Přírodovědecká fakulta Jihočeské univerzity

Bakalářská práce:

Detekce červů a botnetů pomocí volně šiřitelného software

Vypracoval: Dominik Marek

Školitel: Ing. Petr Břehovský

České Budějovice 2011

Bibliografické údaje

Marek D., 2011: Detekce červů a botnetů pomocí volně šiřitelného software.

[Detection of worms and botnets using free software. Bc. Thesis, in Czech.] – 46 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Abstract

This bachelor thesis aims to introduce methods of detection of worms and other malicious code using free software and analysis of malware's network activity. This thesis presents such methods that can be used for protecting personal or corporate networks, along with their advantages and disadvantages. The work also comprises description and practical application of honeypot detection method in the university network. The analysis of malware's network activity is based on data collected by the honeypot. Furthermore, this thesis deals with the structure and function of botnets, which is necessary to understand how to defend against them.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

27. dubna 2011

Dominik Marek

Poděkování

Děkuji svému školiteli Ing. Petru Břehovskému za odborné vedení práce, cenné rady a připomínky a pomoc při řešení problémů. Dále děkuji Radoslavu Bodóovi za jednání se sdružením CESNET a Lence Novákové za pomoc při korektuře textu.

OBSAH

SLOVNÍČEK POJMŮ **1**

SEZNAM OBRÁZKŮ **1**

1 ÚVOD **2**

1.1 CÍLE PRÁCE 2

2 STRUKTURA A FUNGOVÁNÍ BOTNETŮ **3**

2.1 CO TO JE BOTNET 3
2.2 SLOŽENÍ A TOPOLOGIE BOTNETŮ 3
2.3 TYPY BOTNETŮ 4
2.4 JAK SE Z POČÍTAČE STANE BOT-KLIENT 7
2.5 KOMUNIKACE S BOT-MASTEREM 8
2.6 VYUŽITÍ BOTNETŮ 8

3 MOŽNOSTI DETEKCE **10**

3.1 HOST-BASED DETEKCE 10
3.2 NETWORK-BASED DETEKCE 13
3.3 OSTATNÍ MOŽNOSTI 16
3.4 SHRNUTÍ 17
3.5 PŘEHLED MOŽNOSTÍ DETEKCE 17

4 PRAKTICKÁ APLIKACE HONEYPOTU **18**

4.1 INSTALACE SERVERU 19
4.2 HONEYPOT DIONAEA 21

5 ANALÝZA ČINNOSTI MALWARE **28**

5.1 PŘEHLED ÚTOKŮ 28
5.2 ÚTOKY NA MSSQL 29
5.3 ANALÝZA ZACHYCENÉHO VZORKU MALWARE 30
5.4 AKTIVITA VE VNITŘNÍ SÍTI 33

6 NÁVRHY NA DALŠÍ PRÁCI **34**

7 ZÁVĚR **35**

8 SEZNAM POUŽITÉ LITERATURY **36**

PŘÍLOHY **38**

Slovníček pojmů

Malware – z anglického Malicious Software, škodlivý software (též škodlivý kód) po spuštění v počítači vykonává nějakou škodlivou aktivitu (např. sken sítě, krádeže citlivých informací apod.)

Exploit – škodlivý kód, který zneužívá chyby v systému, aby mohl být vykonán

Bot – malware, který umožní vzdálené nelegitimní ovládnutí počítače

Bot-klient – počítač začleněný do botnetu

Botnet – síť nakažených počítačů, která je vzdáleně nelegitimně ovládána

Bot-master/ bot-herder - tvůrce botnetu či ten, který botnetu zadává příkazy

C&C – Command&Control server – server jehož pomocí bot-master předává příkazy botnetu

Zero-day útok/malware – útok či malware, který ještě není obecně znám (zatím pro něj neexistuje obrana)

Honeypot – systém, který obsahuje nebo simuluje známé zranitelnosti, na které láká útočníky a malware

Shellcode – krátký vykonatelný kód, který je použit při zneužití zranitelnosti systému k vykonání příkazu

Seznam obrázků

Obrázek 1 – Komunikace v botnetu pomocí IRC protokolu a několika IRC serverů

Obrázek 2 - Navázání spojení protokolem TCP.

Obrázek 3 – Útok typu SYN Flood

Obrázek 4 – Tabulka nejfrekventovanějších portů

Obrázek 5 – Tabulka použitých přihlašovacích jmen

Obrázek 6 – Tabulka pěti nejpoužívanějších hesel

1 Úvod

Na začátku třetího tisíciletí, kdy se vývoj nových technologií ubírá kupředu „mílovými“ kroky, si jen stěží dokážeme představit náš život bez počítačů. Podle Mezinárodní telekomunikační unie (ITU) počet uživatelů internetu překročil v roce 2010 hranici dvou miliard (POŠVIC, K. 2011) a toto číslo se bude ještě zvětšovat. V tomto ohromném množství uživatelů je největším problémem stálé opomíjení bezpečnosti.

S rostoucím počtem legitimních uživatelů internetu roste i počet těch uživatelů, kteří se chtějí obohatit nelegálním způsobem. Jedná se o různé hackery, tvůrce škodlivých kódů či exploitů a bot-mastery. Každým dnem přibývají nové unikátní vzorky malware, které soukromí uživatelů ohrožují.

A právě malware může zapříčinit začlenění našeho počítače do botnetu. Mnoho společností zabývajících se problematikou bezpečnosti internetu považuje právě botnety za největší a neustále se vyvíjející hrozbu, proto by se měl klást větší důraz na bezpečnost a to jak na prevenci, aby nedošlo k infekci počítače, tak na detekci, aby případná infekce byla včas zaznamenána a následně odstraněna.

Tato práce se zabývá detekcí počítačových červů a botnetů a přispívá k orientaci v této problematice. Měla by pomoci při výběru vhodných metod detekce jak na osobních počítačích, tak i ve vnitřní síti. Zabývá se strukturou a fungováním botnetů, což je důležité pro pochopení, jak se těmto hrozbám bránit. Je zde popsána praktická aplikace honeypotu do počítačové sítě a následná analýza zaznamenaných útoků.

1.1 Cíle práce

- seznámit se se strukturou a fungováním botnetů a následně tuto problematiku popsat
- seznámit se s možnostmi detekce červů a botnetů a vytvořit seznam těchto možností spolu s volně použitelnými nástroji.
- praktické vyzkoušení a aplikace metody detekce pomocí honeypotu
- analýza výsledků detekce

2 Struktura a fungování botnetů

2.1 Co to je botnet

Škodlivý software neboli malware (z anglických Malicious Software) je kód nebo program, který má vykonávat nějakou škodlivou aktivitu v počítači, jako například umožnit nelegitimní vzdálené ovládnutí počítače. Pokud se jedná přímo o tuto aktivitu, je tento malware nazýván bot (ze slova robot). Definice podle společnosti Microsoft říká, že bot je: „*automated software program that perform tasks on a network with some degree of autonomy*“. (Microsoft Corporation 2010)

Tzv. bot-klienti (nazýváni také jako zombie) se shlukují do menších či větších sítí známých jako botnety. Botnet je tedy síť vzdáleně ovládaných počítačů, která podle knihy Botnets: The killer web app musí splňovat dvě podmínky: „*First, the clients in a botnet must be able to take action on the client without the hacker having to log into the client´s operating system (Windows, UNIX, or Mac OS). Second, many clients must be able to act in a coordinated fashion to accomplish a common goal with little or no intervention from the hacker. “* (SCHILLER, C. A., et al. 2007) Akce bot-klientů by měly být prováděny tak, aby majitel kompromitovaného systému nepoznal, že je infikovaný nebo, že je právě součástí nějakého botnetu.

2.2 Složení a topologie botnetů

Tak jako běžná počítačová síť má svoji strukturu, tak i u botnetu můžeme hovořit o různých prvcích této sítě. Nejběžnější botnet se skládá z C&C (Command and Control) serveru a jednoho či více bot-klientů.

V závislosti na velikosti se botnet dělí na menší celky. Pojmenování těchto celků je odvozeno z armádní terminologie. Odtud také vznikl další název pro botnet – armáda zombie. Tato armáda se člení na tzv. divize a každá tato divize je řízena vlastním „bot-serverem“. Bot-herder má tedy ve svém botnetu k dispozici několik „bot-serverů“. V případě, že je jeden komunikační kanál přerušen, ztrácí pouze jednu divizi botů. S ostatními divizemi botnetu může i nadále komunikovat a pokračovat tak v nekalé aktivitě. (SCHILLER, C. A., et al. 2007)

2.2.1 Command&Control a jeho topologie

Rozhodující pro komunikaci mezi bot-herderem a bot-klienty je použití nějakého C&C (Command & Control) mechanizmu. Ten je v současné době podle společnosti Microsoft stále nejvíce založen na IRC (Internet Relay Chat) protokolu. Jsou zde zastoupeny i další mechanizmy, jako HTTP, P2P, Instant Messaging a další. (viz 2.3 Typy botnetů). (Microsoft Corporation 2010) Podle práce Botnet Communication Topologies existují různé topologie C&C mechanizmů. Jednotlivé typy topologií zde budou stručně představeny.

Hvězdicová topologie – základem je jeden C&C server, jehož pomocí bot-herder komunikuje s klienty botnetu. Nevýhodou (pro bot-herdera) této topologie je, že pokud je tento C&C server nějakým způsobem neschopen vysílat příkazy, ztrácí kontrolu nad botnetem.

Multi-server topologie – Její základ tkví v přidání několika serverů do hvězdicové topologie, aby se zajistila redundance ovládání botnetu. Servery komunikují jak s klienty, tak i mezi sebou. To při výpadku jednoho serveru zaručí, že zbylé provozuschopné servery převezmou jeho úlohu.

Hierarchická topologie – V této topologii se někteří bot-klienti stávají určitými druhy proxy serverů, přes které C&C server zadává své příkazy dalším botům. U této topologie je velmi obtížné vypátrat jak celkovou velikost botnetu, tak i zdroj příkazů, avšak není vhodná pro „real-time operace“, jako jsou například DDoS útoky a to kvůli velké latenci na síti.

Náhodná topologie – Zde nevystupuje žádný C&C server. Příkazy jsou distribuovány vzájemně mezi bot-klienty. Tyto příkazy jsou označeny nějakým důkazem pravosti jako autoritativní. Příkazy jsou bot-herderem předány přímo jednomu klientovi, jehož prostřednictvím jsou dále distribuovány. Nevýhodou je opět velká latence, výhodou je velmi těžké vypátrání zdroje. (OLLMAN, G. 2009)

Pokud budeme brát v potaz předchozí zmínku o rozdělení jednoho botnetu do divizí, je vhodné uvažovat tyto topologie v rámci jednotlivých divizí a ne v rámci celého botnetu. Další zajímavostí, která by nahrávala útočníkům, by bylo, kdyby každá divize měla jinou topologii – zde se ale naskýtá otázka, zda by se už nejednalo o úplně jiný botnet.

2.3 Typy botnetů

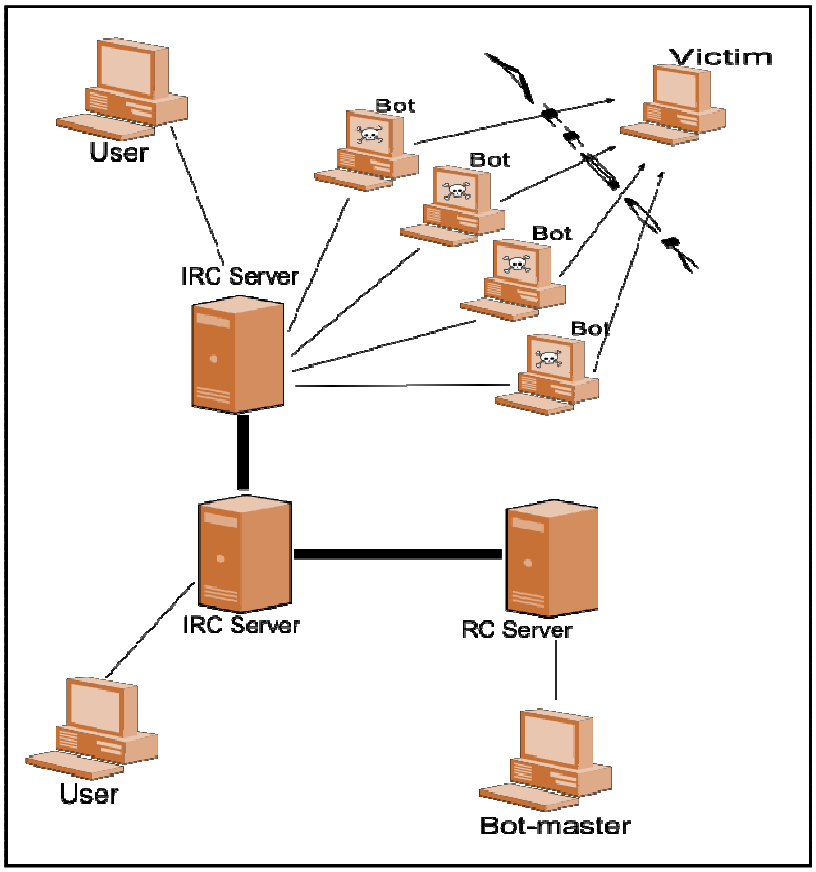
Existují různé typy botnetů, které se odlišují tím, jaká technologie nebo protokoly byly použity pro komunikaci mezi bot-masterem a botnetem. V době, kdy se botnety začínaly teprve objevovat, byl nejrozšířenějším typem IRC botnet, jehož komunikace byla založena na IRC protokolu. Jak doba pokročila, objevily se nové způsoby, jak s botnetem komunikovat. Do hry vstupují například Instant Messaging nebo sociální sítě.

2.3.1 IRC botnet

Internet Relay Chat protokol vznikl začátkem devadesátých let minulého století za účelem textové konference v reálném čase. IRC pracuje na principu klient-server (ale i server-server). Klienti (nebo i servery) se připojují na centrální server k určitému kanálu (channel). Úkolem tohoto centrálního serveru je doručování zpráv jednotlivým účastníkům kanálu nebo přeposílání všem připojeným ke kanálu. To záleží na tom, zda se jedná o komunikaci one-to-one, nebo one-to-many. (REED, D., OIKARINEN, J. 1993)

Právě tohoto principu využívá útočník, aby se spojil s bot-klienty. Bot je nakonfigurován tak, aby se po infiltraci do systému připojil k určitému IRC serveru na určitý kanál. Takovýto bot většinou zná adresu serveru, jméno kanálu a heslo pro vstup – IRC umožňuje skrýt jméno kanálu a vstoupit do něj pouze na základě znalosti hesla, což je prostředek využívaný ke skrytí útočníka. Existují ještě další prostředky, které útočník využívá ke svému ukrytí. Bot-klienti se připojují buď k veřejnému IRC serveru, nebo ke skrytému serveru, který obvykle běží na nějakém infikovaném stroji. Útočník se nemusí připojit přímo k serveru, ke kterému jsou připojeni bot-klienti, ale příkazy zadává z jiného serveru z IRC sítě (viz Obrázek 1). Dalším prostředkem ke skrytí útočníka může být šifrování příkazů a použití nejrůznějších anonymizérů nebo proxy serverů.

(COOKE, E., JAHANIAN, F., MCPHERSON, D. 2005; Microsoft Corporation 2010)



Obrázek 1 – Komunikace v botnetu pomocí IRC protokolu a několika IRC serverů (multi-server topologie).
Předloha: (COOKE, E., JAHANIAN, F., MCPHERSON, D. 2005)

2.3.2 HTTP botnet

Ačkoliv podle Microsoftu jsou ve 2. čtvrtletí 2010 nejvíce rozšířeny IRC botnety, botnety pracující nad protokolem HTTP se dostávají stále více do kurzu. (Microsoft Corporation 2010) HTTP je protokol aplikační vrstvy TCP/IP síťového modelu a je primárně určen pro výměnu dat ve formátu HTML – z toho vyplývá, že se jedná o hlavní protokol pro prohlížení webových stránek v internetu. Když se tento protokol použije k nekalé činnosti, jako je právě komunikace s botnetem, nebo jeho řízení, bude velmi těžké tuto komunikaci odhalit a zablokovat ji, protože ve většině případů je síťová komunikace tímto protokolem na standardních portech povolena.

Podle C. A. Schillera existují 2 typy HTTP botnetů: echo-based a command-based. V prvním případě (echo-based) se jedná o princip, kdy se bot připojí na web-server, na kterém buď najde informaci s příkazy, které má vykonat, anebo tím, že se připojil, o sobě pouze dává vědět, že je aktivní. Toho se poté využívá v druhém typu (command-based), kdy se botnet master připojuje na jednotlivé klienty a předává jim příkazy. V případě echo-based botnetu se bot připojí na server, aby o sobě dal vědět, což útočník obvykle zaznamenává pomocí logování. V log souboru potom najde IP adresy, které jsou pro něj aktivní a na které se může později připojit. Druhou metodou může být, že bot posílá na server celou URL, která obsahuje informace např. o portu, na kterém bot poslouchá, nebo třeba heslo pro přístup. Taková URL může vypadat například takto:

http://botnet1.example.com/blah.txt?port=34556password=qwerty211
(SCHILLER, C. A., et al. 2007).

U echo-based botnetů je navíc ještě možnost předání příkazů. Bot po připojení na server, na kterém je uložen soubor s informacemi, tento soubor stáhne a vykoná příkazy. Tyto příkazy nemusí být jenom provedení nějakého útoku, ale velice často znamenají aktualizaci (update) botnetu nebo doinstalaci nějakého dalšího malware na infikovaný stroj. Command-based botnety fungují obráceně. Bot-master se připojuje na jednotlivé bot-klienty za účelem předání příkazu. Tímto způsobem může útočník docílit například aktualizace botnetu, stažení určitých souborů (screenshoty či soubor s hesly, apod.), nahrání na klienta dalšího malware či modulů. Také může útočník nechat klienta vykonat specifické příkazy shellu. (SCHILLER, C. A., et al. 2007)

2.3.3 IM botnet

Instant messaging je služba, podobně jako IRC, určená k zasílání zpráv v reálném čase na internetu. IM je používáno, stejně jako u IRC, k řízení botnetu – bot-master komunikuje s botnetem přes IM účty – nebo také k šíření červů (botů). (SCHILLER, C. A., et al. 2007) Počítač tak může být infikován například pomocí sociálního inženýrství, kdy uživatel navštíví webové stránky se škodlivým kódem, nebo pomocí cross-site scripting (XSS).

2.3.4 P2P botnet

V oficiálních studijních materiálech ke kurzu Cisco Academy I je Peer-to-peer technologie vysvětlena takto: P2P síť je složená z dvou či více počítačů, které spolu sdílejí své zdroje (např. tiskárny, soubory). Pomocí P2P aplikací mohou klienti P2P sítě vystupovat jako klient i server. (Cisco Systems, Inc. 200?) Tento typ botnetu tedy nemá žádný C&C server, od kterého získává příkazy, ale příkazy jsou šířeny mezi klienty vzájemně. To samozřejmě znamená, že útočník musí nějakým způsobem předat úkoly jednomu botu, který je distribuuje dále. Tomuto typu odpovídá náhodná topologie botnetu. Podle knihy Botnets: The Killer WebApp mají P2P jednu velkou výhodu: botnety jsou decentralizované a to znamená, že nejsou závislé na jednom bodu (serveru), který může selhat, ale také nevýhodu: příkazy mohou být botnetu předány prostřednictvím jakéhokoliv bot-klienta, což může být zneužitelné v případě, že se ho zmocní někdo jiný. Řešením toho je například použití kryptografických klíčů. (SCHILLER, C. A., et al. 2007) Jak už bylo zmíněno dříve (viz 2.2.1 Command&Control a jeho topologie), tyto botnety je velmi těžké vypátrat a také určit jejich velikost, ale na druhou stranu jejich nevýhodou je dlouhá odezva mezi zadáním příkazu a jeho vykonáním celým botnetem.

2.3.5 Ostatní typy botnetů

Do této kategorie se dá zařadit **FTP** botnet, který není v dnešní době příliš rozšířen. Schiller zmiňuje tento způsob spíše jako způsob doručení odcizených údajů bot-masterovi než jako C&C mechanizmus v pravém slova smyslu. (SCHILLER, C. A., et al. 2007) Dále by se sem daly zařadit botnety, které přijímají příkazy pomocí protokolu **TCP/UDP**. Útočník se připojí na určitý port a protokolem TCP nebo UDP posílá klientům textová nebo binární data, která botnet dokáže interpretovat jako příkazy. (Microsoft Corporation 2010) A na konec bych sem zařadil **DNS** systém, i když se nejedná o způsob jak s botnetem komunikovat či ho řídit. DNS je používáno jako vrstva sloužící k zabezpečení C&C mechanizmu botnetu. (SCHILLER, C. A., et al. 2007) Pro bot-mastera je jednoduší vybavit každého bota doménovým jménem C&C serveru (ať už se jedná o IRC či webový server), které si od DNS nechají přeložit na IP adresu. To umožňuje bot-masterovi v případě výpadku nějakého serveru postavit jiný a pouze změnit IP adresu v DNS (který může například tajně běžet na jiném infikovaném stroji) a tím jednoduše přesměrovat bot-klienty na jiný C&C.

2.4 Jak se z počítače stane bot-klient

Existuje mnoho způsobů, jak může být systém infikován. Toto poměrně rozsáhlé téma zde zmíním ve zkrácené formě, aby byly vysvětleny hlavní principy infikování systému botem.

Níže uvedené metody by se daly shrnout pod pojem sociální inženýrství – vždy se totiž jedná o to, že útočník (ať už hacker nebo automat) nějakým způsobem přinutí, přesvědčí či naláká uživatele počítače, aby dovolil vykonat škodlivý kód. Tento škodlivý kód poté umožní útočníkovi např. vzdálený přístup do systému, otevře tzv. „zadní vrátka" (backdoor), která poté umožní instalaci nějakého botu, či nechá stáhnout nějaký další malware. Metody dovolující provedení škodlivého kódu (podle Botnets: The Killer WebApp): **phishingový útok –** Tento typ útoku je většinou spojen s lákáním hesel (nejčastěji k bankovním účtům) kdy je uživateli poslán e-mail s podvrženými webovými stránkami. Zde uživatel zadá své jméno a heslo, které se uloží na serveru hackera. Na pozadí se také většinou provede nějaký škodlivý kód, čímž se počítač infikuje**.** Tento typ útoků je většinou prováděn na cílovou skupinu lidí (nejčastěji na klienty jedné banky). **Nalákání na webové stránky –** Většinou na webových stránkách pochybného a nedůvěryhodného obsahu (např. warez-stránky, stránky s erotickou tematikou, apod.) lze kliknout na odkazy, které vedou na stránky obsahující červa nebo trojského koně. **Přílohy e-mailů –** Při otevření přílohy e-mailu například od neznámého odesílatele může také být nainstalován nějaký malware. **SPIM –** Nevyžádaná zpráva v IM službě (označováno jako SPIM – SPAM IN INSTANT MESSAGING) může obsahovat odkaz na webovou stránku, která pro uživatele může být lákavá. (SCHILLER, C. A., et al. 2007)

Zatímco první způsob byl založen na sociálním inženýrství, další způsob, jak infikovat systém, je založen na zneužívání neopravených chyb operačních systémů či aplikací. V jednom případě je třeba oskenovat systém na otevřené porty – tím pádem nalézt běžící služby a podle typu a verzí těchto služeb najít chyby, které se dají zneužít. Tyto funkce mají většinou naimplementovány bot-klienti – vše probíhá automaticky. V jiném případě se využívá chyb v aplikacích, jako jsou například webové prohlížeče či prohlížeče PDF dokumentů. Tyto chyby umožní útočníkovi spustit libovolný kód jenom tím, že uživatel systému zobrazí webovou stránku či PDF dokument, který obsahuje malware. (SCHILLER, C. A., et al. 2007)

Další způsob infekce počítače je založen na využití „zadních vrátek", které v systému zanechal trojský kůň či červ, jímž byl systém již dříve infikován. Rejstřík pojmů společnosti ESET definuje pojem „backdoor" takto: „*Jde o aplikace typu klient-server, které jsou schopnostmi velice podobné komerčním produktům jako pcAnyWhere, VNC či Remote Administrator. Narozdíl od nich ovšem vystupují anonymně, uživatel není schopen jejich přítomnost běžným způsobem vypozorovat a to je důvodem, proč jsou preventivně detekovány antiviry jako jeden z typů infiltrace. Mluvíme o neautorizovaném vstupu.*" (ESET, spol. s r.o. 200?)

Výše zmiňované komerční produkty slouží pro vzdálenou správu či kontrolu počítače přes internet. Funkci hledání zadních vrátek v systémech zase většinou implementují jednotliví bot-klienti již existujícího botnetu, takže vše probíhá zcela automaticky.

Dalším způsobem jak infikovat počítač je hádání hesel či „brute-force attack“ (útok hrubou silou) většinou na služby NetBIOS. Služba NetBIOS slouží například ke sdílení souborů v síti. Pro tyto služby je většinou nutné zadat login (přihlašovací jméno a heslo), ty jsou ale ve většině případů ponechány „implicitní“ (výchozí). Červ (či bot-klient) má od hackera většinou k dispozici nějaký seznam hesel a slovník přihlašovacích jmen, pomocí kterého se zkouší připojit na tyto služby potencionální oběti. (SCHILLER, C. A., et al. 2007)

2.5 Komunikace s bot-masterem

Správné pochopení toho, jak probíhá komunikace mezi majitelem botnetu a botnetem, je důležité z hlediska detekování aktivity červa v našem systému či prevence infekce našeho počítače. Struktura této části je zpracována dle knihy Botnets: The Killer WebApp. (SCHILLER, C. A., et al. 2007)

2.5.1 První připojení k C&C

V předchozí části bylo popsáno, jak se z počítače stane tzv. „zombie“. Po tomto kroku následují další akce, které bot provádí. Prvním krokem je kontaktování bot-herdera a informování o aktivitě bot-klienta. Schiller toto nazývá termínem *Rallying Botnet Client*. „*Rallying is the term given for the first time a botnet client logins in to a C&C server.*“ (SCHILLER, C. A., et al. 2007) Bot připojující se k IRC serveru může hned po připojení získat nějaký příkaz přečtením témata kanálu, ke kterému se připojuje. Téma kanálu může vypadat dle The Honeynet project například takto:

".http.update http://<server>/~mugenxu/rBot.exe c:\msy32awds.exe 1"

Jedná se o aktualizaci bot-klienta. (The Honeynet project 2008)

2.5.2 Zabezpečení bot-klienta

Po prvním přihlášení k C&C je další akcí, kterou bot provádí, zabezpečení sebe sama proti antivirovým (A/V) nástrojům. Bot nejčastěji stáhne anti-A/V modul, který buď zcela vypne A/V nástroj, nebo se před ním zamaskuje. Vypnutí nebo odstranění A/V aplikace by bylo dosti nápadné, proto bot-klienti používají sofistikovanější metody. Používají například knihovny, které zajistí, že „bot-soubory“ nebudou skenovány, nebo že se nebudou provádět aktualizace (respektive aktualizace skončí nějakou chybou typu „Server nenalezen“).

2.5.3 Čekání na příkazy

V návaznosti na 3.5.1 První připojení k C&C The Honeynet Project dále uvádí: „*And if the topic does not contain any instructions for the bot, then it does nothing but idling in the channel, awaiting commands.*“ (The Honeynet project 2008) V příloze jsou uvedeny příklady příkazů, které může útočník svému botnetu předávat. Příkazy se různí podle typů botů a hlavě podle typu celého botnetu. Základní princip je ovšem stejný u všech. Typy příkazů jsou úzce spjaty s tím, k čemu se botnet využívá.

2.6 Využití botnetů

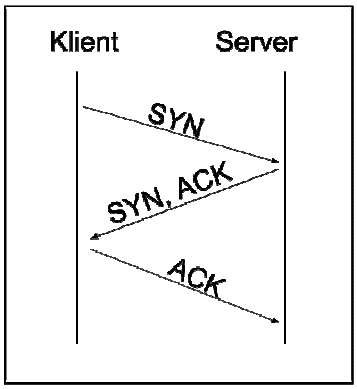
V této části se dostáváme k tomu proč vlastně hackeři věnují tolik času vytváření botnetů. V dnešní době jsou pro ně hlavním motivem peníze. Ty se dají „vydělat“ nejen provozováním nějakého botnetu, ale i např. pronájmem botnetu na černém trhu. Tato část je zpracována podle knihy Botnets: The Killer WebApp (SCHILLER, C. A., et al. 2007), kde je tato problematika podle mého názoru zpracována nejpodrobněji. Následuje přehled jednotlivých možností využití botnetů.

2.6.1 Infikování dalších počítačů

Jak již bylo zmíněno v 2.4 Jak se z počítače stane bot-klient, systém může být infikován různými způsoby. Velká část útoků je konána automaticky pomocí botů k tomu uzpůsobeným. Tito bot-klienti obsahují většinou skenovací modul, jehož pomocí hledají otevřené porty a služby na ostatních počítačích, kterých se pak snaží využít. Další vstupní branou do počítače mohou být již zmíněná „zadní vrátka“, pomocí nichž se bot-klient vzdáleně připojí k počítači a nainstaluje malware. Tito bot-klienti jsou většinou „specializovaní“ - vyhledávají určitou službu, což je mnohem hůře odhalitelné, než kdyby jeden bot skenoval celý systém.

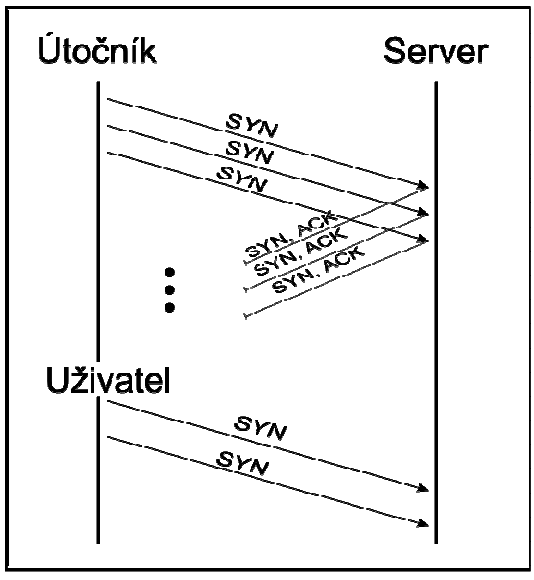
2.6.2 DDoS útoky

Distribuované odepření služby (Distributed Denial of Service) je typ útoku, kdy útočník nechá zahltit například webový server, který se tím pádem stane pro normální uživatele nedostupným. Zahlcení může bot-herder provést například TCP SYN flood útokem, který je založen na principu „trojitého podání ruky“ při navazování komunikace TCP protokolem. Viz Obrázek 3.



Obrázek 2 - Navázání spojení protokolem TCP.

Na Obrázku 3 je znázorněn TCP SYN flood útok. Server je zahlcen SYN segmenty, na které odpovídá SYN+ACK segmenty, ty ale buď dorazí na jiné místo, kde jsou zahozeny, nebo na ně odpovězeno není. V tento okamžik nastává situace, kdy server čeká na odpověď, proto další pokusy o spojení nepřijímá, tím pádem se stává nedostupným.



Obrázek 3 – Útok typu SYN Flood.

2.6.3 Vydírání

Vydírání je jedna z dalších činností, kterou útočníci provozují ke svému obohacení. Může probíhat například tak, že útočník provede DDoS útok na webový server, který neuvolní, dokud majitel nezaplatí výkupné. Dalším způsobem může být zašifrování důležitých dat na počítači, který je bot-klientem. Legální majitel počítače se tak k těmto datům nedostane, což je důvod k vydírání.

2.6.4 Odcizení osobních dat

V tomto případě se jedná například o přihlašovací jména a přístupová hesla k různým službám (nejčastěji k bankovním aplikacím) nebo o sériové klíče k zakoupenému software. To může být provedeno tak, že malware nainstalovaný na zombie počítači vyhledává soubory, které tyto informace obsahují, nebo malware vystupuje jako keylogger, který zaznamenává stisky kláves a takto nasbíraná data odesílá do úložiště.

2.6.5 Ostatní

Mezi další neméně závažné činnosti, které bot-masteři provozují patří: **rozesílání spamu** a s tím spojené **phishingové útoky**, falešné generování klikání na tzv. „**Clicks4Hire**“ reklamy (za každé kliknutí na reklamu dostane majitel webu, na kterém je reklama umístěna, finanční odměnu od majitele reklamy), **instalace adware** a jiného **špionážního malware** pro marketingové účely a ukládání či distribuce ukradených nebo ilegálně získaných filmů, hudby a software.

3 Možnosti detekce

Detekce malware je jedním ze dvou základních prvků obrany. Druhým základním prvkem je prevence před infikováním. Podle Shadowserver foundation se strategie detekce dají rozdělit na host-based detekci a network-based detekci. (The Shadowserver Foundation 2005) V následující kapitole budou představeny jednotlivé možnosti obou strategií.

3.1 Host-based detekce

3.1.1 Antivirové nástroje

Jednou z možností, jak detekovat červa či jiný malware je používání antivirového nástroje. Tyto nástroje jsou zároveň brány jako základní stupeň ochrany. Tato detekce (či ochrana) je založena na dvou fázích:

První – skenování souborů a porovnávání signatur z virové databáze – pomáhá detekovat pokusy o infekci již známým malware

Druhá – detekce na základě heuristických analýz – detekuje nechtěné nebo nestandardní chování souborů a odhaluje tak zatím ještě neznámý malware

Záměrně zde uvádím širší pojem malware místo užšího virus, protože A/V aplikace je schopna detekovat i jiné druhy malware, jako např. backdoor, červ či trojský kůň, pomocí nichž se z počítače může stát „zombie“.

A/V ochranu však nelze považovat za 100% efektivní, jelikož musíme brát v potaz zabezpečení bot-klienta (viz 3.5.2 Zabezpečení bot-klienta) a skutečnost, že botnet je často aktualizován, tzn. že malware je na bot-klientovi často měněn, což může být pro A/V nástroj problém. Je tedy dobré považovat A/V nástroj za základní způsob detekce, který je vhodné doplnit dalšími metodami, viz níže.

Jedním způsobem, jak alespoň částečně obejít zabezpečení malware proti detekci A/V programem nainstalovaným v počítači je možnost bootovat do nějakého jiného (nejlépe read-only) operačního systému, který obsahuje A/V nástroje schopné detekovat malware. Nejlepším příkladem těchto nástrojů jsou Live CD založené na linuxových distribucích. Některé antivirové společnosti nabízejí tyto nástroje volně ke stažení. Jedná se o např. Avira Rescue System, Kaspersky Rescue Disk či F-secure rescue CD.

Při výběru z velké škály A/V programů by mohly pomoci webové stránky neziskové organizace AV-Comparatives[1], která provádí nezávislé testy produktů různých tvůrců A/V aplikací. Zde se jedná o komerční řešení, avšak některé společnosti nabízejí také volně použitelné varianty svých produktů (např. Avira, AVG, Avast či Microsoft).

Z open-source A/V nástrojů lze v současnosti doporučit ClamAV pro platformu UNIX, existují také varianty pro Windows (ClamAV for Windows či ClamWin[2]).

3.1.2 Host-based IDS (HIDS)

Další možností detekce škodlivých aktivit na úrovni jednotlivých strojů je použití systému pro detekci průniku. V tomto případě z kategorie host-based. Jakub Mauric ve své práci definoval intrusion detection system (IDS) takto: „*systém detekce průniků slouží k detekci neoprávněných průniků do systému …*“ (MAURIC, J. 2009).

A podle A. Shaikha rozeznáváme tři základní typy IDS: host-based, network-based a distributed IDS (SHAIKH, A. 2010). NIDS (Network-based IDS) budou popsány v Network-based strategii detekce.

IDS, ať už HIDS či NIDS, používají dvě hlavní detekční metody (zpracováno podle prací A. Shaika aj. Maurice:

Detekce signatur – Systém má v databázi uloženy signatury průniků a na základě znalosti těchto formátů detekuje průnik. Formát zápisu těchto signatur je závislý na implementačním řešení jednotlivých IDS. Tato metoda je schopná detekovat již známé průniky, ale je neúčinná v detekci tzv. zero-day útoků.

Detekce anomálií – IDS vytvoří model či profil normálního fungování systému (v tzv. režimu učení) – ať už jednoho počítače či celé sítě. Jakékoliv odchylky v chování systému od vytvořeného profilu poté IDS označí jako detekci anomálie. V takovém případě systém uvědomí zodpovědnou osobu (např. záznamem do logu či e-mailem). Tato metoda je celkem úspěšná v detekci nových průniků, je zde však riziko velké míry falešných poplachů. (SHAIKH, A. 2010; MAURIC, J. 2009)

Podle práce A. Shaika Botnet Analysis and Detection System HIDS monitoruje různé aktivity v hostitelském systému. Z toho vyplývá funkcionalita HIDS: kontrola integrity souborů, monitoring běhu aplikací, analýza systémových či aplikačních logů a monitoring změn v registrech (v případě OS Windows). (SHAIKH, A. 2010) Velkou výhodou některých HIDS je také centralizovaná architektura, což znamená, že existuje jeden server na kterém je

[1] http://www.av-comparatives.org/

[2] www.clamwin.com

nainstalovaný tzv. manager, který obsahuje databázi a sbírá a vyhodnocuje data od jednotlivých stanic, na kterých jsou nainstalovaní tzv. agenti. V současnosti existují některá volně šiřitelná řešení HIDS:

OSSEC HIDS - jedná se o multiplatformní HIDS. Má centralizovanou architekturu, avšak manager systému je implementován pouze pro UNIX. OSSEC Agent lze nainstalovat na UNIX, Windows, MacOS a další systémy. Více informací na oficiálních stránkách OSSEC[3].

Samhain HIDS - modulární centralizovaný HIDS tentokráte pro operační systémy UNIX a Cygwin/Windows. Na oficiálních stránkách[4] je také návod, jak nainstalovat prostředí Cygwin a samotný HIDS do OS Windows.

Aide – nástroj pro kontrolu integrity souborů. Vytvoří se databáze kontrolních součtů souborů. Při kontrole Aide vytváří hash skenovaných souborů a porovnává je s databází. V konfiguračním souboru lze nastavit, které soubory se budou kontrolovat.

Na tomto místě považuji za vhodné zmínit **Tripwire**, což je také nástroj pro kontrolu integrity. Tento projekt se v roce 1999 rozdělil na dvě větve. Jedna se ubírala komerční cestou, druhá zase open-source. Nyní je k dispozici open-source verze nástroje pouze pro operační systémy typu UNIX s decentralizovanou architekturou.

3.1.3 Sledování chování počítače

Tato metoda není obecně považována za spolehlivou a účinnou, ale myslím, že do detekce také patří. Rozhodně by měla být brána jako doplňková ke všem ostatním. V níže uvedený případech se nemusí vždy jednat o infekci škodlivým kódem, je tedy třeba dalších postupů, které identifikují činnost počítače. Výsledky této metody spíše naznačí, že by se strojem mohlo být něco v nepořádku.

Výrazné zpomalení počítače – Pokud se jinak celkem výkonný počítač náhle výrazně zpomaluje, může to být způsobeno aktivitou nějakého malware. Je však důležité zjistit, zda systém na pozadí nevykonává nějaké rutinní operace, jako automatické aktualizace či nejrůznější kontroly např. disku apod.

Síťová aktivita – Pokud je zaznamenána komunikace po síti a přitom nejsou spuštěny žádné uživatelské síťové aplikace, bylo by vhodné zjistit, co inicializuje spojení. Mohlo by se jednat např. o červa či bot komunikující s C&C serverem. Je tedy vhodné sledovat komunikaci na standardním portu IRC serveru 6667 – za použití příkazu *netstat –a* v příkazovém řádku.

Pop-up okna – Dle The Shadowserver foundation náhodné vyskakování těchto oken může znamenat nákazu adware, ale také to může znamenat začlenění do primitivního botnetu zaměřeného na podvodné klikání na reklamy, tzv. „Clics4Hire“ (viz 3.6 Využití botnetů) (The Shadowserver Foundation 2005).

[3] www.ossec.net

[4] www.la-samhna.de/samhain/index.html

3.2 Network-based detekce

3.2.1 SNMP

Schiller zmiňuje další metodu detekce botnetů na síti pomocí SNMP. Simple Network Management Protocol je protokol pro správu sítí. Pomocí tohoto protokolu lze odhalit některé aktivity spojené s botnety jako DDoS útoky, rozesílání spamu či skenování sítě, …

Zde uvedu volně šiřitelný nástroj Cricket pracující se SNMP protokolem. Cricket každých pět minut posílá SNMP žádost směrovačům a přepínačům a data ukládá. RRDTOOL[5] (pomocný nástroj pro Cricket) poté data analyzuje a na jejich základě vykresluje graf. Z tohoto grafu lze vyčíst abnormální zatížení směrovače či abnormální aktivitu klienta na síti (např. podle vytížení portu na přepínači), které mohou znamenat jednu z výše uvedených aktivit. (SCHILLER, C. A., et al. 2007)

SNMP (respektive nástroje pracující se SNMP) nerozezná o jakou aktivitu se jedná, avšak podle grafu lze škodlivou aktivitu zaznamenat. Záleží potom na dalším zkoumání a monitorování, aby byla aktivita správně klasifikována.

3.2.2 NetFlow

Dalším způsobem jak monitorovat aktivitu červů či botnetů na síti je technologie NetFlow. Marián Krčmárik ve své diplomové práci definoval NetFlow takto: *„NetFlow technológia je založená na generovaní štatistických dát o jednotlivých dátových tokoch prúdiacich cez aktívny prvok siete (väčšinou sa jedná o CISCO routre) a ich odosielanie na kolektor, kde dochádza k zberu, analyzovaniu a vyhodnoteniu NetFlow dát“* (KRČMÁRIK, M. 2009). Takovýmto kolektorem podle Schillera může být buď samotný aktivní prvek, jako je směrovač či přepínač (nejčastěji od společnosti Cisco), což je ale komerční řešení, nebo UNIXová stanice s volně šiřitelnými nástroji k tomu určenými (SCHILLER, C. A., et al. 2007). Těmito nástroji mohou být například Flow-tools či nfdump a jeho grafický front-end NfSen.

NetFlow data obsahují různé informace o toku dat. Schiller uvádí tyto: zdrojová a cílová IP adresa, zdrojový a cílový port protokolu TCP a UDP, označení IP protokolu, kontrolní TCP značky (např. SYN či FIN), množství dat a další (SCHILLER, C. A., et al. 2007). Pomocí těchto informací se dají zjistit aktivity spojené s botnety či s červy – skenování portů na síti, připojování stanic k C&C serveru, DDoS útok apod. Na rozdíl od detekce pomocí nástrojů pracujících se SNMP lze tyto aktivity již přesněji určit a dokonce specifikovat. Detekcí botnetů pomocí NetFlow se podrobně zabývá Marián Krčmárik ve své diplomové práci Analýza možnosti detekcie botnet sietí pomocou NetFlow dát (KRČMÁRIK, M. 2009).

[5] oss.oetiker.ch/rrdtool/

3.2.3 Firewall

Dobře nakonfigurovaný firewall je hned vedle antivirového nástroje považován za základní stupeň ochrany. Z teoretického hlediska by mohl být brán spíše jako preventivní opatření před nežádoucí aktivitou na síti, může však být vhodně využit i k detekci.

Prevence – pomocí firewallu lze filtrovat příchozí i odchozí komunikaci ze sítě (popř. z jedné stanice pokud se jedná o personální firewall). Je tedy vhodné uvažovat o tom, které síťové aplikace a služby v rámci sítě používáme či nabízíme. Ostatní služby můžeme zablokovat, nebo k nim omezit přístup (filtrování na základě IP adresy). Schiller zmiňuje porty: 135 – 139, 445 (Microsoft File Share služby), 1433/TCP, 1434/UDP (Microsoft SQL server). Tyto porty je vhodné zablokovat z důvodu mnoha zranitelností těchto služeb. (SCHILLER, C. A., et al. 2007)

Detekce – Aby firewall mohl být použit jako nástroj pro detekci škodlivé aktivity, musí být povoleno logování aktivit. Lze povolit logování blokovaných či úspěšných spojeních. Z takovýchto logů lze potom poznat jisté aktivity spojené s botnety. Např. když stanice často odesílá e-mailové zprávy (napojení na mail-server na port 25) – tak často, jak člověk nedokáže odesílat e-maily – může to znamenat, že se jedná o začlenění do spam botnetu apod. (SCHILLER, C. A., et al. 2007)

Avšak prohlížení většinou značně obsáhlých log souborů a hledání v nich nějaké nekalé aktivity je poměrně náročná práce, proto existují nástroje pro analýzu logů. Jedním takovým nástrojem je volně šiřitelný psad (Port Scan Attack Detektor) pro také volně šiřitelný firewall iptables.

3.2.4 Network-based IDS (NIDS)

Zde bych chtěl citovat definici použitou v práci Alana Shaikha Botnet Analysis and Detection System: „*Network IDSs (NIDS) monitor and analyse network traffic by reading all network packets and alerting of any suspicious activity*.“ (SHAIKH, A. 2010) Z této definice vyplývá, že se jedná o zařízení vhodně umístěné v síti tak, aby bylo schopno zachytávat veškerý provoz na síti. Rozeznáváme dva základní druhy sběru dat v přepínaných sítích:

Port mirroring – nastavení jednoho portu na přepínači tak, aby na něj byla posílána kopie veškerého provozu. Síťová karta IDS musí být v promiskuitním módu.

Větvení sítě – tzv. network-tap, hardwarové zařízení, které umožňuje monitoring sítě.

Výhodou je centralizovaná architektura, kdy agenti v různých segmentech sítě zasílají „nachytaná“ data na centrální server (manager), který data vyhodnocuje a generuje výstrahy. IDS mohou mít různé architektury:

Jednovrstevná – existuje pouze jedna sonda (agent) a zároveň manager, který data sbírá a rovnou vyhodnocuje.

Vícevrstevná – již zmíněná centralizovaná architektura.

Jelikož se jedná o monitoring sítě, měla by být tato metoda dostatečně efektivní pro detekci činnosti červa či botnetů, jejichž hlavní činností je právě aktivita po síti (odesílání citlivých dat, sken portů či DDoS útoky, apod.). Lze zde také například sledovat provoz na portech typických pro IRC komunikaci (6667) či služby instant messaging (5190/TCP – ICQ, 5222/TCP – Jabber, …). Pokud se jedná např. o HTTP botnet, jehož komunikace bude převážně na portu 80, škodlivá aktivita se bude odhalovat těžko, proto je třeba zaměřit se na další charakteristiky spojení apod.

Jak již bylo zmíněno (viz 4.1.2 Host-based IDS), detekce probíhá na základě signatur nebo na základě vytvořeného profilu sítě (v případě NIDS). Příkladem open-source „signaturového“ NIDS je známý Snort. V příloze je obrázek jednoduché síťové topologie s nasazeným Snortem.

3.2.5 Honeypot

Honeypot je takový systém, který emuluje, či obsahuje známé zranitelnosti, na které láká škodlivou aktivitu útočníků či červů. Tuto aktivitu poté zaznamená k pozdějšímu zkoumání a analyzování. Honeypoty se mimo jiné podle M. H. Ligha dělí na low-interaction a high-interaction. (LIGH, M. H. 2011)

Low-interaction – pro vysvětlení pojmu low-interaction honeypotu bych použil definici fungování nízko-interaktivního honeypotu Dionaea z oficiálních webových stránek: „*dionaea intention is to trap malware exploiting vulnerabilities exposed by services offerd to a network, the ultimate goal is gaining a copy of the malware.“* (Carnivore 200?) Tyto systémy emulují zranitelnosti síťových služeb k nalákání malware. Systému, který ukládá kopie malware pro další zkoumání, se někdy také říká malware collector. Takto „nachytaný“ malware se může poté dále analyzovat a to pomocí buď debuggerů, nebo on-line služeb poskytovaných různými společnostmi, které mají automatizované analytické nástroje. Jedná se například o CWSandbox[6] či SandBox společnosti Norman[7].

Výhodou low-interaction honepotů je jednodušší konfigurace a nasazení do sítě, nevýhodou je zase nízká míra interakce s útočníkem a tím i horší „stopování“. Z těchto důvodů se podle The Honeynet project ve větší míře nasazují pro „obranné“ či detekční účely. Oproti tomu se high-interaction nasazují více pro účely vědecké či výzkumné. Ovšem není to pravidlo. (The Honeynet project 2003)

High-interaction – tyto honeypoty žádné zranitelnosti nesimulují. Zde se jedná o reálné systémy, které zranitelné skutečně jsou. Výhodou oproti low-interaction je, že nabízejí zranitelný celý operační systém a tím vysokou míru interakce s útočníkem a také lepší stopování útočníka. Nevýhodou však je náročnost na konfiguraci, celkové nasazení, monitoring a analýzu útoků. Může se jednat o fyzické nebo virtuální stroje. Většinou se tyto honeypoty spojují do tzv. honeynets (viz 4.2.6 Honeynet).

Tím, že budeme mít v síti nasazený honeypot, můžeme detekovat činnost červů jak z internetu, tak i z vnitřní sítě. Detekcí červů z internetu můžeme dosáhnout např. obohacení virových databází či databází signatur pro IDS. Detekcí z vnitřní sítě se na druhou stranu dozvíme o infikovaných strojích. Pokud honeypot zaznamená aktivitu červa ve vnitřní síti, je to impuls pro zodpovědnou osobu k odstranění malware ze stanice, z které byl vedený útok.

Volně šiřitelných honeypotů je několik. Já zde uvedu tři příklady – Nepenthes[8], Dionaea a Mwcollectd. Použitelné jsou z nich jenom dva - Nepenthes již není dále podporován, místo něj nastoupil honeypot jménem Dionaea. Honeypotem Dionaea se zabývám podrobněji v další kapitole (viz 5.2 Honeypot Dionaea).

---

[6] www.sunbeltsecurity.com/sandbox/

[7] www.norman.com/security_center/security_tools/

[8] nepenthes.carnivore.it/

3.2.6 Honeynet

Tato část je zpracována podle „seriálu“ Know your enemy cyklus Honeynets. Honeynet je sám o sobě vysoce interaktivním honeypotem a skládá se také z dalších honeypotů (nejlépe také typu high-interaction). Největší silou honeynetu je, že v něm lze provozovat jakékoliv systémy – může zde být implementován například linuxový DNS server, „windowsovský“ web server, různé databáze pod různými operačními systémy apod. Klíčovým prvkem topologie honeynetu je tzv. honeywall. Jedná se o určitý druh brány, přes kterou prochází celý provoz, který je dále vyhodnocován a analyzován. Tento prvek je tedy klíčový jak pro výzkumnou, tak pro bezpečnostní stránku věci, protože brání tomu, aby byl celý honeynet zneužit k dalším útokům. The Honeynet project nabízí Honeywall CDROM, což je „balík“ open-source nástrojů k vytvoření honeywallu. Jedná se o bootavelné CD, pomocí kterého se jednotlivé prvky honeywallu nainstalují. (The Honeynet project 2006)

Jak již bylo zmíněno výše, honeynet je spíše vhodný pro výzkumné účely, než pro nasazení ve firmě jako detekční nástroj.

3.3 Ostatní možnosti

Níže uvedené možnosti jsou uvedeny jako „Ostatní“, protože se může jednat buď o host-based, nebo network-based detekci, nebo se může jednat o obě varianty zároveň.

Sledování IRC provozu – například sledováním komunikace na běžných IRC portech (např. 6667) se může odhalit botnet, avšak bot-masteři většinou volí jiné porty pro komunikaci, což může být v případě, že se nebude jednat o některá běžná čísla portů, podezřelé. Dále by se dala takováto činnost detekovat na základě chování připojeného počítače – po připojení na server je počítač v nečinnosti a pak velmi rychle odpovídá. (COOKE, E., JAHANIAN, F., MCPHERSON, D. 2005) Se sledováním IRC provozu by mohl pomoci nástroj Ourmon.

Detekce na základě DNS provozu – Některé botnety také využívají DNS serveru k připojení k C&C (viz 3.3.5 Ostatní typy botnetů). Pokud počítač z naší sítě odešle dotaz DNS serveru, mělo by se zkontrolovat, jestli se opravdu jedná o nějaký legitimní DNS server. Jako příklad nástroje zde uvedu Ourmon a jeho modul Topn DNS.

Forenzní techniky – sem patří zpětné zkoumání log souborů. Může se jednat o log soubory antivirových programů, firewallů či dokonce operačních systémů (event log). Procházení těchto logů a hledání konkrétních činností je zdlouhavá a náročná práce, proto existují různé nástroje pro analýzu těchto logů.

Hybrid IDS – hybridní IDS (také nazývaný distributed-based IDS) spadají jak do kategorie host-based tak i do network-based. Jedná se o to, že na jednotlivých počítačích je implementován host-based IDS (jako např. AIDE) a v celé síti je nasazen network-based IDS (např. Snort) (SHAIKH, A. 2010).

Ourmon – „mocný“ open-source nástroj pro monitorování sítě a detekci anomálií je podrobně popsán v knize Botnets: The Killer WebApp. Jedná se o nástroj, který pracuje na základě zachytávání paketů, které je řešeno pomocí knihovny libpcap.

Ourmon může pracovat ve dvou typech nastavení. Prvním typem nastavení je síťové zachytávání. V tomto případě musí být Ourmon umístěn v síti tak, aby měl přístup ke všem paketům, což může být realizováno pomocí mirror portu na switchi či network tapu. Druhým typem je zachytávání přímo na stanici (host-based). Takto nastavený Ourmon může být nasazen například na serverech s důležitými daty.

Celý systém je řešen jako dva prvky. Prvním prvkem je tzv. sonda, která zachytává pakety a následně je analyzuje a zpracovává z nich různé statistiky. Druhým prvkem je tzv. grafický engine (back-end graphics engine), který zpracovává data ze sondy, vytváří z nich grafy (pomocí nástroje RDDTOOL), ASCII zprávy a log soubory. Oba prvky mohou být implementovány na jedné stanici.

Ourmon lze využít např. pro detekci DDoS útoků (podobně jako u SNMP), skenování portů či IRC komunikaci. Detekce IRC komunikace neprobíhá pouze na základě IRC portů, ale hlavně na typu dat v paketech. Pokud data obsahují známé IRC zprávy, jako jsou JOIN, PING, PONG, PRIVMSG, atd., jsou vyhodnocena jako IRC komunikace (SCHILLER, C. A., et al. 2007).

3.4 Shrnutí

Jako nejlepší volbou v síťové detekci se jeví nasazení low-interaction honeypotu, který generuje menší množství dat a téměř vždy se jedná o nějakou škodlivou aktivitu oproti datům z NIDS. Honeynet (popř. high-interaction honeypot) je vhodný pro výzkumné účely, méně vhodný je pro detekci červů pro „obranné“ účely např. ve firmě. Také je mnohem složitější a náročnější ho správně nakonfigurovat.

Při nasazení IDS je dobré uvažovat o obou možnostech – HIDS a NIDS – tedy o hybridním typu IDS. Firewall a antivirový program jsou považovány za základní prvky. Ourmon, Netflow a SNMP, forenzní techniky jsou zajímavé doplňující varianty. Možností detekce je mnoho a vybrat tu správnou není jednoduché. Rozhodně je za dobrou volbu považována kombinace několika těchto možností.

3.5 Přehled možností detekce

U jednotlivých možností jsou uvedeny příklady nástrojů, pomocí nichž lze provádět detekci. Tyto nástroje jsou buď free nebo open-source. Nejedná se o vyčerpávající přehled. U každého nástroje je uveden operační systém a adresa webových stránek zabývajících se nástrojem.

1. Host-based detekce
    a. Antivirové nástroje
        i. Avira AntiVir Personal - Free Antivirus – Windows (http://www.avira.com/en/avira-free-antivirus)
        ii. ClamAV – UNIX (www.clamav.net)
        iii. Avira Rescue System – bootable (www.avira.com/en/support-download-avira-antivir-rescue-system
        iv. Kaspersky Rescue Disk – bootable (support.kaspersky.com/viruses/rescuedisk)
    b. Host-based IDS
        i. OSSEC HIDS – UNIX, Windows, MacOS (www.ossec.net)
        ii. Samhain HIDS – UNIX, Cygwin/Windows (www.la-samhna.de/samhain/)
        iii. Aide – UNIX (aide.sourceforge.net)

c. Sledování chování počítače
    i. Netstat – UNIX, Windows

2. Network-based detekce
    a. SNMP protokol
        i. Cricket – UNIX (cricket.sourceforge.net/)
    b. NetFlow nástroje
        i. Nfdump + NfSen – UNIX (nfdump.sourceforge.net)
        ii. Flow-tools – UNIX (www.splintered.net/sw/flow-tools/)
    c. Firewall
        i. Pf – Open BSD (www.openbsd.org/faq/pf)
        ii. Iptables – UNIX (www.netfilter.org)
    d. Network-based IDS
        i. Snort – UNIX, Windows (www.snort.org)
        ii. Bro IDS – UNIX (bro-ids.org)
    e. Honeypot
        i. Dionaea – UNIX (dionaea.carnivore.it)
        ii. Mwcollectd – UNIX (code.mwcollect.org)
    f. Honeynet
        i. Honeywall CDROM – bootable (projects.honeynet.org/honeywall/)
    g. Ostatní
        i. IRC provoz viz Network-based IDS, NetFlow nástroje, firewall
        ii. DNS provoz viz Network-based IDS, NetFlow nástroje, firewall
        iii. Forenzní techniky
            - Psad – UNIX + iptables (www.cipherdyne.org/psad/)
        iv. Ourmon – Linux, FreeBSD (ourmon.sourceforge.net/)

4 Praktická aplikace honeypotu

Metoda detekce pomocí honeypotu je celosvětově rozšířená a představuje poměrně spolehlivou metodu pro detekování škodlivého kódu – ať už jde o známý či tzv. „zero-day“ malware, proto jsem se rozhodl tuto metodu prakticky vyzkoušet nasazením low-interaction honeypotu do univerzitní sítě.

Nejdříve bylo nutné vybrat honeypot. Velmi známý Nepenthes již není dále podporován, protože byl vyvinut jeho „následovník“ Dionaea. Tento honeypot byl v mnoha ohledech vylepšen (viz 5.2.1.1 Změny oproti Nepenthesu), proto jsem zvolil právě Dionaeu. Dalšími kritérii pro výběr vhodného software bylo to, zda je stále podporován (možnosti aktualizací) a zda existuje nějaká podpora pro řešení problémů.

Dále bylo nutné vybrat operační systém. Dionaeu lze zkompilovat pouze pod operačním systémem UNIX. Zvolil jsem tedy stabilní verzi linuxové distribuce Debian GNU/Linux ve verzi 5.0 (později proveden upgrade na verzi 6.0). Důvodem pro zvolení Debianu byly mé předchozí zkušenosti s touto distribucí a výborná podpora od vývojářů honeypotu právě pro distribuce Debian a Ubuntu a samozřejmě stabilita operačního systému.

Nakonec bylo třeba získat stroj, na který vše nainstaluji a připravit pro něj vhodnou konfiguraci sítě. Využil jsem jeden z počítačů, které byly vyměněny z poslucháren pavilonu Veselovského Přírodovědecké fakulty. Umístění počítače v síti muselo být takové, aby na něj nebyl filtrován žádný provoz a samozřejmostí byla nutnost veřejné IP adresy. Nastavení sítě poskytl správce sítě PřF.

4.1 Instalace serveru

Nejprve bylo nabootováno z předem připraveného flash disku s .iso obrazem Debianu. Jednalo se o obraz „netinstall“, jehož pomocí se nainstaluje minimální množství software potřebného pro započetí instalace, zbývající balíčky se samy stáhnou z internetu. Po zvolení instalace bez grafického prostředí následovala série nastavení. Bylo třeba nastavit:

- Jazyk instalace a nainstalovaného systému + umístění (časové pásmo) + rozložení klávesnice.
- Konfiguraci sítě (veřejnou IP adresu, masku podsítě, gateway, DNS servery) – dle poskytnutých údajů od správce sítě PřF.
- Hostname – dle zvyklostí pojmenování počítačů na PřF – ze složeniny lze zjistit: fakultu, zda se jedná o server či stanici, operační systém a pořadové číslo.
- Doménu – prf.jcu.cz.
- Heslo pro superuživatele (root) – bylo zvoleno heslo, které obsahuje 10 znaků. Tyto znaky jsou velká a malá písmena, číslice a speciální znaky, takže heslo splňuje důležité bezpečnostní požadavky. Navíc tyto znaky jsou více méně náhodné, neznamenají žádné slovo.
- Účet uživatele pro běžnou práci (jméno, přihlašovací jméno a heslo) – zde jsou nároky na bezpečnost hesla o něco menší, proto bylo zvoleno 8 znaků, malá a velká písmena + číslice, ale i tak je heslo dosti silné. Znaky opět nedávají dohromady smysl.
- Rozložení a formátování disku – celý disk rozdělen na dva oddíly, první jako kořenový (souborový systém ext3) a druhý jako swap.

Po těchto nastavení proběhla instalace samotného systému. Poté následovala instalace dalšího software, kdy bylo třeba nastavit:

- „Mirror“ odkud se mají stahovat balíčky – zvolen „mirror“ s českou lokalizací.
- Proxy server – ponecháno prázdné (žádný proxy server.)
- Zda se zúčastnit průzkumu využívání balíků – ne.
- Výběr instalovaného software – pouze standardní systém (standard system utilities) – další možnosti byly např. web server, dns server, mail sever, apod., které byly pro naše účely nežádoucí. Nebylo nainstalováno ani grafické prostředí, které by bylo zbytečné.

Nainstalovaly se potřebné balíky a na závěr bylo nutné potvrdit instalaci GRUB boot loaderu do master boot rekordu (MBR) disku. Jedná se o jediný operační systém na tomto

stroji, nebyl tedy důvod, proč nepotvrdit. Bootování z pevného disku po instalaci systému proběhlo v pořádku. Instalace tedy byla úspěšná.

4.1.1 Zabezpečení serveru

Jako každý jiný server, který se umisťuje do internetu, bylo zapotřebí i tento, zatím bez nainstalovaného honeypotu, zabezpečit. Bylo zjištěno, které služby a procesy jsou aktivní (naslouchají na síti) příkazem *lsof –i* a příkazem *ps –ef*. Byly to portmap (server pro RPC) a exim4 (agent pro přenos poštovních zpráv). Bylo nežádoucí, aby tyto služby byly spuštěny na našem serveru (kvůli dalšímu fungování honeypotu a jejich zranitelnosti), proto byly příkazem *aptitude remove portmap* (popř. *aptitude remove exim4*) odinstalovány. Odinstalace byla nejrychlejší a nejjednodušší možnost, jak zabránit těmto službám ve spuštění např. při restartu systému. Zadáním příkazu *netstat –a* (a také *lsof –i* a *ps –ef*) bylo ověřeno, že žádné služby již na síti nenaslouchají.

Dále bylo vhodné uvažovat o host-based IDS. V tomto případě nepožadujeme složitý HIDS složený ze senzoru a manageru, proto byl vybrán Tripwire ve verzi open-source na kontrolu integrity systému souborů. Pro linuxouvou distribuci je dostupný jako balík ve verzi 2.4.2-9, proto byl jednoduše nainstalován příkazem *aptitude install tripwire*. Při instalaci Tripwire (TW) požadoval dvě hesla (passphrase) pro vytvoření klíčů k bezpečné manipulaci s následujícími soubory: konfigurační soubor a soubor s politikou TW (site-key passphrase) a databáze TW (local-key passphrase). Bezpečnostní politika obou hesel byla zvolena stejná jako u uživatelského účtu operačního systému. Po instalaci následovala editace souboru etc/tripwire/twcfg.txt (ponecháno implicitní nastavení) a příkazem *twadmin --create-cfgfile --site-keyfile site.key twcfg.txt* se vytvořil soubor etc/tripwire/tw.cfg, který TW používá jako konfigurační. Z twcfg.txt se vytvoří tw.cfg až po zadání site-key passhrase, což zabraňuje neoprávněnému změnění konfiguračního souboru. Dále bylo třeba editovat soubor etc/tripwire/twpol.txt, kde je zaznamenáno, které soubory budou kontrolovány. V tomto případě ponecháno implicitní nastavení a zakomentování některých souborů (prefix /proc, /home, /tmp). Postup při vytváření /etc/tripwire/tw.pol (soubor s politikou TW) je identický jako u konfiguračního souboru. Příkaz pro vytvoření je *twadmin --create-polfile --site-keyfile site.key twpol.txt*. Po vytvoření těchto souborů bylo možné inicializovat databázi příkazem *tripwire --init*, proti které se budou kontrolovat vypočtené signatury všech kontrolovaných souborů. Tato databáze je opět chráněná heslem (tentokrát local-key passphrase) proti neoprávněným modifikacím. Dále byl vytvořen soubor etc/cron.daily/tripwire, což zajistí každodenní kontrolu systému a odesílání reportu e-mailem uživateli – vše na základě konfiguračního souboru.

4.1.2 Vzdálený přístup

Posledním úkolem před instalací honeypotu bylo zajištění vzdáleného přístupu na server. V tomto případě nejvhodnějším způsobem, jak zajistit zabezpečený komunikační kanál mezi serverem a vzdálenou stanicí, je použití SSH protokolu. Na serveru byl nainstalován SSH server v open-source variantě (OpenSSH). Instalace proběhla jednoduše pomocí příkazu *aptitude install openssh-server*. Po nainstalování byl zároveň automaticky spuštěn SSH démon (sshd), který implicitně naslouchá na portu 22.

Dále bylo nutné provést určitá nastavení SSH. Toho bylo docíleno editací souboru /etc/ssh/sshd_config. V první řadě bylo nepřípustné, aby přes SSH bylo umožněno přihlásit se jako superuživatel. Změna byla tedy provedena na řádku *PermitRootLogin*. Dále bylo zakázáno *X11Forwarding*. Poslední změny byly na řádcích *RSAAuthentication*, *PubkeyAuthentication* a *AuthorizedKeysFile* pro zajištění autentizace pomocí soukromého a veřejného klíče.

Příkazem *ssh-keygen* byl vygenerován pár RSA klíčů (ponechány implicitní hodnoty – RSA s délkou klíče 2048 bitů) s tím, že byla zadána passphrase pro zašifrování privátního klíče. Passphrase byla o délce 11 znaků a obsahovala velká a malá písmena a číslice. Poté byl zašifrovaný soukromý klíč uložen na klientský počítač, ze kterého bude prováděno připojení na server a obsah veřejného klíče byl příkazem *cat id_rsa.pub >> &h/.ssh/authorized_keys* vložen do souboru, odkud bude vyžadován při autentizaci.

Na závěr bylo provedeno nastavení klientské aplikace PUTTY tak, aby byl použit privátní klíč pro autentizaci.

4.2 Honeypot Dionaea

Low-interaction honeypot Dionaea (momentálně ve verzi 0.1.0) byl vyvinut za finanční podpory Honeynet project během programu Google Summer of Code v roce 2009. Jedná se o mezinárodní program, který umožňuje studentům-vývojářům získat stipendium za účelem vytvoření open-source software. Sami developeři považují Dionaeu za nástupce známého a možná ještě stále nejrozšířenějšího Nepenthesu.

4.2.1 Obecné informace

Čerpáno z webových stránek zabývajících se problematikou honeypotu Dionaea. (Carnivore 2009; Carnivore 200?)

4.2.1.1 Změny oproti Nepenthesu

Oproti Nepenthesu byla Dionaea změněna v mnoha ohledech. Následuje krátký seznam změn a vylepšení:

Detekce shellcode - Nepenthes detekoval shellcode porovnáváním vzorů pomocí regulárních výrazů programovacího jazyka Perl. To znamená, že nebyl schopen rozpoznat nové útoky (tzv. zero-day). Dionaea místo toho používá knihovnu libemu, která detekuje shellcode pomocí emulace a tím je schopna zachytit i neznámý shellcode.

Podpora SMB – Nepenthes jako takový nepodporoval službu SMB na portu 445 (bylo toho docíleno pomocí různých modulů, které simulovaly jen určité zranitelnosti této služby). Dionaea má tuto službu podporovanou přímo v jádře.

Python – Nepenthes pro přidávání modulů používal programovací jazyk C++. Dionaea používá pro přidávání modulů a jako skriptovací jazyk Python.

TLS – Nepenthes TLS nepodporoval, Dionaea využívá openssl.

IPv6 – Přestože v současnosti žádný malware pravděpodobně nepoužívá protokol IP verze 6, Dionaea ho oproti Nepenthesu podporuje.

4.2.1.2 Moduly

Dionaea využívá některé moduly k zajištění detekce na všech poskytovaných službách. Zde jsou uvedeny ty nejdůležitější:

EMU – emu modul využívá knihovnu libemu. Je to knihovna napsaná v jazyku C, její pomocí Dionaea detekuje shellcode. Je to provedeno tak, že Dionaea vytvoří kopii všech vstupů/výstupů připojení, která je předána emu pluginu. Pokud se jedná o spustitelný kód, libemu vytvoří virtuální stroj a v něm kód spustí. Určení, zda se jedná o shellcode a případně jaký, je provedeno sledováním volání API funkcí a jejich parametrů.

CURL – modul určen pro přenos souborů po síti. Stahování souborů pomocí protokolů FTP/TFPT je implementováno v Dionaee jazykem python, stahování protokolem HTTP je realizováno knihovnou libcurl. Curl modul je také zodpovědný za přenos vzorků malware třetím stranám.

PCAP – pcap modul využívá knihovnu libpcap, jejíž pomocí jsou zachytávány pakety. Tuto knihovnu využívá např. známý tcpdump. Úkolem pcap modulu je zaznamenávat odmítnutá spojení. Dozvíme se tedy o pokusu o připojení na službu, kterou Dionaea neposkytuje.

PYTHON – pomocí tohoto modulu je možné využívat python interpreter pro Dionaeu a zároveň Dionaea může využívat nejrůznější skripty napsané v tomto jazyce. Může se jednat například o logsql – logování do sqlite databáze je řešeno pomocí python skriptu nebo o p0f - tato služba poskytuje informace o operačním systému útočníka.

4.2.1.3 Protokoly

Neboli síťové služby, které Dionaea poskytuje:

SMB – je protokol, který slouží ke sdílení prostředků (např. soubory, tiskárny) v síti. Tvůrci Dionaei ho považují za klíčový, protože je hlavním cílem většiny počítačových červů. Služba SMB naslouchá na portu 445.

HTTP – naslouchá na portu 80, podporován je i HTTPS (na portu 443). Bohužel zde Dionaea zaznamenává pouze pokusy o spojení. Žádná data zde nevyhodnocuje.

FTP – na portu 21 je poskytován jednoduchý FTP server, který umožňuje vytváření adresářů a nahrávání a stahování souborů.

MSSQL – otevírá port 1433. Dovoluje přihlášení do databáze a spuštění příkazu nad databází. Více Dionaea nezaznamená kvůli absenci databáze, veškerá interakce s útočníkem zde končí.

SIP – SIP protokol se používá při IP telefonii (VoIP – Voice over IP). Modul neumožňuje připojení na externí VoIP server.

4.2.1.4 Logování

Dionaea používá vedle zápisu všech událostí do jediného textového souboru *dioanaea.log* také zapisování všech útoků či pokusů o útok do sqlite databáze. To přináší četné výhody, jako je například dotazování se pomocí SQL jazyka. Jednoduchým dotazem lze zjistit např. na jaký port byly vedeny nejčetnější útoky, či z jakých IP adres byl honeypot nejčastěji atakován. Lze také využít dvou utilit napsaných v pythonu. Readlogsqltree je utilita, která vypíše útoky z databáze ve stromové struktuře. Pomocí přepínačů lze nadefinovat různá kritéria jako je například čas útoků či typ. Gnuplotsql utilita ve spolupráci s programem Gnuplot vytvoří z databázových dat přehledný graf útoků.

4.2.2 Instalace

Instalace honeypotu Dionaea není tak jednoduchá, jak tomu bylo u Nepenthesu, kde existoval .deb balík, jež lze nainstalovat jednoduchým příkazem *aptitude install nepenthes*. Je zde mnoho závislostí (knihoven a modulů), které bylo nutno vyřešit. Většinou se ale jedná o knihovny, jejichž potřebná verze je vyšší než ta, která je dostupná pro stabilní distribuci Debianu. Proto bylo nutné zkompilovat zdrojové kódy těchto knihoven a až poté je

nainstalovat. Na oficiálních webových stránkách projektu Dionaea je velmi podrobný návod[9] pro instalaci Dionaea pod operačními systémy Ubuntu a Debian, podle kterého jsem postupoval. Zde budou uvedeny pouze problémy při instalaci a jejich řešení.

4.2.2.1 Python

První nesrovnalosti se vyskytly při kompilaci pythonu verze 3.1.2. Postup byl následující:

Stažení souboru se zdrojovými kódy + rozbalení archivu:

wget http://python.org/ftp/python/3.1.2/Python-3.1.2.tgz

tar xfz Python-3.1.2.tgz

Konfigurace:

./configure --enable-shared --prefix=/opt/dionaea --with-computed-gotos \

--enable-ipv6 LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/"

Kompilace zdrojových kódů:

Make

Výsledek kompilace byl následující:

Python build finished, but the necessary bits to build these modules were not found:

_dbm *_gdbm* *_tkinter*

bz2

Řešení:

Bylo tedy nutné doinstalovat potřebné moduly. Jednalo se o libgdbm-dev, tk-dev, libbz2-dev. To bylo provedeno příkazem *aptitude install libgdbm-dev tk-dev libbz2-dev*, protože nebylo uvedeno, zda je potřeba nainstalovat vyšší verzi než je dostupná pro stabilní Debian. Následně byla znovu provedena kompilace příkazem *make*, která napodruhé s doinstalovanými moduly proběhla v pořádku.

Instalace:

make install

Instalace proběhla úspěšně.

4.2.2.2 UDNS

Stažení souboru se zdrojovými kódy + rozbalení archivu:

wget http://www.corpit.ru/mjt/udns/old/udns_0.0.9.tar.gz

tar xfz udns_0.0.9.tar.gz

Konfigurace:

./configure

Vytvoření sdílené knihovny:

make shared

---

[9] http://dionaea.carnivore.it/#compiling

Zkopírování hlavičky do adresáře /opt/dionaea/include:

```
cp udns.h /opt/dionaea/include/
```

Zkopírování knihovny do adresáře s knihovnami /opt/dionaea/lib:

*cp *.so* /opt/dionaea/lib/*

Výsledek kopírování:

cp: cannot stat `.so*': No such file or directory*

Řešení:

Po vypsání obsahu adresáře příkazem *ls –la* bylo zjištěno, že se zde opravdu soubory potřebné ke zkopírování nevyskytují. Předpokládaným důvodem byly problémy s verzí knihovny, jež byla 0.0.9, proto byla stažena z internetu knihovna verze 0.1 a celý postup byl zopakován:

wget http://www.corpit.ru/mjt/udns/udns-0.1.tar.gz

tar xfz udns-0.1.tar.gz

./configure

make shared

```
cp udns.h /opt/dionaea/include/
```

*cp *.so* /opt/dionaea/lib/*

Vytvoření symbolického odkazu:

```
cd /opt/dionaea/lib
ln –s libudns.so.0 libudns.so
```

„Instalace“ proběhla úspěšně.

4.2.2.3 Dionaea

Získání souborů z repozitáře:

git clone git://git.carnivore.it/dionaea.git dionaea

Konfigurace:

autoreconf -vi

./configure --with-lcfg-include=/opt/dionaea/include/ \

--with-lcfg-lib=/opt/dionaea/lib/ \

--with-python=/opt/dionaea/bin/python3.1 \

--with-cython-dir=/usr/local/bin \

--with-udns-include=/opt/dionaea/include/ \

--with-udns-lib=/opt/dionaea/lib/ \

--with-emu-include=/opt/dionaea/include/ \

--with-emu-lib=/opt/dionaea/lib/ \

--with-gc-include=/usr/include/gc \

--with-ev-include=/opt/dionaea/include \

--with-ev-lib=/opt/dionaea/lib \

--with-nl-include=/opt/dionaea/include \

--with-nl-lib=/opt/dionaea/lib/ \

--with-curl-config=/opt/dionaea/bin/ \

--with-pcap-include=/opt/dionaea/include \

--with-pcap-lib=/opt/dionaea/lib/ \

--with-glib=/opt/dionaea

Výsledek konfigurace:

checking netlink/netlink.h usability... no

checking netlink/netlink.h presence... yes

configure: WARNING: netlink/netlink.h: present but cannot be compiled

configure: WARNING: netlink/netlink.h: check for missing prerequisite headers?

configure: WARNING: netlink/netlink.h: see the Autoconf documentation

configure: WARNING: netlink/netlink.h: section "Present But Cannot Be Compiled"

configure: WARNING: netlink/netlink.h: proceeding with the compiler's result

configure: WARNING: ## ---------------------------------- ##

configure: WARNING: ## Report this to nepenthesdev@gmail.com ##

configure: WARNING: ## ---------------------------------- ##

checking for netlink/netlink.h... no

Řešení:

Dle instrukcí byla chyba zaslána na nepenthesdev@gmail.com. Náprava chyby:

echo "#include <linux/netlink.h>" > /opt/dionaea/include/netlink/netlink-kernel.h (dle doručeného e-mailu). Provedena znovu konfigurace. Tentokrát proběhla v pořádku, avšak bylo zjištěno (z výpisu konfigurace), že chybí modul nfq.

Doinstalace chybějícího modulu nfq:

aptitude install libnetfilter-queue-dev

Opětovné provedení konfigurace.

Kompilace a Instalace:

make; make install

Instalace proběhla úspěšně.

4.2.3 Konfigurace, spuštění a ověření běhu

Po úspěšné instalaci bylo třeba editovat soubor /opt/dionaea/etc/dionaea/dionaea.conf. Pro samotný bezchybný běh honeypotu nebylo třeba do konfiguračního souboru zasahovat, bylo ale vhodné pozměnit e-mailovou adresu, na kterou budou posílány rozbory zachycených vzorků malware.

```
submit =
{
  defaults = {
        urls = ["http://anubis.iseclab.org/nepenthes_action.php",
              "http://onlineanalyzer.norman.com/nepenthes_upload.php",
              "http://luigi.informatik.uni-mannheim.de/submit.php?action=verify"]
        email = "dmdionaea@gmail.com"
        file_fieldname = "upfile"
        MAX_FILE_SIZE = "1500000"
        Submit        = "Submit for analysis"
  }

  joebox = {
        urls = ["http://analysis.joebox.org/submit"]
        email = "dmdionaea@gmail.com"
        file_fieldname = "upfile"
        MAX_FILE_SIZE = "1500000"
        submit        = "Submit for analysis"
        service       = "agree"
        xp            = "1"
        vista         = "1"
        w7            = "1"
        pcap          = "1"
  }
```

Program lze spustit z /opt/dionaea/bin příkazem *./dionaea* pomocí více přepínačů. V tomto případě byly použity přepínače *-D -l all,-debug -L '**', které zajistily, aby bylo logováno úplně vše a program se spustil na pozadí jako démon.

Ověření, zda je Dionaea opravdu spuštěná, proběhlo několika způsoby:

Výpisem běžících procesů příkazem *ps –ef*

```
root    31494     1  0 16:29 ?        00:00:05 ./dionaea -D -l all,-debug -L *
root    31495 31494  0 16:29 ?        00:00:00 ./dionaea -D -l all,-debug -L *
```

Výpisem běžících procesů naslouchajících na síti příkazem *lsof -i |grep dionaea*

dionaea 31494 root 15u IPv4 1290914 0t0 TCP localhost:nameserver (LISTEN)

dionaea 31494 root 16u IPv4 1290915 0t0 TCP localhost:microsoft-ds (LISTEN)

Celý výpis je uveden v příloze.

Výpisem otevřených portů ve spojení s procesy příkazem *netstat -ano --tcp –p*:

tcp 0 0 127.0.0.1:1433 0.0.0.0: LISTEN 31494/dionaea off (0.00/0/0)*

tcp 0 0 160.217.209.190:443 0.0.0.0: LISTEN 31494/dionaea off (0.00/0/0)*

Celý výpis je uveden v příloze.

Oskenováním serveru nástrojem nmap na otevřené porty

PORT	*STATE*	*SERVICE*
21/tcp	*open*	*ftp*
22/tcp	*open*	*ssh*
80/tcp	*open*	*http*
135/tcp	*open*	*msrpc*
443/tcp	*open*	*http*
445/tcp	*filtered*	*microsoft-ds*
1433/tcp	*open*	*ms-sql-s*

4.2.4 Řešení problémů

Po spuštění a ověření běhu honeypotu bylo třeba vyřešit řadu problémů. Dále jsou uvedeny ty nejkritičtější, které zabraňovaly normálnímu fungování honeypotu a jejich řešení.

4.2.4.1 Chyba v programu

Chyba v kódu zapříčinila, že se program dostal do nekonečné smyčky a tím pádem nemohl správně fungovat. Po třech dnech běhu nebyla zaznamenána žádná síťová aktivita a velikost logovacího souboru narostla na 69 GB dat. Byl kontaktován jeden z vývojářů Dionaei, který chybu opravil. Poté stačilo pouze provést update.

4.2.4.2 Filtrovaný provoz

Dalším problémem bylo, že honeypot neukládal žádné vzorky malware – binární soubory. Pomocí skriptů readlogsqltree a gnuplotsql bylo zjištěno, že program nedetekuje žádný shellcode a nepřijímá žádné nabídky (offers) na stažení souboru. Klíčovou informací bylo, že útok na službu SMB (port 445) byl prováděn vždy pouze z jedné IP adresy z univerzitního rozsahu. Z toho tedy bylo zřejmé, že je provoz na portu 445 pravděpodobně někde v síti filtrován. Pro ověření této domněnky byl server oskenován programem nmap (konkrétně port 445) z místa, na kterém zcela jistě není filtrován odchozí provoz. Na serveru byl spuštěn program tcpdump pro zachytávání paketů a bylo zjištěno, že pakety směřující na port 445 se opravdu na server vůbec nedostanou. Bylo tedy třeba zajistit zrušení filtrování paketů pro IP adresu serveru.

Avšak tím problém nebyl zcela vyřešen, jelikož síťový provoz na službách NetBIOS byl filtrován již v síti CESNET2, do které je univerzita připojena. Proto bylo znovu vyjednáno zrušení filtrování.

4.2.4.3 Directory traversal

Directory traversal útok zneužívá chybu v programu, kdy jsou nedostatečně ošetřeny vstupy – v tomto případě URL. Dionaea obsahovala chybu v HTTP službě, prostřednictvím které bylo možné za pomoci vhodně zvolené URL získat jakýkoliv soubor operačního systému, jako je například /etc/shadow či /etc/passwd. Využití této chyby bylo možné pouze v případě, že honeypot byl spuštěn pod uživatelem root.

Řešením tedy bylo spustit program pod jiným uživatelem a změnit kořenový adresář pro daný proces (systémové volání chroot) – což Dionaea umožňuje při spuštění pomocí přepínačů. Konkrétně *-u nobody -g nogroup -r /opt/dionaea/. C*hyba byla navíc odeslána vývojářům.

Toto řešení zapříčinilo problémy se zaznamenáváním útoků do sqlite databáze, které byly avizovány na webových stránkách[10]. Po opravení chyby directory traversal byl program opět spuštěn pod uživatelem root a byl proveden test, zda se chyba v HTTP službě již nevyskytuje. Testovací řetězec pro získání souboru /etc/shadow byl (X.X.X.X představuje IP adresu – z bezpečnostních důvodů zde není uvedena):

http://X.X.X.X/%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fetc%2f%2fshadow

5 Analýza činnosti malware

Tato kapitola obsahuje analýzu útoků zachycených honeypotem jménem Dionaea, který byl umístěn v univerzitní počítačové síti. Honeypot zaznamenával útoky do souboru logsql.sqlite (sqlite databáze). K analýze byla použita data zaznamenaná od 5. 3. 2011 do 11. 3. 2011 a pak od 17. 3. 2011 do 6. 4. 2011 – tedy 28 dní. Log soubor měl k tomuto dni velikost 3 026 944 B. Za tuto dobu se podařilo honeypotu uložit 4 unikátní vzorky malware. Unikátnost byla posouzena na základě MD5 kontrolních součtů souborů, avšak jedná se o soubory, které jsou stejně kategorizovány a mají pravděpodobně totožnou funkci.

Nejprve byl vytvořen přehled všech útoků, ze kterého byly vybrány útoky pro analýzu. Kritériem pro výběr dat pro analýzu bylo to, zda honeypot byl schopen zaznamenat nějaké informace o útoku, popř. zajímavost či důležitost útoku.

5.1 Přehled útoků

Pro vytvoření přehledu útoků byly použity dotazy do logsql.sqlite databáze zveřejněné na webových stránkách Carnivore, které se zabývají honeypoty Dioanea a Nepenthes. (Carnivore 2009) Tyto dotazy byly pozměněny či použity v nezměněné podobě.

Celkový počet provedených spojení na honeypot byl 9208. V tomto součtu jsou zahrnuty spojení typu accept (872) – přijatá spojení na porty 21, 80, 135, 445 a 1433 protokolem TCP; typu connect (44) – spojení pomocí UDP protokolu na port 5060 (SipSession) a typu reject (8291) – odmítnutá spojení, většinou se jednalo o sken portů. Dále budou brány v úvahu pouze spojení typu accept.

[10] http://dionaea.carnivore.it/#logsql

Porty, na které přišlo nejvíce spojení, jsou 1433 (MSSQL) a 80 (HTTP). V tabulce ve sloupci hitcount lze vidět počet připojení a ve sloupci port číslo portu.

hitcount	port
9	21
248	80
24	135
125	445
465	1 433

Obrázek 4 – Tabulka nejfrekventovanějších portů

IP adresy, z kterých byl nejčastěji veden útok, byly 119.194.170.15, 1.202.139.80 a 200.161.82.63. Tyto adresy pocházejí z Jižní Korey, Číny a Brazílie. Prvních pět nejfrekventovanějších IP adres je uvedeno v příloze.

Nejvíce útoků v průběhu dne bylo zaznamenáno mezi jedenáctou a čtrnáctou hodinou (více jak tisíc) a pak kolem osmnácté hodiny. Celkový přehled počtu útoků v průběhu dne je uveden v příloze.

Dále byly zachyceny pouze 4 vzorky malware šířícího se pomocí protokolu SMB na portu 445 a zaznamenány dva útoky z rozsahu IP adres patřící do sítě Jihočeské univerzity. Větší množství útoků ve vnitřní síti pravděpodobně nebude možné odhalit z důvodu filtrovaného provozu. Větší počet vzorků malware pravděpodobně také nebude možno zachytit, protože honeypot funguje pouze na jedné IP adrese, která se snadno může dostat na tzv. blacklisty IP adres, kterým se budou červy vyhýbat.

Z tohoto přehledu byly vybrány pro další analýzu vzorky malware, útoky na službu MSSQL a útoky pocházející z vnitřní sítě. Služba MSSQL byla vybrána, protože na ní bylo vedeno nejvíce útoků a honeypot dokázal zaznamenat o tomto typu útoku nejvíce informací.

5.2 Útoky na MSSQL

Pro zkoumání útoků na MSSQL byly použity SQL dotazy zveřejněné na webových stránkách Carnivore. (Carnivore 2010) Za 28 dní bylo provedeno celkem 465 útoků na port 1433. Jako aplikační rozhraní pro přístup k datům byla červy použita ODBC a OLEDB. Použité aplikace pro přístup k SQL serveru buď nebyly detekovány vůbec, nebo byly použity OSQL-32 a AttricScanner – pravděpodobně se nejedná o standardní aplikaci, nýbrž o malware.

Pro autentizaci bylo použito celkem 7 různých přihlašovacích jmen a 124 jedinečných hesel (zahrnuta i prázdná hesla). V následující tabulce jsou vypsána uživatelská jména a počty jejich použití. Ve sloupci COUNT je počet použití daného přihlašovacího jména, ve sloupci login_username jsou jednotlivá jména a ve sloupci COUNT(login_password) je počet jedinečných použitých hesel pro dané přihlašovací jméno.

COUNT	login_username	COUNT(login_password)
302	sa	112
24	admin	23
24	database	24
24	root	23
24	server	23
24	sql	23
20	administrator	20

Obrázek 5 – Tabulka použitých přihlašovacích jmen

Kombinace přihlašovacího jména SA a prázdného hesla je u Microsoft SQL serveru implicitní. Ostatní jména jsou taková, která se v praxi často vyskytují. K nim jsou zkoušena nejrůznější hesla. Nejčastěji prázdné heslo, poté hesla admin, server a sql. V tabulce je pět nejvíce použitých hesel. Ve sloupci COUNT je počet použití hesla a ve sloupci login_password je hodnota použitého hesla.

COUNT	login_password
71	
12	admin
10	server
10	sql
9	12345

Obrázek 6 – Tabulka pěti nejpoužívanějších hesel

V příloze je uveden seznam padesáti nejpoužívanějších hesel. Tato hesla a žádná jim podobná by v žádném případě neměla být v praxi používána. To samozřejmě platí i pro přihlašovací jména, která by také neměla zůstávat implicitní. Pro hesla by měla být zvolena dostatečná bezpečnostní politika - alespoň 8 znaků pro heslo, použití malých a velkých písmen, číslic a speciálních znaků. Obzvláště pokud se jedná o administrátorské či superuživatelké účty, hesla by měla být maximálně bezpečná a neuhodnutelná.

5.3 Analýza zachyceného vzorku malware

Binární soubor s MD5 kontrolním součtem 022aeb126d2d80e683f7f2a3ee920874 byl zachycen 18. března 2011 v 20:17 z umístění smb://38.65.64.14/csrss.exe pomocí SMB protokolu. IP adresa je z rozsahu 38.0.0.0/8, který patří americké společnosti Cogent Communications. Velikost souboru je 65 814 B.

Postup analýzy vzorku malware:

1. Zaslání souboru do on-line analyzátorů
2. Sken souboru antivirovým programem
3. Vypsání textových řetězců v souboru příkazem strings
4. Spuštění souboru v sandboxu
5. Spuštění souboru ve virtuálním stroji
6. Porovnání vlastních poznatků s on-line analýzou

5.3.1 Online analyzátory

Dionaea umožňuje automatické zasílání získaného malware do on-line analyzátorů pomocí modulu submit_http.py. V tomto případě byly využity služby Norman Sandbox[11] a Anubis[12]. Avšak oba reporty se značně rozcházejí v síťové aktivitě tohoto počítačového červa. Shodují se pouze kategorizace malware (W32/EMailWorm – Norman, popřípadě Email-Worm.Win32.Atak – Anubis). Manipulace se soubory ve zprávě od Norman Sandbox analyzátoru není vůbec zmíněna. Aktivita po síti se u obou reportů také rozchází. Norman Sandbox zmiňuje šíření přes e-mail, Anubis analyzátor zmiňuje četné změny registrů OS.

Síťová aktivita dle Norman Sandboxu – připojení na adresy 210.204.41.7 a 210.204.41.1 na port číslo 445 a na yahoo.no na port 25. Dále připojení na \\210.204.41.1\ipc$ s různými kombinacemi přihlašovací jméno + heslo – jméno a heslo prázdné; jméno CretoaUe a heslo CretoaUe; jméno CretoaUe a heslo prázdné.

Síťová aktivita dle Anubise – MX DNS dotaz na gmail.com, skenování různých IP adres na port 445 a pokusy o připojení na tyto adresy na port 445.

5.3.2 Sken souboru antivirovým programem

Pro sken souboru byl použit antivirový nástroj Avira Antivir Premium. Tímto A/V nástrojem byl škodlivý soubor kategorizován jako TR/Agent.mtv. Společnost Avira o tomto malware žádné informace neposkytuje, avšak společnost McAfee ano. Jsou zde pouze jiné konvence v pojmenování. McAfee malware nazývá W32/Bagz.a@MM, Avira ho nazývá TR/Agent.mtv a společnost Norman W32/EMailWorm. Zde tedy dochází ke shodě, avšak porovnáním reportů Norman Sandbox a Anubis analyzátorů a informací od McAfee lze zjistit shodu pouze v šíření červa – pomocí e-mailu. Ostatní charakteristiky se neshodují. (McAfee, Inc. c2003-2011)

5.3.3 Použití příkazu *strings*

Pomocí příkazu *strings* v „unixových“ operačních systémech lze vypsat tisknutelné textové řetězce v souborech. Tedy i v binárních souborech. Následuje výčet nejzajímavějších nalezených řetězců.

E-mailové adresy - johnS203@yahoo.com a whiat1001@gmail.com. První adresa na serveru yahoo.com se zdá být nezaregistrovaná.

[11] http://www.norman.com/security_center/security_tools/

[12] http://anubis.iseclab.org/

Hesla – nalezené řetězce vypadají jako hesla, která červ zkouší pro autentizaci nějakému serveru či službě. Příklady těchto potencionálních hesel jsou : mail1234, mail123, mail1, web1234, web123, web1, mail, 111111, , 4321, 54321, 1234567, 12345, asdfgh, asdfg, angel, passwd, !@#$, root, !@#$%, admin, test1234, pass, 654321, 123456, password, db2admin, administrator.

IP adresy - 209.85.223.33, 209.85.210.24 a 209.85.223.27 patří společnosti Google. S největší pravděpodobností se jedná o mailové servery.

Příkazy - cmd.exe /c "net share c$ /d", cmd.exe /c "net share admin$ /d" a cmd.exe /c "net share admin$" mají za úkol zrušit a znovu vytvořit sdílené zdroje.

5.3.4 Spuštění malware v sandboxu

Před spuštěním malware v sandboxu bylo připraveno virtuální prostředí pomocí nástroje Oracle VM VirtualBox ve verzi 4.0.4. Jako operační systém virtuálního počítače byl zvolen Microsoft Windows XP SP 3. V tomto virtuální počítači byl nainstalován nástroj Sandboxie. Sandbox je nástroj, který dovoluje spuštění programů v izolovaném prostředí (tzv. sandboxu), což zabraňuje těmto programům provádět změny v jiných programech či datech v počítači. (*Sandboxie c2004-2011*) Je tedy možné bezpečně spouštět neznámé či škodlivé programy bez rizika nákazy počítače.

Vzorek malware byl přejmenován na m1.exe a spuštěn v sandboxu. Programu m1.exe byla v sandboxu povolena komunikace po internetu. Pomocí nástroje Wireshark na zachytávání paketů byla sledována síťová aktivita. Byl zaznamenán pouze MX DNS dotaz na gmail.com. Proces byl ukončen kvůli své neaktivitě.

5.3.5 Spuštění malware ve virtuálním počítači

Dalším krokem bylo spuštění binárního souboru ve virtuálním počítači (VM). Před tím byl udělán snímek VM, aby bylo možno vrátit se do stavu před spuštěním malware. Dále byl pomocí nástroje SysInspector zkontrolován systém a log soubor uložen pro pozdější kontrolu činnosti malware.

Po spuštění souboru m1.exe wireshark zaznamenal opět MX DNS dotaz na gmail.com a navíc byly zachyceny pakety představující sken IP adres na port 445. Po opětovném zkontrolování systému nástrojem SysInspector a porovnáním obou logů (před spuštěním a po spuštění malware) nebyl zjištěn žádný zásah do registrů či souborů OS. Opětovné spuštění malware a porovnání obou logů programu Wireshark ukázalo, že se nejspíše jedná o náhodně generováné IP adresy.

Druhý pokus byl proveden analogicky, avšak byl odpojen síťový kabel. Jediná změna v chování červa oproti prvnímu pokusu byla v opakování dotazu do DNS serveru. Po několika pokusech to malware „vzdal“ a spustil sken IP adres.

5.3.6 Porovnání výsledků a shrnutí

Porovnány budou výsledky vlastního výzkumu s výsledky on-line analýzy obou výše zmíněných společností.

První shoda byla v kategorizaci malware při skenu antivirovým programem. Zde musel být brán zřetel na různé konvence antivirových společností při pojmenovávání malwaru. Další shoda byla ve struktuře e-mailu, jehož prostřednictvím by se podle Norman Sandbox reportu měl červ šířit. Tato struktura byla popsána v reportu a také nalezena pomocí příkazu *strings* v binárním souboru. Poslední shodou je síťová aktivita dle Anubis analyzátoru. Shoduje se jak DNS dotaz, tak sken IP adres na port 445.

Ostatní aktivity zaznamenané oběma společnostmi se s vlastním výzkumem neshodují. Zásah do registrů zaznamenán nebyl žádný a síťová aktivita byla dle Norman Sandbox zcela odlišná (liší se IP adresy i hesla nalezené v binárním souboru).

V některých aktivitách se tedy chování malware liší oproti reportům zkušených společností, v jiných zase shoduje. To může být způsobeno mnoha faktory. Může se jednat například o malware, který v sobě obsahuje ochranu před spouštěním v sandboxu či kód, který rozpozná, zda je spuštěn ve virtuálním počítači. Popřípadě může obsahovat jiné ochrany, aby nebylo možné poznat, co je jeho účelem. Může se také jednat o mutaci téhož malware. Tato tvrzení jsou však pouze spekulace, které nejsou nijak podložené.

5.4 Aktivita ve vnitřní síti

Po dobu provozu honeypotu se povedlo zaznamenat dvě aktivity z vnitřní sítě, což lze považovat za úspěšnou detekci infikovaných strojů. Aktivita malware na těchto strojích byla totožná. Probíhala následovně:

Nejprve infikovaný stroj zjistil, zda je otevřena služba (NetBIOS) na portu 139. Zde honeypot spojení odmítl a zaznamenal pokus o spojení do databáze útoků. Dále se infikovaný počítač pokusil připojit na port 139 a 445. Po úspěšném připojení nechal vypsat informace o všech sdílených zdrojích na serveru (pomocí dce/rpc volání funkce NetShareEnum()) a následně použitím funkce NetServerGetInfo() zjistil informace o aktuální konfiguraci serveru. Po obdržení informací červ ukončil spojení a dále nijak nepostupoval. Tyto informace byly zjištěny z logu honeypotu a pomocí nástroje tcpdump na zachytávání paketů.

K dispozici byl binární soubor, který pravděpodobně zapříčinil infekci jednoho systému. Testování tohoto vzorku malware proběhlo analogicky jako první analýza binárního souboru (viz. 5.3 Analýza zachyceného vzorku malware).

Soubor, jehož název je bmvyzo.exe, má kontrolní součet MD5 15331e7f88975864d15d6364c0c80e7a a velikost 539 kB. Velikost tohoto malware (a také velikost malware zachyceného honeypotem) potvrzuje všeobecně známé pravidlo, že škodlivé programy by měly být co „nejmenší“, z důvodu rychlého přenosu po síti a těžší identifikace škodlivé aktivity.

5.4.1 On-line analyzátory

On-line analyzátory v tomto případě neposkytly žádná zajímavá data. Síťovou aktivitu nezaznamenal žádný z nich. Pouze Anubis zaznamenal změny v registrech OS, avšak činnost škodlivého programu byla předčasně ukončena chybou.

5.4.2 Sken souboru antivirovým programem

Sken byl opět proveden pomocí A/V nástroje společnosti Avira. Byl klasifikován jako WORM/AutoIt.xl. Tento typ malware se dle společnosti McAfee šíří pomocí USB Flash disků a je schopen stáhnout do počítače dodatečné soubory. (McAfee, Inc. c2003-2011)

5.4.3 Použití příkazu *strings*

Pomocí příkazu *strings* nebyly nalezeny žádné zajímavé či užitečné řetězce, pouze některá systémová volání, jako například *LoadLibraryA* nebo *GetProcAddress*.

5.4.4 Spuštění malware v sandboxu

Při spuštění malware v sandboxu nebyl téměř vůbec aktivní. Pouze změnil svůj název z bmvyzo.exe na crscs.exe a odeslal DNS dotaz na whatismyip.com. S touto informací nijak nenaložil.

5.4.5 Spuštění malware ve virtuálním počítači

Dalším krokem bylo spuštění malware ve virtuální počítači. DNS dotaz na whatismyip.com se opakoval. Následovalo zjištění IP adresy „inkovaného“ stroje. Dále bylo provedeno TCP spojení na stanici, která byla lokalizována v Rusku. S touto stanicí proběhl přenos šifrovaných dat. Dále proběhl DNS dotaz na server, který poskytuje údaje o lokalizaci IP adres. Na základě těchto informací malware zjistil svoje umístění (EU). Na závěr malware spustil sken IP adres na port 445. Tentokrát nebyly IP adresy generovány náhodně, ale byly použity ty z vnitřní podsítě, v které byl umístěn „infikovaný“ počítač. Malware začal na první adrese v podsíti a její hodnotu vždy zvýšil o 1. Zde byl pokus ukončen, aby nedošlo k infekci jiného stroje. Změny v registrech OS nebyly pomocí nástroje SysInspector zaznamenány žádné, avšak po restartu virtuálního počítače došlo i ke spuštění malware – byl vytvořen proces crscs.exe pod aktuálním uživatelem a proběhl DNS dotaz. Dotaz byl neúspěšný z důvodu odpojení počítače od sítě.

5.4.6 Shrnutí

Na základě těchto poznatků usuzuji, že by se mohlo jednat o poměrně nebezpečného počítačového červa, který se šíří nejen pomocí USB Flash disků, ale i po síti. Šifrovaná komunikace s počítačem (či serverem) v Rusku by mohla znamenat začlenění do botnetu – toto je opět pouze spekulace. Zajímavé je, že on-line analyzátory nezaznamenaly žádnou síťovou aktivitu, která byla dle vlastního pozorování velmi značná.

6 Návrhy na další práci

Nainstalovaný low-interaction honeypot Dioanea v kombinaci s dalšími nástroji (např. A/V nástroj, IDS a firewall) je dle mého názoru dostatečná detekční metoda použitelná v praxi, avšak hodí se spíše pro tzv. produkční detekci, kdy jde hlavně o bezpečnost počítačů v síti.

Pro výzkumné účely by bylo vhodné zabývat se high-interaction honeypotem, popř. přímo honeynetem. Návrh na další práci tedy je nasadit do sítě high-interaciton honeypot, popř. celý honeynet, který by byl schopen zaznamenávat více útoků. Praktické využití tohoto řešení by mohlo být například to, že takovýto honeypot by mohl být zdrojem signatur pro IDS či A/V nástroj.

Dalším návrhem pro práci do budoucna je podrobněji se zabývat analýzou zachycených vzorků malware, což úzce souvisí s high-interaction honeypotem. Zajímavým řešením by bylo vytvořit simulované reálné prostředí a zjistit, jak se v tomto prostředí počítačový červ chová. Toto prostředí by se mohlo skládat z dvou počítačů. Na jednom by byl spuštěn vzorek červa, druhý by byl zranitelnou obětí. Jak bylo zjištěno, červ náhodně generuje IP adresy. Bylo by tedy třeba dát do cesty v síti nějaké zařízení, které by bylo schopno přesměrovat pakety na připravenou oběť. To by bylo proveditelné pomocí jednoduchého linuxového serveru s dvěma síťovými rozhraními a pro tyto účely správně nakonfigurovaným překladem adres (NAT) v iptables. V tomto prostředí by bylo možné sledovat chování škodlivého programu hlavně v případě, kdy skenovaný počítač kladně odpoví. Další výhodou tohoto

prostředí by byla absence virtuálního stroje a sandboxu, proti kterým může škodlivý kód obsahovat ochranu. Jednalo by se vlastně o high-interaction honeypot.

7 Závěr

Na závěr této práce bych rád shrnul co vše bylo uděláno a také z jakého důvodu. V první části byla popsána struktura a fungování botnetů. To zahrnuje vysvětlení, co je botnet, jaké je jeho složení a jaké jsou topologie C&C mechanizmu. Dále tato část zahrnuje popis různých druhů botnetů, jak infikované počítače komunikují se svým majitelem, jak se z počítače může stát bot-klient a celkové využití botnetů. Pokud jsme schopni porozumět tomu, jak botnety fungují, bude mnohem snadnější porozumět metodám jak se jim bránit a detekovat je. Touto částí jsem chtěl přispět k rozšíření zdrojů zabývajících se problematikou botnetů, protože dle mého názoru je stále málo literatury v českém jazyce, a proto je tato část zpracována podrobněji.

V druhé části jsou rozebrány metody detekce počítačových červů a botnetů. Jednotlivé metody jsou popsány a je vytvořen přehledný seznam těchto metod spolu s volně dostupnými nástroji. Přehled je využitelný při výběru vhodných metod detekce v praxi.

Hlavními cíli této práce bylo praktické nasazení honeypotu do počítačové sítě a analýza útoků na tento honeypot. Tomu je věnována třetí a čtvrtá část práce. Byl vybrán software představující low-interaction honeypot. Na základě tohoto výběru byl zvolen operační systém. Obojí bylo nainstalováno a chyby a problémy bránící správnému chodu honeypotu byly vyřešeny. Za necelý měsíc provozu honeypotu bylo zaznamenáno velké množství útoků. Za velký úspěch považuji detekci dvou škodlivých aktivit v rámci univerzitní sítě. V rámci analýzy útoků na službu MSSQL vznikl seznam padesáti nejpoužívanějších hesel, která by se rozhodně neměla objevit v reálných systémech. Byla provedena analýza vzorku malware, která spočívala mimo jiné ve spuštění škodlivého kódu v sandboxu a ve virtuálním počítači. Mé vlastní pozorování nebylo zcela totožné s výsledky od společností zabývajících se analýzou malware. Na základě těchto výsledků byla navrhnuta další práce.

Cíle tedy byly splněny. Tato práce bude vhodným materiálem, který usnadní orientaci v problematice botnetů a jejich detekce.

Jak již bylo zmíněno, za necelý měsíc bylo zaznamenáno více než devět tisíc pokusů o připojení na honeypot. Z toho více jak devět set připojení by se dalo nazvat pokusem o útok. Tato čísla dokazují, že hrozba narušení bezpečnosti počítačů v internetu je reálná a že bychom ji neměli v žádném případě podceňovat.

8 Seznam použité literatury

Carnivore [online]. 2009. Introducing dionaea. Dostupné z WWW: <http://carnivore.it/2009/10/27/introducting_dionaea>.

Carnivore [online]. 2009. Logging. Dostupné z WWW: <http://carnivore.it/2009/11/06/dionaea_sql_logging>.

Carnivore [online]. 2010. MSSQL attacks examined. Dostupné z WWW: <http://carnivore.it/2010/09/11/mssql_attacks_examined>.

Carnivore [online]. [200?]. Dionaea - catches bugs. Dostupné z WWW: <http://dionaea.carnivore.it/>.

Cisco Systems, Inc. CCNA Exploration : Network Fundamentals. In *Official Course Material* [online]. Version 4.0. [200?]. Dostupné z WWW: <https://auth.netacad.net/idp/Authn/NetacadLogin>.

Cisco Systems, Inc. *Cisco* [online]. Volume 9, Number 4. [200?]. Defenses Against TCP SYN Flooding Attacks. Dostupné z WWW: <https://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-4/syn_flooding_attacks.html>.

COOKE, Evan; JAHANIAN, Farnam; MCPHERSON, Danny The Zombie Roundup : Understanding, Detecting, and Disrupting Botnets. In *SRUTI'05 : Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop* [online]. USA : USENIX Association Berkeley, CA, USA ©2005, 2005. Dostupné z WWW: <www.eecs.umich.edu/~emcooke/pubs/botnets-sruti05.pdf>.

ESET, spol. s r.o. *ESET : Chráníme vaše digitální světy* [online]. [200?]. Rejstřík pojmů. Dostupné z WWW: <http://www.eset.cz/podpora/rejstrik/>.

GULLETT, David. *Symmetrix Technologies : News and Articles* [online]. 2010 [cit. 2011-04-11]. Snort 2.8.6 and Snort Report 1.3.1 on Ubuntu 10.04 LTS Installation Guide. Dostupné z WWW: <http://www.symmetrixtech.com/articles/004-snortinstallguide286.pdf>.

KRČMÁRIK, Marián. *Analýza možnosti detekcie botnet sietí pomocou NetFlow dát*. Brno, 2009. 57 s. Diplomová práce. Masarykova univerzita, Fakulta informatiky.

LIGH, Michael H., et al. *Malware Analyst's Cookbook and DVD*. Indianapolis (Indiana) : Wiley Publishing, Inc., 2011. 746 s. ISBN 978-0-470-61303-0.

MAURIC, Jakub. *IDS systém SNORT*. České Budějovice, 2009. 59 s. Bakalářská práce. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta, Katedra informatiky.

McAfee, Inc. *McAfee : Threat Intelligence* [online]. c2003 - 2011. W32/Bagz.a@MM. Dostupné z WWW: <http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=128692>.

McAfee, Inc. McAfee Virus Info [online]. c2003-2011. Virus Profile: W32/Renocide Threat Search. Dostupné z WWW: <http://home.mcafee.com/VirusInfo/VirusProfile.aspx?key=153535#>.

Microsoft Corporation. Microsoft Security Intelligence Report : Battling Botnets for Control of Computers. *Security Intelligence Report* [online]. 2010. Dostupný z WWW: <http://www.microsoft.com/security/sir/>.

OLLMANN, Gunter Botnet Communication Topologies: Understanding the intricacies of botnet Command-and-Control. 2009. s. 9. Dostupné z WWW: <http://www.technicalinfo.net/papers/BotnetCommunicationTopologies.html>.

POŠVIC, Kamil. *Root.cz* [online]. 2011. Počet uživatelů internetu v roce 2010 překročil 2 miliardy. Dostupné z WWW: <http://www.root.cz/zpravicky/pocet-uzivatelu-internetu-v-roce-2010-prekrocil-2-miliardy/>.

REED, David; OIKARINEN, Jarkko. *IRC RFC 1459*. 1993

Sandboxie [online]. c2004-2011. Sandboxie. Dostupné z WWW: <http://www.sandboxie.com/>.

SCHILLER, Craig A., et al. *Botnets : The killer web app*. Syngress, 2007. 480 s. ISBN 1597491357.

SHAIKH, Alan. *Botnet Analysis and Detection System*. Edinburgh, 2010. 86 s. Bakalářská práce. Edinburgh Napier University.

The Honeynet project. *Honeypots : Tracking Hackers* [online]. 2003. Honeypots: Definitions and Value of Honeypots. Dostupné z WWW: <http://www.tracking-hackers.com/papers/honeypots.html>.

The Honeynet project. *The Honeynet project: Know your enemy : Tracking Botnets* [online]. 2008. What Bots Do and How They Work. Dostupné z WWW: <http://www.honeynet.org/node/54>.

The Honeynet project. *The Honeynet Project* [online]. 2006. Know Your Enemy: Honeynets. Dostupné z WWW: <http://old.honeynet.org/papers/honeynet/>.

The Shadowserver Foundation. *Shadowserver* [online]. 2005. Botnet Detection. Dostupné z WWW: <http://www.shadowserver.org/wiki/pmwiki.php/Information/BotnetDetection>.

Přílohy

Tabulka 1 – Implicitní uživatelská jména zkoušená RBotem (SCHILLER, C. A., et al. 2007)

Seznam 1 – Příklady příkazů pro botnet od bot-mastera (Microsoft Corporation 2010)

Obrázek 1 – Jednoduchá topologie počítačové sítě s N-IDS Snort.

Předloha: (GULLETT, D. 2010)

Obrázek 2 – Celý výpis běžících procesů naslouchajících na síti příkazem *lsof -i |grep dionaea,* citlivé údaje byly z obrázku odstraněny

Obrázek 3 - Celý výpis otevřených portů ve spojení s procesy příkazem *netstat –ano –tcp -p,* citlivé údaje byly z obrázku odstraněny

Tabulka 2 – Nejčastější IP adresy, z kterých bylo útočeno na honeypot

Tabulka 3 – Tabulka počtu útoků v průběhu dne

Tabulka 4 – Seznam 50 nejpoužívanějších hesel pro přihlášení ke službě MSSQL

Tabulka 1

Administrator	student
Administrador	teacher
Administrateur	wwwadmin
administrat	guest
admins	default
admin	database
staff	dba
root	oracle
computer	db2
owner	

Seznam 1

.capture. Generates and saves an image or video file. Depending on the parameters used, this file could be a screenshot of the victim's desktop or a still image or video from the victim's webcam. The operator can recover the saved picture using the **.get** command.

.ddos.syn, .ddos.ack, .ddos.random. Launches a DDoS attack on a specified IP address for a specified length of time.

.download. Downloads a file from a specified URL to the victim's computer and optionally executes it.

.findfile. Searches for files on the victim's computer by name and returns the paths of any files found.

.getcdkeys. Returns product keys for software installed on the victim's computer.

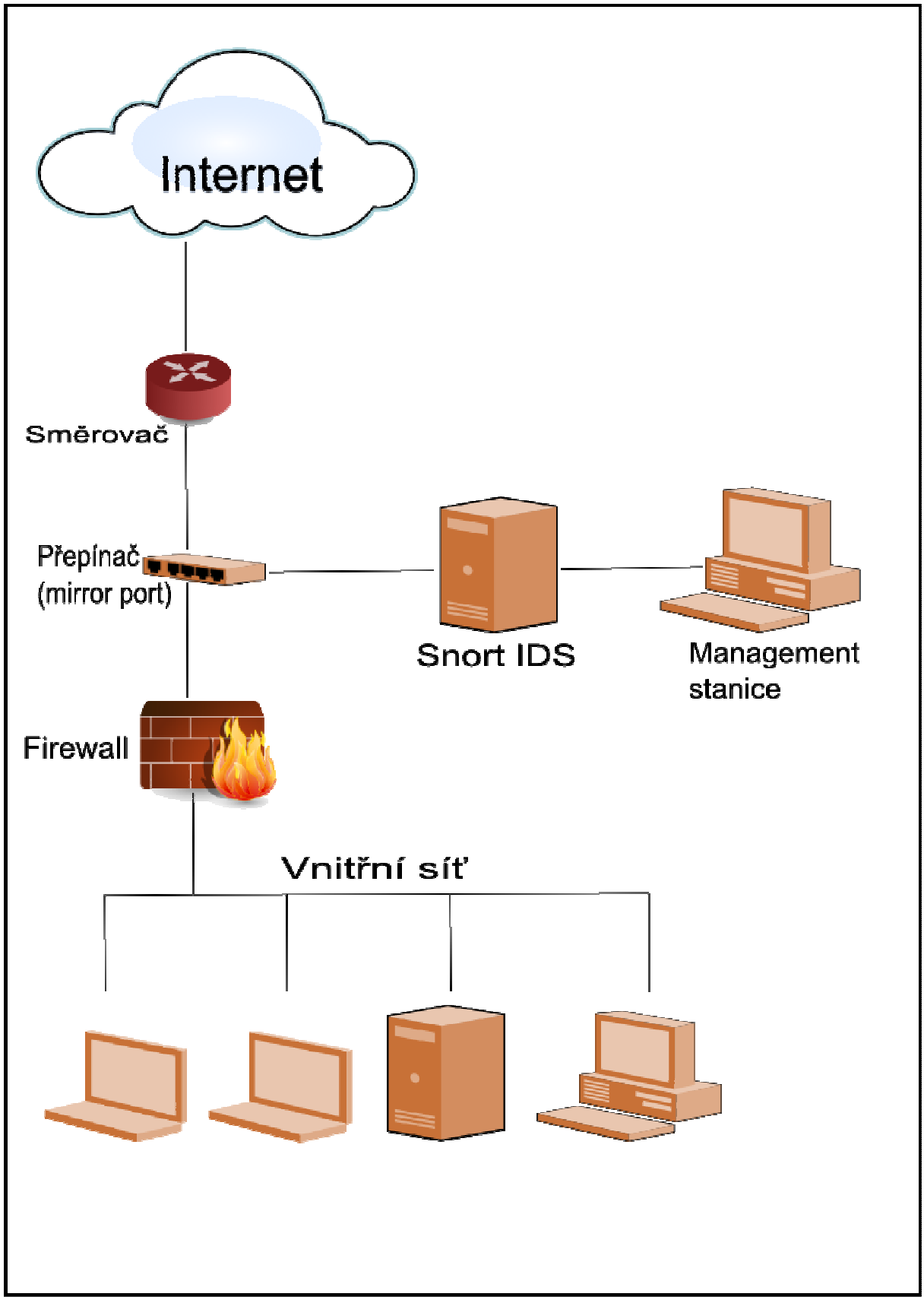
.keylog. Logs the victim's keystrokes and saves them to a file.

.login, .logout. Authenticates the bot-herder with the bots. Before issuing commands to any bots in the channel, the bot-herder must use the **.login** command with a password that is specified in the bots' configuration data so the bots will recognize the bot-herder as an authorized controller.

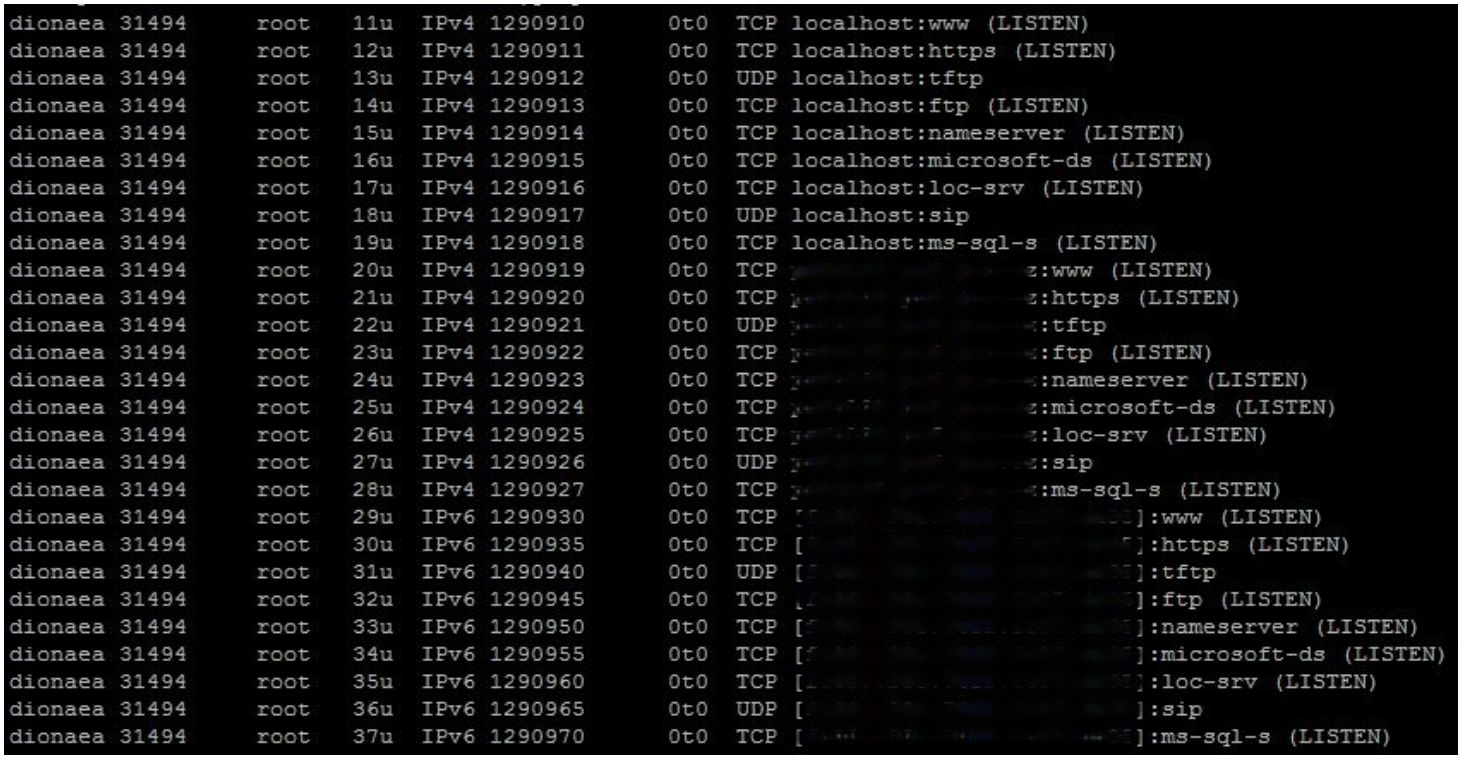
.open. Opens a program, an image, or a URL in a web browser.

.procs. Lists the processes running on the victim's computer. Other commands can then be used to kill processes by name or ID.

Obrázek 1

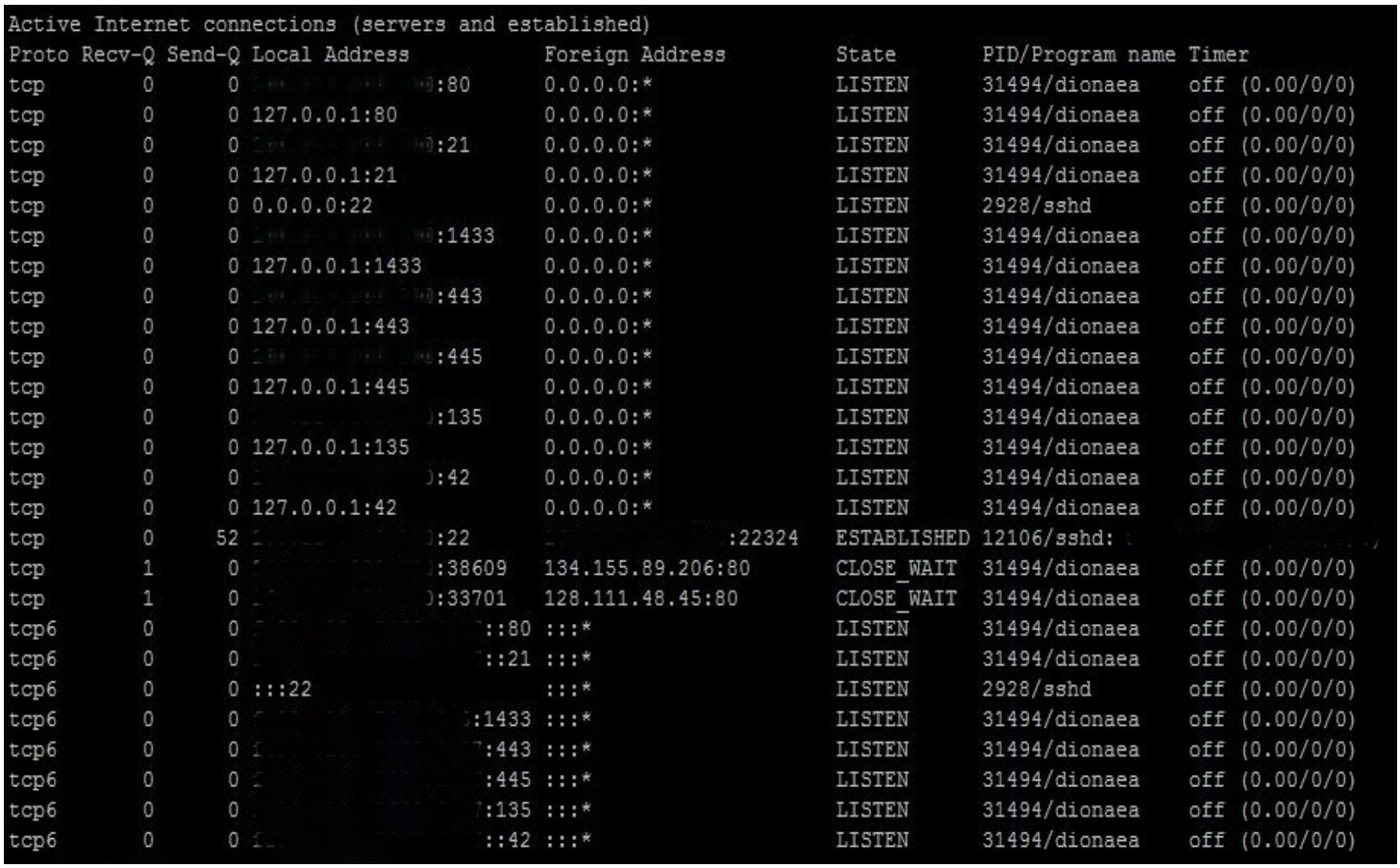

Internet
Směrovač
Přepínač
(mirror port)
Snort IDS
Management
stanice
Firewall
Vnitřní síť

Obrázek 2



```
dionaea 31494     root   11u  IPv4 1290910      0t0  TCP localhost:www (LISTEN)
dionaea 31494     root   12u  IPv4 1290911      0t0  TCP localhost:https (LISTEN)
dionaea 31494     root   13u  IPv4 1290912      0t0  UDP localhost:tftp
dionaea 31494     root   14u  IPv4 1290913      0t0  TCP localhost:ftp (LISTEN)
dionaea 31494     root   15u  IPv4 1290914      0t0  TCP localhost:nameserver (LISTEN)
dionaea 31494     root   16u  IPv4 1290915      0t0  TCP localhost:microsoft-ds (LISTEN)
dionaea 31494     root   17u  IPv4 1290916      0t0  TCP localhost:loc-srv (LISTEN)
dionaea 31494     root   18u  IPv4 1290917      0t0  UDP localhost:sip
dionaea 31494     root   19u  IPv4 1290918      0t0  TCP localhost:ms-sql-s (LISTEN)
dionaea 31494     root   20u  IPv4 1290919      0t0  TCP                 z:www (LISTEN)
dionaea 31494     root   21u  IPv4 1290920      0t0  TCP                 z:https (LISTEN)
dionaea 31494     root   22u  IPv4 1290921      0t0  UDP                 :tftp
dionaea 31494     root   23u  IPv4 1290922      0t0  TCP                 :ftp (LISTEN)
dionaea 31494     root   24u  IPv4 1290923      0t0  TCP                 :nameserver (LISTEN)
dionaea 31494     root   25u  IPv4 1290924      0t0  TCP                 z:microsoft-ds (LISTEN)
dionaea 31494     root   26u  IPv4 1290925      0t0  TCP                 z:loc-srv (LISTEN)
dionaea 31494     root   27u  IPv4 1290926      0t0  UDP                 z:sip
dionaea 31494     root   28u  IPv4 1290927      0t0  TCP                 :ms-sql-s (LISTEN)
dionaea 31494     root   29u  IPv6 1290930      0t0  TCP                 ]:www (LISTEN)
dionaea 31494     root   30u  IPv6 1290935      0t0  TCP                 ]:https (LISTEN)
dionaea 31494     root   31u  IPv6 1290940      0t0  UDP                 ]:tftp
dionaea 31494     root   32u  IPv6 1290945      0t0  TCP                 ]:ftp (LISTEN)
dionaea 31494     root   33u  IPv6 1290950      0t0  TCP                 ]:nameserver (LISTEN)
dionaea 31494     root   34u  IPv6 1290955      0t0  TCP                 ]:microsoft-ds (LISTEN)
dionaea 31494     root   35u  IPv6 1290960      0t0  TCP                 ]:loc-srv (LISTEN)
dionaea 31494     root   36u  IPv6 1290965      0t0  UDP                 ]:sip
dionaea 31494     root   37u  IPv6 1290970      0t0  TCP                 ]:ms-sql-s (LISTEN)
```

Obrázek 3



```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name Timer
tcp        0      0                 :80      0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0 127.0.0.1:80             0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0                 :21      0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0 127.0.0.1:21             0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0 0.0.0.0:22               0.0.0.0:*               LISTEN      2928/sshd        off (0.00/0/0)
tcp        0      0                 :1433    0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0 127.0.0.1:1433           0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0                 :443     0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0 127.0.0.1:443            0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0                 :445     0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0 127.0.0.1:445            0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0                 :135     0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0 127.0.0.1:135            0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0                 :42      0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0      0 127.0.0.1:42             0.0.0.0:*               LISTEN      31494/dionaea    off (0.00/0/0)
tcp        0     52                 :22                  :22324     ESTABLISHED 12106/sshd:
tcp        1      0                 :38609   134.155.89.206:80       CLOSE_WAIT  31494/dionaea    off (0.00/0/0)
tcp        1      0                 :33701   128.111.48.45:80        CLOSE_WAIT  31494/dionaea    off (0.00/0/0)
tcp6       0      0                 ::80 :::*                        LISTEN      31494/dionaea    off (0.00/0/0)
tcp6       0      0                 ::21 :::*                        LISTEN      31494/dionaea    off (0.00/0/0)
tcp6       0      0 :::22                    :::*                    LISTEN      2928/sshd        off (0.00/0/0)
tcp6       0      0                 :1433 :::*                       LISTEN      31494/dionaea    off (0.00/0/0)
tcp6       0      0                 :443 :::*                        LISTEN      31494/dionaea    off (0.00/0/0)
tcp6       0      0                 :445 :::*                        LISTEN      31494/dionaea    off (0.00/0/0)
tcp6       0      0                 :135 :::*                        LISTEN      31494/dionaea    off (0.00/0/0)
tcp6       0      0                 ::42 :::*                        LISTEN      31494/dionaea    off (0.00/0/0)
```

Tabulka 2

Ve sloupci COUNT jsou počty útoků odpovídající IP adresám ve sloupci repote_host.

COUNT	remote_host
173	119.194.170.15
110	1.202.139.80
110	200.161.82.63
93	210.51.37.80
44	84.33.199.44

Tabulka 3

Ve sloupci hour jsou hodiny a k nim odpovídající počty útoků ve sloupci COUNT.

hour	COUNT	hour	COUNT
1.0	31	14.0	2008
2.0	138	15.0	42
3.0	16	16.0	35
4.0	20	17.0	59
5.0	41	18.0	1040
6.0	31	19.0	204
7.0	54	20.0	17
8.0	14	21.0	17
9.0	30	22.0	22
10.0	75	23.0	27
11.0	2307	24.0	33
12.0	1036	-	-
13.0	1903	-	-

Tabulka 4

Slovem *null* je označeno prázdné heslo.

null	asdfgh	!@#$%	123123	sa123	111111	123qwe	9876543210
admin	passwd	!@#$%^	1q2w3e	%null%	121212	168168	x
server	password	!@#$%^&	hjkl	000000	123321	1q2w	x
sql	root	!@#$%^&*	jkl	001	123456789	1q2w3e4r	x
12345	111	1234	administrator	002	1234qwer	1qaz	x
123456	123	654321	database	003	123abc	1qaz2wsx	x
asdf	!@#$	1	sa	007	123asd	321	x