopirujte
cely balik LaBrea6. Adresarova struktura musi vypadat
takto:
===
/opt/labrea6/
|-- bin (spustitelne soubory)
|   |-- labrea6
```

```
|   `-- labrea6-config
|-- config (veskere konfiguracni soubory)
|   |-- backup (zaloha konfiguracnich souboru ostatnich systemu)
|   |   |-- bind
|   |   |   |-- named.conf.local
|   |   |   `-- named.conf.options
|   |   |-- deploy_scripts
|   |   |   |-- labrea6.deploy
|   |   |   `-- labrea6-www.deploy
|   |   |-- php
|   |   |   `-- php.ini
|   |   |-- postfix
|   |   |   |-- main.cf
|   |   |   `-- master.cf
|   |   `-- www_loadup
|   |       `-- request_log.php
|   |-- db.ciz.zcu.cz (konfigurace Bind9)
|   `-- ip_list.txt   (konfigurace LaBrea6)
|-- install.txt
|-- other_apps (zdrojove soubory ostatnich honeypotu)
|   |-- kippo
|   |   |-- kippo-0.5
|   |   |-- kippo.init
|   |   |-- kippo.loadup
|   |   |-- kippo.start
|   |   `-- kippo.stop
|   `-- sink
|       `-- sink.loadup.php
|-- script (skripty systemu LaBrea6)
|   |-- labrea6.ignore
|   |-- labrea6.init
|   |-- labrea6.loadup
|   |-- labrea6.logrotate
|   |-- labrea6.reconf
|   `-- labrea6.sql
|-- src (zdrojove soubory ...)
|   |-- labrea6 (...systemu LaBrea6)
|   |   |-- answers.c
|   |   |-- answers.h
|   |   |-- answers.o
|   |   |-- catchPacket.c
|   |   |-- catchPacket.h
|   |   |-- catchPacket.o
|   |   |-- ipList.c
|   |   |-- ipList.h
|   |   |-- ipList.o
|   |   |-- main.c
|   |   |-- main.o
|   |   |-- Makefile
```

```
|   |   |-- packets.h
|   |   |-- utils.c
|   |   |-- utils.h
|   |   `-- utils.o
|   `-- labrea6-config (...generatoru konfigurace)
|       |-- config
|       |   `-- nouns.txt
|       `-- src
|           |-- generator.c
|           |-- generator.h
|           |-- generator.o
|           |-- genfunc.c
|           |-- genfunc.h
|           |-- genfunc.o
|           |-- list.c
|           |-- list.h
|           |-- list.o
|           |-- main.c
|           |-- main.o
|           `-- Makefile
|-- var (logy systemu LaBrea6)
`-- www (zdrojove soubory administracniho weboveho rozhrani)
===

5) V~adresari /opt/labrea6/src/labrea6 zkompilujte zdrojove kody
systému LaBrea6 prikazy:
---
sudo make clean
sudo make
---
Spustitelny soubor je zkompilovan do /opt/labrea6/bin/labrea6

6) V~adresari /opt/labrea6/src/labrea6-config zkompilujte zdrojove kody
generatoru konfigurace systemu LaBrea6 a DNS serveru Bind9 prikazy:
---
sudo make clean
sudo make
---
Spustitelny soubor je zkompilovan do /opt/labrea6/bin/labrea6-config


7) Vytvorte soft-link do /etc/init.d/ na init skript systemu LaBrea6
a prilozeneho SSH honeypotu Kippo prikazy:
---
ln -s /opt/labrea6/script/labrea6.init /etc/init.d/labrea6
ln -s /opt/labrea6/other_apps/kippo/kippo.init /etc/init.d/kippo
---

8) Vytvorte soft-link do /etc/logrotate.d/ na logrotate
```

```
skript labrea6.logrotate pro rotaci logu prikazem:
---
ln -s /opt/labrea6/script/labrea6.logrotate /etc/logrotate.d/labrea6
---

9) Vytvorte soft-link do /etc/bind/ na zonovy soubor systemu LaBrea6:
ln -s /opt/labrea6/config/db.ciz.zcu.cz /etc/bind/db.ciz.zcu.cz

10) Nastavte DNS server Bind9 podle vzoru konfiguracnich souboru:
/opt/config/backup/bind/named.conf.local
/opt/config/backup/bind/named.conf.options

!!! KROK 11 az 13 jsou nepovinne pro beh samotneho systemu LaBrea6 !!!

11) Nastavte soubor request_log.php pro funkci prepend u~weboveho
serveru Apache pro ukladani www logu do databaze pridanim nasledujici
radky do souboru /etc/php5/apache2/php.ini (viz vzor
/opt/labrea6/config/ backup/php/php.ini):
---
auto_prepend_file = /var/www/www-log/request_log.php
---

12) Pro nastaveni Postfix mailoveho serveru pouzijte vzory nastaveni
/opt/labrea6/config/backup/postfix (main.cf, master.cf a transport),
ktere umistete na umisteni:
---
/etc/postfix/
---
Jeste je nutne vygenerovat konfiguraci nastaveni transportni mapy
prikazem:
---
postmap /etc/postfix/transport
---

13) Pro implementaci vsech nastavenich restartujte vsechny systemy:
---
/etc/init.d/apache2 restart
/etc/init.d/bind9 restart
/etc/init.d/postfix restart
---

>Instalace MySQL databaze:
============================================================
Predpoklady:
- Balik LaBrea6 je umisten v~/opt/labrea6/.
- Nainstalovany web server (Apache) a MySQL
databaze:
---
$ apt-get install  mysql-server apache2
```

```
---

1) Prihlaste se k~MySQL databazi:
===
mysql -u root -p
===

2) Vytvorte novou databazi labrea6:
===
mysql> CREATE DATABASE labrea6;
===

2) Vytvorte nove tabulky tarpit a icmp podle vzoru speci-
fikovanych v~souboru /opt/labrea6/script/labrea6.sql (doporucuji
pouzit import):
===
-- phpMyAdmin SQL Dump
-- version 3.3.7deb7
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: May 07, 2013 at 04:19 PM
-- Server version: 5.1.63
-- PHP Version: 5.3.3-7+squeeze15

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `labrea6`
--

-- --------------------------------------------------------

--
-- Table structure for table `bind9`
--

CREATE TABLE IF NOT EXISTS `bind9` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `time` int(11) NOT NULL,
  `ipsrc` varchar(39) NOT NULL,
  `query` text NOT NULL,
  PRIMARY KEY (`id`),
```

```
  KEY `ipsrc` (`ipsrc`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=9144 ;

-- --------------------------------------------------------

--
-- Table structure for table `icmp`
--

CREATE TABLE IF NOT EXISTS `icmp` (
  `time` int(11) NOT NULL,
  `ipsrc` varchar(39) NOT NULL,
  `ipdst` varchar(39) NOT NULL,
  `type` varchar(30) DEFAULT NULL,
  KEY `bytime` (`time`),
  KEY `byipsrc` (`ipsrc`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 MAX_ROWS=4294967295
AVG_ROW_LENGTH=512;

-- --------------------------------------------------------

--
-- Table structure for table `kippo`
--

CREATE TABLE IF NOT EXISTS `kippo` (
  `time` datetime NOT NULL,
  `kippo_modul` varchar(500) COLLATE utf8_unicode_ci NOT NULL,
  `session_id` int(11) NOT NULL,
  `ip_from` varchar(39) COLLATE utf8_unicode_ci NOT NULL,
  `log_text` text COLLATE utf8_unicode_ci NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-- --------------------------------------------------------

--
-- Table structure for table `sink`
--

CREATE TABLE IF NOT EXISTS `sink` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `time` int(11) NOT NULL,
  `ipsrc` varchar(39) NOT NULL,
  `email` text NOT NULL,
  `metadata` text NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=16 ;

-- --------------------------------------------------------
```

```
--
-- Table structure for table `tarpit`
--

CREATE TABLE IF NOT EXISTS `tarpit` (
  `time` int(11) NOT NULL,
  `ipsrc` varchar(39) NOT NULL,
  `psrc` int(11) NOT NULL,
  `ipdst` varchar(39) NOT NULL,
  `pdst` int(11) NOT NULL,
  `type` varchar(30) DEFAULT NULL,
  KEY `bytime` (`time`),
  KEY `byipsrc` (`ipsrc`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 MAX_ROWS=4294967295
AVG_ROW_LENGTH=512;

-- --------------------------------------------------------

--
-- Table structure for table `www`
--

CREATE TABLE IF NOT EXISTS `www` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `time` int(11) NOT NULL,
  `ipsrc` varchar(39) NOT NULL,
  `type` varchar(20) NOT NULL,
  `server_data` text NOT NULL,
  `get_data` text NOT NULL,
  `post_data` text NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5746 ;
===

3) Zmente prihlasovaci udaje k~databazi (nize vypsane
radky) v~souborech:

/opt/labrea6/script/labrea6.loadup
/opt/labrea6/other_apps/kippo/kippo.loadup:
---
my $host = "127.0.0.1";
my $db = "labrea6";
my $user = "vase uzivatelske jmeno";
my $pass = "vase heslo k~MySQL databazi";
---

/opt/labrea6/www/db.inc.php
/opt/labrea6/www/www-log/request_log.php
```

```
/opt/labrea6/other_apps/sink/sink.loadup
---
$host = "127.0.0.1";
$db = "labrea6";
$user = "vase uzivatelske jmeno";
$pass = "vase heslo k~MySQL databazi";
---



>Instalace weboveho rozhrani:
============================================================
Predpoklady:
- Balik LaBrea6 je umisten v~/opt/labrea6/.
- Nainstalovany web server (Apache) a MySQL
databaze:
---
$ apt-get install  mysql-server apache2
---

1) Zkopirujte veskery obsah adresare /opt/labrea6/www/ do
adresare vaseho web serveru (Apache). Defaultne do
/var/www/:
---
cp -r /opt/labrea6/www/* /var/www/
---



>Spusteni LaBrea6:
============================================================

Ovladani systemu LaBrea6 s~vygenerovanou konfiguraci je
nasledujici:
1) START:
---
/etc/init.d/labrea6 start
---

2) STOP:
---
/etc/init.d/labrea6 stop
---

3) RESTART:
---
/etc/init.d/labrea6 restart
---
```

```
4) Vypis konce logu:
---
/etc/init.d/labrea6 log
---

5) Vygenerovani novych konfiguracnich souboru pro system
LaBrea6 a DNS server Bind9 + restart obou systemu:
---
/etc/init.d/labrea6 reconfigure
---


>Spusteni SSH honeypotu Kippo:
============================================================
Ovladani SSH honeypotu Kippo je nasledujici:
1) START:
---
/etc/init.d/kippo start
---

2) STOP:
---
/etc/init.d/kippo stop
---

3) RESTART:
---
/etc/init.d/kippo restart
---


>Spusteni ostatních honeypotu:
============================================================
>>POSTFIX honeypot:
------------------------------------------------------------
Nastaveni Postfix mail serveru bylo probrano (provedeno)
drive. Start Postfix serveru se provede pomoci:
/etc/init.d/postfix start
------------------------------------------------------------

>>WWW honeypot:
------------------------------------------------------------
Nastaveni WWW honeypotu bylo probrano (provedeno) drive.
Start WWW serveru se provede pomoci:
/etc/init.d/apache2 start
------------------------------------------------------------

>>DNS honeypot:
------------------------------------------------------------
```

```
Nastaveni DNS honeypotu bylo probrano (provedeno) drive.
Start DNS serveru se provede pomoci:
/etc/init.d/bind9 start
------------------------------------------------------------
```

A.2 Adresářová struktura balíku LaBrea6

```
‘-- bin (spustitelne soubory)
|   |-- labrea6
|   ‘-- labrea6-config
|-- config (konfiguracni soubory)
|   |-- backup (zaloha konfiguracnich souboru ostatnich systemu)
|   |   |-- bind
|   |   |   |-- named.conf.local
|   |   |   ‘-- named.conf.options
|   |   |-- deploy_scripts
|   |   |   |-- labrea6.deploy
|   |   |   ‘-- labrea6-www.deploy
|   |   |-- php
|   |   |   ‘-- php.ini
|   |   |-- postfix
|   |   |   |-- main.cf
|   |   |   ‘-- master.cf
|   |   ‘-- www_loadup
|   |       ‘-- request_log.php
|   |-- db.ciz.zcu.cz (konfigurace Bind9)
|   ‘-- ip_list.txt   (konfigurace LaBrea6)
|-- install.txt
|-- other_apps (zdrojove soubory ostatnich honeypotu)
|   |-- kippo
|   |   |-- kippo-0.5
|   |   |-- kippo.init
|   |   |-- kippo.loadup
|   |   |-- kippo.start
|   |   ‘-- kippo.stop
|   ‘-- sink
|       ‘-- sink.loadup.php
|-- script (skripty systemu LaBrea6)
|   |-- labrea6.ignore
|   |-- labrea6.init
|   |-- labrea6.loadup
|   |-- labrea6.logrotate
|   |-- labrea6.reconf
|   ‘-- labrea6.sql
|-- src (zdrojove soubory ...)
|   |-- labrea6 (...systemu LaBrea6)
|   |   |-- answers.c
|   |   |-- answers.h
|   |   |-- answers.o
|   |   |-- catchPacket.c
|   |   |-- catchPacket.h
|   |   |-- catchPacket.o
|   |   |-- ipList.c
```

```
|   |   |-- ipList.h
|   |   |-- ipList.o
|   |   |-- main.c
|   |   |-- main.o
|   |   |-- Makefile
|   |   |-- packets.h
|   |   |-- utils.c
|   |   |-- utils.h
|   |   `-- utils.o
|   `-- labrea6-config (...generatoru konfigurace)
|       |-- config
|       |   `-- nouns.txt
|       `-- src
|           |-- generator.c
|           |-- generator.h
|           |-- generator.o
|           |-- genfunc.c
|           |-- genfunc.h
|           |-- genfunc.o
|           |-- list.c
|           |-- list.h
|           |-- list.o
|           |-- main.c
|           |-- main.o
|           `-- Makefile
|-- var (logy systemu LaBrea6)
`-- www (zdrojove soubory administracniho weboveho rozhrani)
```

A.3 Kofigurační soubor master.cfg

```
pspam_route unix -    n    n    -    -    pipe
flags=FR
user=kippo
argv=/opt/labrea6/other_apps/sink/sink.loadup.php
{
        "client_address":"${client_address}",
        "client_helo":"${client_helo}",
        "client_hostname":"${client_hostname}",
        "client_port":"${client_port}",
        "client_protocol":"${client_protocol}",
        "domain":"${domain}",
        "extension":"${extension}",
        "mailbox":"${mailbox}",
        "original_recipient":"${original_recipient}",
        "recipient":"${recipient}",
        "sasl_method":"${sasl_method}",
        "sasl_sender":"${sasl_sender}",
        "sasl_username":"${sasl_username}",
        "sender":"${sender}",
        "size":"${size}",
        "user":"${user}"
}
```

A.4 Ukázka denního reportingu

Overall

Time frame: 2013-02-19 11:06:47 - 2013-04-28 09:47:34
Last tarpit at: 2013-04-28 09:47:34 (**1212m ago**) from **2002:36f5:2be2:0000:0fe2:ef12:d21d:b001** (
2002:36f5:2be2:0000:0fe2:ef12:d21d:b001)

Ownnet attackers for 24 hours

Port trends

Legend:
Port rise
Port fall
Well-known port
Registered port
Dynamic/private/local port
trend = count(dpst) - avg(last 24 hours)

pdst/name	count	count72	trendRatio
80/www	8	0	8
443/https	8	0	8
2042/	8	0	8
55056/	7	0	7
8045/	7	0	7
545/	7	0	7
1594/	5	0	5

Top 10 attackers for last 24 hours

ips	count	first	last	hostname
2002:36f5:2be2:0000:0fe2:ef12:d21d:b001	4128	2013-04-28 09:46:41	2013-04-28 09:47:34	2002:36f5:2be2:0000:0fe2:ef12:d21d:b001

Top 10 ICMPv6 attackers for last 24 hours

ips	count	first	last	hostname
2001:0718:1801:10f4:0216:3eff:fe23:8202	39	2013-04-28 19:59:15	2013-04-28 20:51:02	2001:0718:1801:10f4:0216:3eff:fe23:8202
fe80:0000:0000:0000:0208:e3ff:feff:fc2c	8	2013-04-28 09:46:41	2013-04-28 20:51:16	fe80:0000:0000:0000:0208:e3ff:feff:fc2c

SSH attackers for last 24 hours

time	ips	session_count	hostname
2013-04-28 14:44:39	2001:0718:1801:10f4:0216:3eff:fe23:8202	4	2001:0718:1801:10f4:0216:3eff:fe23:8202
2013-04-28 09:21:39	0000:0000:0000:0000:0000:ffff:93e4:0185	5	bodik.zcu.cz

WWW visitors for last 24 hours

time	ips	hostname
2013-04-28 09:14:40	2002:36f5:2be2:0000:dead:beef:dead:beef	2002:36f5:2be2:0000:dead:beef:dead:beef
2013-04-28 08:07:17	2001:4860:4801:2004:0000:6006:1300:b075	crawl-2001-4860-4801-2004-0000-6006-1300-b075.googlebot.com
2013-04-28 08:06:27	2001:4860:4801:2001:0000:6006:1300:b075	crawl-2001-4860-4801-2001-0000-6006-1300-b075.googlebot.com
2013-04-28 08:05:47	2001:4860:4801:2002:0000:6006:1300:b075	crawl-2001-4860-4801-2002-0000-6006-1300-b075.googlebot.com
2013-04-28 08:05:40	2001:4860:4801:2003:0000:6006:1300:b075	crawl-2001-4860-4801-2003-0000-6006-1300-b075.googlebot.com

SMTP messages for last 24 hours

A.5 Informace o IPv4 adrese útočníka

General IP Information

IP: 54.245.43.226
Decimal: 922037218
Hostname: ec2-54-245-43-226.us-west-2.compute.amazonaws.com
ISP: Amazon Technologies
Organization: Amazon Technologies
Services: None detected
Type: Corporate
Assignment: Static IP
Blacklist: Blacklist Check

Geolocation Information

Country: United States
State/Region: Oregon
City: Boardman
Latitude: 45.7788 (45° 46′ 43.68″ N)
Longitude: -119.529 (119° 31′ 44.40″ W)
Area Code: 541
Postal Code: 97818

A.6 Ukázka výstupních grafů