Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra Informačních Technologií

Principy honeypotů a jejich využití pro zabezpečení síťového provozu

Principles of honeypots and their use to secure network traffic

Diplomová práce

Autor: Bc. Ondřej Líbal
Studijní obor: Informační Management

Vedoucí práce: Mgr. Josef Horálek, Ph.D.

Hradec Králové Říjen 2020

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 13.11.2020

Poděkování:

Děkuji vedoucímu diplomové práce Mgr. Josefu Horálkovi, Ph.D. za metodické vedení práce, podnětné připomínky a nápady pro moji práci.

Anotace

Diplomová práce s názvem Principy honeypotů a jejich využití pro zabezpečení síťového provozu se snaží vysvětlit fungování a principy honeypotů, jejich rozdíly a jejich výhody nebo nevýhody při nasazení. Cílem práce je ukázat, jak nainstalovat, nakonfigurovat a otestovat určité vybrané honeypoty, tak aby plnili svůj účel. Výsledkem práce je pak zhodnocení vybraných honeypotů, ukázka, jak vypadá honeypot pod útokem a jak se chová a také jak si na honeypot zaútočit sám pro otestování správné funkčnosti.

Annotation

Title: Principles of honeypots and their use to secure network traffic

Diploma thesis named Principles of honeypots and their use to secure network traffic with an explanation of the operation and principles of honeypots, their differences and advantages or disadvantages in deployment. The aim of this thesis is to show how to modify, set and test certain selected honeypots, so that they fulfil their purpose. The result of thesis is the evaluation of selected honeypots, demonstration how honeypot looks like under attack and how it behaves and how to attack honeypot by myself to test the correct functionality.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Tato diplomová práce se zabývá principy honeypotů a jejich využití v praxi pro bezpečnost síťového provozu. V dnešní době je totiž velmi těžké se bránit všem druhům útoků, ale existují antivirové programy, které dokáží detekovat malware prostřednictvím signatur, ale už není jednoduché zjistit, jakou činnost daný malware na stanici prováděl. Stejně tak je to i s útočníky. Snažíme se jim bránit, aby nenapadli náš systém, ale už nevíme, pokud ho napadnou, co, kde a jak konají. Z tohoto důvodu existují honeypoty, které dokáží sledovat chování ať už malwaru nebo samotného útočníka a na venek se tváří jako reálný systém, aby si útočník myslel, že je opravu v reálném systému. Honeypoty jsou také velmi efektivní, co se sběru dat týče oproti detekci infikovaných konečných stanic, protože honeypoty umožnují sběr velkého množství vzorků. Důležité však je vždy honeypot správně nastavit, aby se nestal naopak bezpečnostní dírou pro útok. S tím je spojené vhodné umístění honeypotu do sítě. Honeypotů existuje velké množství, dle druhu využití, a proto je možné různé honeypoty kombinovat, ale samozřejmě je možné je používat i samostatně.

2 Cíl práce

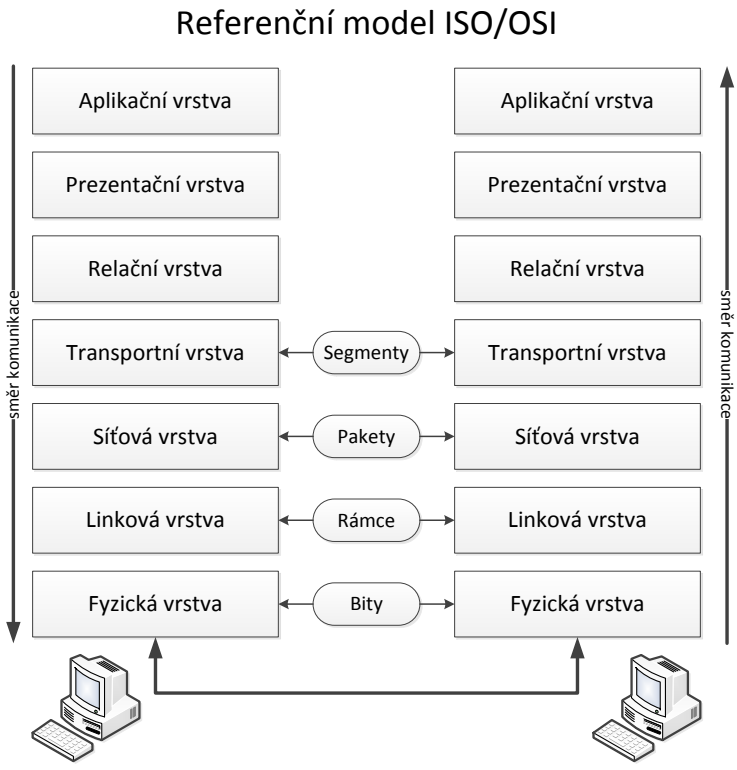
Smyslem a cílem této práce je ukázat, jak honeypoty fungují, jaké typy honeypotů existují, jaké typy útoků existují a jaké bezpečností rizika a benefity lze díky honeypotům získat nebo naopak jaké hrozby mohou nastat, pokud jsou honeypoty špatně nastaveny. Také by práce měla ukázat, jak nainstalovat, nakonfigurovat, otestovat funkčnost honeypotů a jak je následně nasadit do síťového provozu. Díky možnostem pronajmout si veřejně dostupný server, by práce měla ukázat, jak velkému množství útoků jsou dnes a denně veškerá síťová zařízení vystavována.

Práce by měla být rozdělena do dvou částí, kdy za úkol je ukázat, jak si nainstalovat za pomoci MHN (Modern Honey Network) honeypot, kde lze můžeme k nasbíraným datům přistupovat graficky. V druhé části by měla ukázat, jak nainstalovat standalone verzi honeypotu a následně ji nakonfigurovat pro správný běh. U standalone verze by mělo být ukázáno, jak lze na honeypot zaútočit a následně jak takový útok v honeypotu detekovat.

Tuto diplomovou práci jsem si vybral, protože si myslím, že ponětí o tom, co je to honeypot je velmi malé a chtěl jsem poukázat na to, že tento nástroj pro zabezpečení sítě existuje a že může být dobrým pomocníkem proti útokům, které jsou na denním pořádku.

3 Model ISO/OSI

Nejprve si představíme referenční model ISO/OSI, který je vhodné znát a je v sítích využíván. *Tento model dle* (1) *OSI (Open System Interconnection) byl představen organizací ISO (International Standard Organization). Není to zcela tak úplně protokol, ale spíše model či standard, který je založen na konceptu vrstvení.* Je to model, který je založen na 7 referenčních vrstvách, kdy každá vrstva má své funkce a specifika. Při postupu dat skrze ISO/OSI model se vždy postupuje zezdola nahoru. Standard ISO/OSI byl zaveden, aby bylo možné v počítačových sítích používat různé síťové technologie a operační systémy (2). Celkově můžeme rozdělit tento model na 3 části. První část se zabývá fyzickým přenosem, a to na fyzické, linkové a síťové vrstvě. Druhá část se zabývá zpracováním dat, a to na relační, prezentační a aplikační vrstvě. Poslední část je propojovací mezi první a druhou a leží přesně mezi nimi. Tato vrstva se jmenuje transportní vrstva. První část zařizuje tedy spíše hardware, kdežto druhou část zařizuje software. Níže si blíže rozebereme jednotlivé vrstvy.



Obrázek 1 Referenční model ISO/OSI (vlastní zpracování)

3.1 Fyzická vrstva (Physical layer)

Nejnižší vrstva ISO/OSI modelu je fyzická vrstva, která zajišťuje fyzický přenos dat v podobě bitů. Specifické pro tuto vrstvu je využívání elektrických signálů, které definují 0 a 1, druhy kabelů nebo typy konektorů a další. Zjednodušeně je to vrstva, která je odpovědná za přenos a příjem nestrukturovaných nezpracovaných dat po síti. Tato vrstva také provádí kódování dat. Podrobnější význam vrstvy popisuje (3).

3.2 Linková vrstva (Data link layer)

Tato vrstva také bývá občas nazývána jako spojová vrstva. V této vrstvě jsou data spojována do tzv. rámců, kdy vrstva musí poznat začátek a konec takového rámce a také kontrolovat jejich správnost pomocí CRC součtu (4). Musí také zařídit správnost odeslaných a přijatých rámců a v případě chyby musí opětovně zaslat rámec znovu. Také je jejím úkolem signalizovat, aby se vysílací uzel zastavil, pokud je vyrovnávací paměť plná a příjemce již nemůže rámce nadále přijímat. Rámec v této vrstvě je složen ze záhlaví a zápatí, kde je obsažena tzv. MAC adresa odesílatele a příjemce. Obecně hlavní funkcí této vrstvy je zajistit, aby přenos dat byl bezchybný z jednoho uzlu do druhého přes fyzickou vrstvu. Podrobněji vrstvu rozebírá (5).

3.3 Síťová vrstva (Network layer)

Data v této vrstvě jsou členěna do tzv. paketů. Na rozdíl od linkové vrstvy, kdy rámce obsahovali MAC adresu pro směrování, u paketů je to IP adresa obou účastníků komunikace a další informace jako potvrzení o doručení a řízení komunikačního toku. Hlavním úkolem síťové vrstvy je zajistit doručení paketů, kdy je vyhledána vhodná cesta v případě nepřímého spojení. Zajišťuje také směrování v uzlech mezi odesílatelem a příjemcem. Rozděluje odchozí zprávy do paketů a sestavuje příchozí pakety do zpráv pro vyšší úrovně modelu ISO/OSI (6). Podrobněji vrstvu popisuje (3).

3.4 Transportní vrstva (Transport layer)

„Tato vrstva se zabývá rozdělením balíku dat do jednotlivých paketů, které pak síťová vrstva posílá směrem k příjemci nebo naopak k sestavení přijatých paketů dohromady. Jedním z úkolů, které tato vrstva zajišťuje, je vyrovnání rozdílů mezi síťově orientovanými spodními vrstvami a aplikačně orientovanými vyššími vrstvami.“ (4) Umí také rozpoznat a někdy i opravit detekované chyby. Stejně tak se stará o sestavení paketů do správného pořadí, pokud dorazí v jiném pořadí, a to díky tomu, že jsou pakety číslované. Podrobněji je vrstva popsána na (7).

3.5 Relační vrstva (Session layer)

Relační vrstva se stará hlavně o tzv. relaci, což je doba, po kterou spolu jednotlivé uzly komunikují. Tato vrstva se tedy stará o navázání, řízení, rušení spojení a rozhodování o typu spojení. Zda se bude jednat o half-duplex nebo full-duplex spojení. Rozhoduje též o tom, zda se bude jednat o šifrovaný přenos dat. Podrobněji je vrstva popsána na webu (8).

3.6 Prezentační vrstva (Presentation layer)

„Prezentační vrstva se stará o to, aby byla data odesílána tak, aby příjemce porozuměl datům, respektive informacím, které data skrývají a byl schopen je používat. Prezentační vrstva také při přijímání dat transformuje data tak, aby byla připravena pro aplikační vrstvu.“ (6) Prezentační vrstva hraje také roli překladatele, a to ve smyslu, že přijatá data musí být převedena do takového formátu, aby cílová stanice mohla předat data správné aplikaci. Řeší tedy kompresi a kódování dat. Podrobněji je vrstva vysvětlena na webu (9).

3.7 Aplikační vrstva (Application layer)

Tato vrstva je poslední a také nejvyšší vrstvou referenčního modelu ISO/OSI. Za úkol má poskytování služeb aplikacím. *Podle* (4) *takovým příkladem služby aplikační vrstvy jsou například mechanismy pro přenos elektronické pošty, ale nikoliv jeho rozhraní.* Rozhraní je pak posunuto dále na aplikační vrstvu. Aplikační vrstva je jediná vrstva, ke které má vlastně uživatel přístup. Podrobněji je vrstva rozebrána na (7).

4 Zabezpečení síťového provozu

V této kapitole se budeme zabývat zabezpečením síťového provozu, jaké nástroje pro zabezpečení využít a jak fungují. Jelikož zabezpečení obecně je stále více aktuální téma, je nutné mu věnovat vcelku značnou pozornost. Je důležité pro mnoho firem, aby byli v bezpečí proti různým útokům, ať už se jedná o útoky na data společnosti nebo útoky na zdroje, jako jsou různá síťová zařízení. Důležité je si na začátku uvědomit, jak je velká firma a jaké nástroje pro zabezpečení by měla tedy využít. Některé nástroje pro zabezpečení totiž nejsou levnou záležitostí a pro menší firmy mohou být velmi nákladné, někdy i zbytečné. Veškerá zabezpečení, která si zde ukážeme níže, jsou koncipována převážně pro velké společnosti, ale mohou je využít samozřejmě i menší společnosti.

4.1 Firewall

Firewall je jedním ze základních ochranných mechanismů sítě a je též označován perimetrem obrany sítě. Funkcí firewallu je omezovat odchozí a příchozí provoz sítě. Firewallem tedy prochází veškerá příchozí a odchozí komunikace a leží na pohraničí vnitřní a vnější sítě. Firewall může být jak hardwarový, tak softwarový nebo kombinace těchto dvou platforem. Firewally se dělí do tří základních kategorií, a to nestavové firewally, stavové firewally a aplikační firewally. V dnešní době však firewall bývá kombinací všech tří kategorií, a to z důvodu větší bezpečnosti, funkčnosti a lepšího výkonu. Novou skupinou pak jsou tedy firewally nové generace (NGFW).

4.1.1 Nestavový firewall (Packet Filter Firewall)

Nestavový firewall sleduje síťový provoz a omezuje nebo blokuje pakety na základě zdrojových a cílových adres nebo jiných statických hodnot. K tomu slouží pravidla pro filtrování paketů, a pokud nejsou pravidla shody naplněny, pak filtr brány firewall bude pakety blokovat. *Nestavový firewall používá jednoduché sady pravidel, které nepředpokládají, že by firewall mohl obdržet paket, který předstírá, že má být vyžádán. Takže staticky vyhodnocuje obsah paketů a nevede přehled o stavu síťových připojení.* (10) *Výhodou nestavového firewallu je jeho operační rychlost a pružnost konfigurace. Naopak jejich slabinou je neschopnost zabránit útokům, které*

využívají potenciálně zranitelná místa konkrétních aplikací. (11) Nestavový firewall operuje zejména na L3 ISO/OSI modelu.

4.1.2 Stavový firewall (Stateful Packet Inspection Firewall)

Stavový firewall je firewall, který monitoruje celkový stav aktivních síťových připojení. To znamená, že stavové firewally neustále analyzují úplný kontext provozních a datových paketů. Jakmile je tedy určitý typ provozu schválen stavovým firewallem, je přidán automaticky do tabulky stavů a může volně cestovat do privátní sítě. *Konkrétně jsou zaznamenávány například informace o IP adrese a zdrojovém a cílovém portu. Také prověřuje určité hodnoty v záhlaví přenášených paketů a sleduje stav jednotlivých připojení po určitou vymezenou dobu.* (11) Díky tomu dokáže stavový firewall realizovat mnohem vyšší míru zabezpečení. *Přesto však má stavový firewall i nějaké zranitelnosti. Jednou z nich je například, že stavový firewall může být napaden útoky typu DDoS, což pro stavový firewall představuje velmi velkou náročnost na výpočetní výkon.* (12) Stavový firewall operuje také hlavně na L3 ISO/OSI modelu.

4.1.3 Web Aplikační firewall – WAF (Web Application Firewall)

Web Aplikační firewall je kombinací klasických stavových kontrolních technologií a také schopností provádět hloubkovou kontrolu aplikací. Zahrnuje v sobě možnost analýzy protokolů na L7 ISO/OSI modelu tedy na aplikační vrstvě, jako je například u protokolů FTP (File Transfer Protocol) nebo HTTP (Hyper Text Transfer Protocol). Monitorováním provozu lze porovnávat neškodné aktivity protokolů s aktuální aktivitou daného protokolu a tím identifikovat potencionální odchylky v jejich chování, které mohou znamenat potenciální útok (11). Díky těmto vlastnostem je umožněno firewallu zakázat nebo povolit přístup aplikaci k dostupným síťovým prostředkům v závislosti, jak se aplikace chová v síti. Web aplikační firewall operuje na L3 až L7 ISO/OSI modelu.

4.1.4 Firewall nové generace – NGFW (Next–Generation Firewall)

Podle společnosti Gartner je firewall nové generace definován jako „integrovaná síťová platforma, která provádí hloubkovou kontrolu provozu a

blokování útoků.“ (13) Obecně firewall nové generace v podstatě poskytuje služby nad rámec stavového firewallu. Jsou to například funkce jako je povědomí o aplikacích a jejich ovládání, integrovaná prevence narušení, hrozby cloudové inteligence, odposlechy SSL a SSH, filtrování webových stránek, správa kvality služeb a další. Nevýhodou těchto firewallů je, že využívají většinou samostatné interní moduly, pro zajištění jednotlivých bezpečnostních funkcí, což znamená, že analyzovaný paket může být podroben několikanásobnému procesu posuzování, než je oprávněn pro vstup do sítě nebo naopak zablokován. Firewall nové generace zahrnuje více vrstev z ISO/OSI modelu, až po L7.

4.2 Data Loss Prevention (DLP)

Prevence ztráty dat je sada nástrojů a procesů používaných k zajištění, aby nedocházelo ke ztrátě, zneužití nebo přístupu citlivých dat neoprávněnými uživateli mimo podnikovou síť (14). Další definice dle (15) zní: Je to systém, který slouží k monitorování uživatelů a procesů, jak pracují s citlivými (klasifikovanými) daty společnosti a má možnost i blokovat nežádoucí aktivity a komunikace, sledovat nebo blokovat připojená zařízení přes sběrnice počítače a následně reportovat zjištěné incidenty na centrální konzoli nebo do SIEM (Security Information and Event Management). Software DLP klasifikuje regulovaná, důvěrná, obchodní kritická data a identifikuje porušení zásad definovaných organizacemi, obvykle řízeného dle předpisů jako je například GDPR (General Data Protection Regulation) nebo HIPAA (Health Insurance Portability and Accountability Act) a další. Pokud DLP detekuje porušení, vynucuje nápravu pomocí výstrah, šifrování a dalších ochranných opatření, aby se zabránilo koncovým uživatelům v náhodném nebo zlomyslném sdílení dat, která by mohla organizaci ohrozit.

Pokud by se například zaměstnanec pokusil přeposlat obchodní e-mail mimo firemní doménu nebo nahrát podnikový soubor do cloudové služby pro zákazníky například DropBox, byl by zaměstnanci odepřen souhlas. Správce sítě může díky softwarovým produktům řídit, jaká data mohou koncoví uživatelé přenášet. (16)

DLP lze dělit na Network DLP systémy a Host DLP systémy. Network DLP systémy jsou většinou dedikované fyzické nebo virtuální appliance, které se

umisťují na perimetr společnosti a kontrolují emailový provoz, HTTP, HTTPS a FTP (15). Nasazení takových DLP je snazší, ale nezasahují do zařízení uživatelů ve společnosti, ale dokážou monitorovat veškeré uživatelské aktivity. Naopak Host DLP mají vždy softwarového agenta, který se instaluje na koncové zařízení, většinou se jedná o pracovní stanice a notebooky uživatelů, ale jsou podporované i instalace na servery (15). Implementace takových DLP je časově náročnější. *Výhodou však je, že lze monitorovat koncového uživatele i v offline režimu, kdy je zařízení mimo organizaci. V případě tohoto DLP je také nutné mít nějakou centrální správu systému, jelikož se instalace provádí na koncová zařízení.* (15)

4.3 Virtual Private Network (VPN)

Jak již název říká, jedná se o virtuální síť, která je soukromá, takže se mohou uživatelé připojit k internetu nebo podnikové síti způsobem, který je zabezpečený a šifrovaný. Hlavním cílem VPN je v podstatě zachovat bezpečnost a soukromí informací uživatele. *Definice VPN podle* (17) *je: Virtuální privátní síť (VPN) je šifrované připojení přes internet ze zařízení do sítě. Šifrované připojení pomáhá zajistit bezpečný přenos citlivých dat. Zabraňuje neoprávněným osobám odposlouchávat provoz a umožňuje uživateli provádět práci na dálku. Technologie VPN je široce používána v podnikových prostředích.* VPN tedy funguje tak, že data jsou nejprve zašifrována na našem počítači skrze VPN klienta a potom se tunelem odešlou na vybraný VPN server. Server data dešifruje a přepošle cílovému serveru. Opačný provoz pak funguje stejně. VPN fungují hlavně na vrstvách L2 a L3 ISO/OSI modelu.

4.3.1 Site-to-site VPN

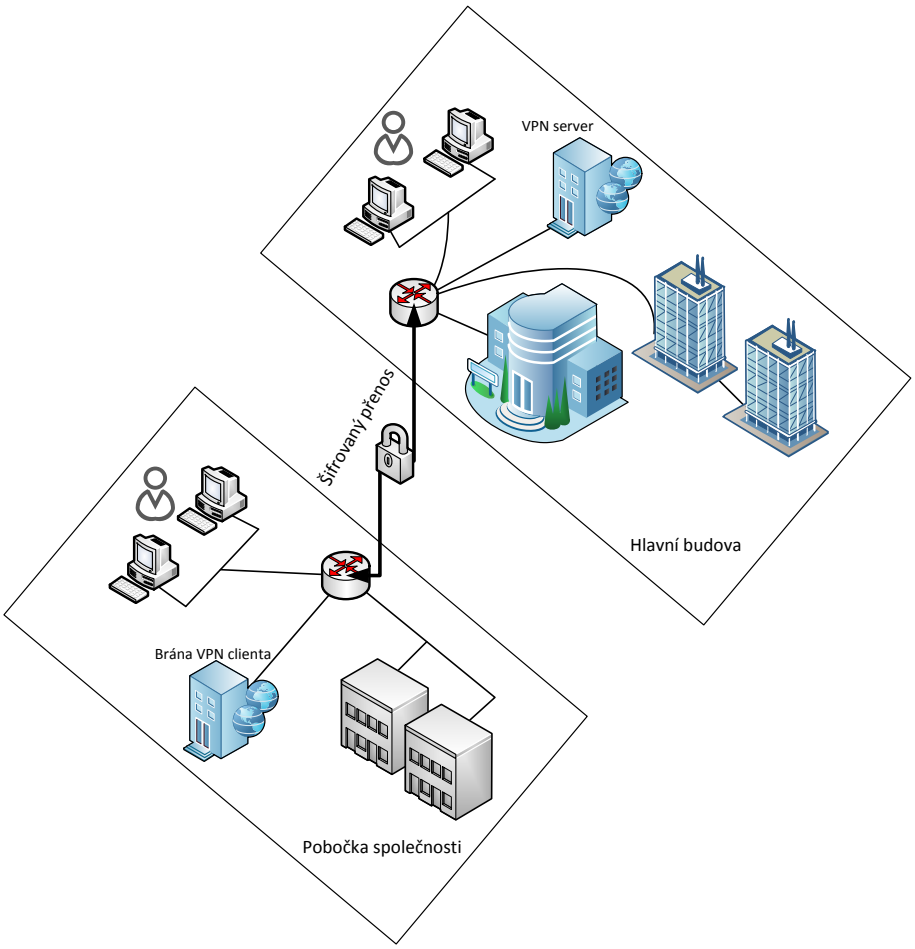
Site-to-site VPN spojuje dvě a více míst, která jsou od sebe vzdálená a nelze je spojit pomocí fyzického propojení. Například pokud máme hlavní budovu firmy a potřebujeme ji spojit s dalšími pobočkami této společnosti, využijeme právě site-to-site VPN, kdy komunikace mezi pobočkami probíhá přes internet. *K vytvoření a udržování spoje se používá jednoúčelové zařízení. Je to v podstatě, jako bychom přistupovali fyzicky ze sítě do sítě, ale komunikace probíhá šifrovaně skrze internet.* (17) Brána VPN ve vzdálené síti se musí spojit s druhou bránou VPN, která je

v hlavní společnosti, a to za účelem vytvoření bezpečného tunelu. *Na rozdíl od VPN vzdáleného přístupu zařízení nepotřebují klienta VPN, ale v podstatě odesílají běžný provoz přes brány VPN.* (18)

Site-to-site VPN můžeme rozdělit na dva typy:

Intranet-based – Pokud společnost má jednu nebo více vzdálených poboček, které chce připojit k jediné soukromé síti, může vytvořit tzv. intranetovou VPN. Každá pobočka se pak může připojit k interní síti společnosti.

Extranet-based – Pokud má společnost úzký vztah s jinou společností jako je například partner, dodavatel nebo zákazník, může vybudovat tzv. extranetovou VPN. Taková VPN umožňuje společnostem spolupracovat v bezpečném sdíleném síťovém prostředí, ale zároveň zabraňuje přístup k interní části.



Obrázek 2 Site-to-site VPN (vlastní zpracování)

4.3.2 Remote access VPN

VPN vzdáleného přístupu bezpečně připojuje uživatele mimo podnikovou síť. Tyto zařízení jsou známé pod pojmem jako koncové body a mohou to být například notebooky, chytré telefony nebo i tablety. Díky technologiím je možné provést bezpečnostní kontroly v koncových bodech ještě před tím, než jim je přístup do firemní sítě povolen. (17)

Pro navázání vzdáleného přístupu skrze VPN potřebujeme server NAS (Network Access Server) nebo bránu VPN, které ověří přihlašovací údaje jakéhokoliv zařízení, které se pokouší přihlásit do sítě VPN. Také potřebujeme na zařízení klientský software, který komunikuje s bránou VPN, který zařízení autentizuje pro vzdálený přístup a vytváří tak zabezpečený „virtuální" tunel. (18)

Příkladem může být například pracovník, který má home office nebo pracuje v terénu a potřebuje tedy pracovat z místa mimo společnost, ale potřebuje být připojen do firemní sítě. Takový pracovník spustí ve svém notebooku klienta VPN, který nastaví tunelové připojení k NAS nebo bráně VPN a také zaručí šifrování potřebné k zabezpečenému připojení. Následně je vytvořen tunel do firemní sítě společnosti a tím získá pracovník plnohodnotný přístup do firemní sítě, jako by tam seděl.

4.4 Advanced Malware Protection (AMP)

Tento pojem není až zas tak starý, takže přesný pojem, co obnáší je těžko specifikovatelné. Každá firma totiž pod AMP nabízí trochu něco jiného, ale některé komponenty jsou vždy v nabídce stejné. Většinou tedy obsahují prevenci proti škodlivému softwaru, implementaci pro více útočných směrů / vstupních bodů jako (firewall, síť, koncový bod, e-mail) a retrospektivní varování a nápravu. AMP operuje na několika vrstvách ISO/OSI modelu, a to od L1 až po L5.

4.4.1 Malware Threat Prevention

Prevence ohrožení malwarem je základní funkčností řešení kybernetické bezpečnosti. Velkou sílu má v prevenci proti malwaru v dnešní době cloud, dále je to rychlá a bezproblémová aktualizace virové inteligence a automatizovaný sandboxing. (19) Nyní si je trochu podrobněji rozebereme.

Cloudová kybernetická bezpečnost

Velcí dodavatelé jako jsou Cisco a Check Point a další podle (20) *jako třeba Microsoft, McAfee, Netskope, Symantec nebo Bitglass* jsou schopni díky velkému množství instalací na koncových zařízeních sbírat i velké množství dat celosvětově. Takže čím více údajů o kybernetických útocích mají prodejci, tím lépe se mohou proti nim bránit. V tom je tedy síla cloudu.

Rychlé a plynulé nasazení

Jelikož sběr dat je již zařízen, nyní je nutné údaje o kybernetických útocích zpracovat a opětovně doručit na koncová zařízení, aby mohli být právě chráněny proti nejnovějším kybernetickým hrozbám. *Díky propojení s cloudem je AMP schopen rychle a hladce nasadit nápravná opatření k nedávno objeveným hrozbám malwaru, aby došlo k menšímu počtu úspěšného infikování koncových zařízení.* (19)

Automatizovaný sandboxing

Třetí funkce, kterou AMP nabízí je automatizovaný sandboxing pro detekci a blokování malwaru na základě určitých pravidel. Dnešní malware je však navržen tak, aby rychle mutoval a nemohlo dojít k jeho detekci. Přestože jsme díky cloudu a rychlému nasazení záplat vir schopni detekovat a vyléčit, můžeme být znovu infikováni. To ale řeší právě zmiňovaný sandboxing, kdy je virus vložen do karantény, kde se spouští ve virtuálním prostředí a nemůže nic poškodit. Zde se sleduje a vytváří se proti němu ochrana.

4.4.2 Threat Extraction

Jelikož jsou útoky stále více rafinované a jejich počet, každým dnem roste, je důležité se proti nim bránit. (21) *uvádí, že v roce 2013 si dokonce až 84 % společností stáhlo infikovaný dokument.* Threat extraction je tedy specifický druh ochrany, který se zabývá „čištěním“ infikovaných dokumentů. „Technologie Threat Extraction proto preventivně odstraní hrozby rekonstrukcí dokumentů jen se známými bezpečnými prvky a současně extrahuje aktivní obsah, vložené objekty a další zneužitelný obsah. Dokument je následně zrekonstruován bez potenciálních

hrozeb, takže je zajištěn jeho 100 % bezpečný obsah.“ (22) To vše probíhá v rámci několika sekund a je tím zařízen bezpečný obchodní tok. Další výhodou tohoto nástroje je, že zpětně můžeme analyzovat původní infikované soubory a zjistit tak, co se v nich za infikaci skrývalo. Threat extraction funguje na více vrstvách modelu ISO/OSI, ale primárně to jsou L3 a L7.

4.5 Denial of Service (DoS) a Distributed Denial of Service (DDoS)

Útok DDoS je nebezpečný pokus o znepřístupnění online služby uživatelům, obvykle dočasným přerušením nebo pozastavením služeb hostitelského serveru. Útok DDoS je spuštěn z mnoha zařízení, často distribuovaných globálně v tzv. botnetu. Odlišuje se však od útoku typu DoS, protože tento typ útoku používá pouze jediné zařízení s jedním síťovým připojením pro zaplavení cíle škodlivým přenosem. (23) Jelikož existuje mnoho typů útoků, tak se nám tyto útoky prolínají celým spektrem ISO/OSI modelu, jak je vyobrazeno na obrázku 3 níže.

OSI Vrstva	Protocol Data Unit (PDU)	Popis vrstvy	Protokoly	Příklady DoS (Denial of Service) Techniky na každé úrovni	Potenciální dopad DoS útoku	Možnosti pro zmírnění typu útoku
Aplikační vrstva	Data	Začíná zde vytváření zpráv a paketů. Přístup DB je na této úrovni. V této vrstvě fungují protokoly koncových uživatelů, jako jsou FTP, SMTP, Telnes a RAS.	Využití protokolů FTP, HTTP, POP3, SMTP a zařízení je bránou	PDF GET žádosti, HTTP GET, HTTP POST, = webové forumálře (přihlášení, nahrání fotky/videa, odesílání zpětné vazby	Dosáhnutí limitů zdroje, kdy je nedostatek	Monitorování aplikací je praxe monitorování softwarových aplikací pomocí vyhrazené sady algoritmů, technologií a přístupů k detekci útoků nultého dne a aplikační vrstvy (útoky vrstvy 7). Jakmile jsou tyto útoky identifikovány, lze je zastavit a vysledovat zpět ke konkrétnímu zdroji snadněji než jiné typy DDoS útoků.
Prezentační vrstva	Data	Překládá formát dat od odesílatele k příjemci	Používá protokoly komprese a šifrování	Znetvořený SSL požadavek -- Kontrola SSL šifrování paketů je náročná na zdroje. Útočníci vyoužívají SSL k tunelování skrze HTTP na cílený server.	Zasažené systémy mohou přestat přijímat SSL připojení nebo se automaticky restartovat	Chceme-li zmírnit dopady, zvažme možnosti, jako je uvolnění SSL z původní infrastruktury a kontrola provozu aplikace, zda nevykazuje známky útoků, provozu nebo porušení zásad na platformě pro doručování aplikací (ADP). Dobrý ADP také zajistí, že váš provoz je poté znovu zašifrován a předán zpět do původní infrastruktury s nezašifrovaným obsahem, který se kdykoli nachází v chráněné paměti na zabezpečeném bastion hostiteli.
Relační vrstva	Data	Řídí relaci, ukončuje a synchronizuje relaci s OS přes síť. (např.: když se odhlásíme a přihlásíme)	Používá protokol Přihlášení/Odhlášení	Telnet DDoS útočník využívá chybu Telnet serveru a jeho softwaru spuštěném na přepínači, což znemožňuje služby Telnet	Zabraňuje správci v provádění funkcí správy přepínačů	Poradit se s poskytovatelem hardwaru a zjistěte, zda existuje aktualizace verze nebo oprava ke zmírnění této chyby zabezpečení
Transportní vrstva	Segment	Zajišťuje bezchybný přenos mezi hostiteli: spravuje přenos zpráv z vrstev 1 až 3	Používá protokoly TCP a UDP	SYN Flood, Smurf Attack	Dosáhnutí šířky pásma, nebo omezení připojení hostitelů nebo síťových zařízení	Blokování útoku DDoS, běžně označované jako blackholing, je metoda, kterou ISP obvykle používají k zastavení útoku DDoS na jednoho ze svých zákazníků. Tento přístup k blokování útoků DDoS činí dotyčný web zcela nepřístupným pro veškerý provoz, a to jak škodlivý útok, tak legitimní provoz uživatele. Black holding je obvykle nasazen poskytovatelem internetových služeb k ochraně ostatních zákazníků ve své síti před nepříznivými účinky DDoS útoků, jako je pomalý výkon sítě a narušená služba
Síťová vrstva	Paket	Zajišťuje směrování a přepínání informací do různých sítí. LAN nebo interní sítě	Používá protokoly IP, ICMP, ARP a RIP a jako zařízení používá směrovače	ICMP Flooding - metoda útoku na infrastrukturu DDoS vrstvy 3, která využívá zprávy ICMP k přetížení šířky pásma cílové sítě	Může ovlivnit dostupnou šířku pásma sítě a způsobit další zátěž pro firewall	Rychlostně omezit přenos ICMP a zabráňte tomu, aby útok ovlivnil šířku pásma a výkon brány firewall
Linková vrstva	Rámec	Vytváří, udržuje a rozhoduje o tom, jak se provádí přenos přes fyzickou vrstvu	Používá protokoly 802.3 a 802.5, což jsou zařízení NIC, přepínače, mosty a WAP	Zaplavení MAC - zaplavuje síťový přepínač datovými pakety	Přeruší obvyklý tok dat odesílatele příjemci - odesílání přes všechny porty	Mnoho pokročilých přepínačů lze nakonfigurovat tak, aby omezovaly počet MAC adres, které lze zjistit na portech připojených ke koncovým stanicím; povolit autentizaci objevených MAC adres proti autentizačnímu, autorizačnímu a účetnímu (AAA) serveru a následně filtrovat
Fyzická vrstva	Bity	Zahrnuje mimo jiné kabely a rozbočovače	Používá protokoly 100 Base-T a 1000 Base-X a jako zařízení používá rozbočovače, patch panely a konektory RJ45	Fyzické zničení, překážky, manipulace nebo nesprávná funkce fyzického zařízení	Fyzická aktiva přestanou reagovat a bude třeba je opravit, aby se zvýšila dostupnost	Cvičit hloubkovou taktiku obrany, používat řízení přístupu, odpovědnost a audit ke sledování a kontrole fyzických aktiv

Obrázek 3 Typy útoků na různých vrstvách (původní nepřeložený zdroj: https://www.us-cert.gov/)

DoS a DDoS útoky můžeme rozdělit do několika kategorií:

Objemové útoky

Tento typ útoků je klasifikován jako jakýkoliv útok, kdy útočník úmyslně spotřebuje veškerou šířku pásma sítě (24). Jakmile je celá šířka pásma spotřebována, dané zařízení není v tu dobu pro uživatele v síti k dispozici. K tomuto typu útoků dochází, když útočník zaplaví síťové zařízení požadavky ICMP, dokud není celá šířka pásma využita. Velikost se měří v bitech za sekundu (Bps).

Fragmentační útoky

Fragmentační útoky jsou jakýkoliv druh útoku, který nutí síť k opětovnému sestavení zmanipulovaných paketů (24). Během tohoto typu útoku útočník odešle zmanipulované pakety do sítě, takže jakmile se snaží síť pakety sestavit, už je znovu sestavit nejde. Je to z důvodu, že pakety mají více informací v hlavičce, než je povoleno. Tento typ útoku využívá prostředky serveru nebo prostředky komunikačních zařízení, jako jsou brány firewall nebo vyrovnávače zatížení a měří se v paketech za sekundu (Pps) (23). Výsledkem je tedy velké množství paketů, které již nejde znovu složit, a začnou se hromadit.

Útoky na vyčerpání stavu TCP

U tohoto typu útoků útočník útočí na webový server nebo firewall ve snaze omezit počet relací, které mohou navázat (24). Hlavní myšlenkou tohoto útoku je posunout zařízení na jejich hranici možných souběžných připojení.

Útoky na aplikační vrstvě

Útoky na aplikační vrstvě tedy na 7 vrstvě referenčního ISO/OSI modelu jsou útoky, které cílí na aplikace nebo servery ve snaze využít jejich zdroje, vytvořením co největšího počtu procesů a transakcí. Tento typ útoků zahrnuje tzv. útoky low-and-slow attacks. *Cílem těchto útoků, které se skládají ze zdánlivě oprávněných a nevinných požadavků, je zhroucení například webového serveru. Velikost se měří v požadavcích za sekundu (Rps).* (23) Útoky na aplikační vrstvě je velmi obtížné detekovat a nějakým způsobem řešit, protože k útoku nepotřebují velké množství počítačů.

4.5.1 Denial of Service (DoS)

Jak již bylo výše vysvětleno, DoS útoky jsou využívány za účelem vyřazení služby z provozu. *Jejich velkou výhodou je snadnost, s jakou mohou být tyto útoky koordinovány. Útoky DoS jsou jednoduché, ale velmi efektivní a mohou způsobit ničivé škody společnostem nebo i jednotlivcům, na které jsou mířeny. Jedním takovým útokem může být organizace vyřazena z činnosti na dny až týdny.* (24) To společnost může stát mnoho peněz. Zde si nyní uvedeme několik konkrétních typů útoků,

které DoS útočníci využívají. Většina útoků je však stejná jak pro DoS tak DDoS útoky.

Útok Buffer Overflow

Toto je nejčastější typ útoku DoS. Útočník při tomto útoku přetíží síťovou adresu provozem, takže ji vyřadí z provozu. Vykonání tohoto typu útoku může být za pomoci paketů ICMP, UDP nebo TCP. Princip tohoto útoku spočívá v přetečení vyrovnávací paměti, kdy program překročí určitou hranici a přepíše tím sousední paměťová místa.

Útok Ping of Death nebo ICMP Flood útok

Tyto typy útoků jsou detailněji popsány níže u DDoS útoků.

Útok SYN Flood

Tento typ útoku je také detailně popsán níže u DDoS útoků.

Teardrop útok

Během tohoto typu útoku útočník odešle fragmenty datových IP paketů do sítě. Síť se poté pokouší překompilovat tyto fragmenty do původních paketů. Proces kompilace vyčerpá systémový výkon a končí to jeho zhroucením. Důvod selhání je, že jsou fragmenty navrženy tak, aby je systém nemohl znovu složit dohromady.

4.5.2 Distributed Denial of Service (DDoS)

Útok DDoS je jedním z nejčastějších typů útoků DoS, který se dnes používá. Důvodem je, že útočníci mají k dispozici větší počet strojů a pro případné oběti je obtížné určit původ útoku. *Útočník může útoky také spouštět vzdáleně pomoci podřízených počítačů, které jsou označovány jako zombie nebo roboti. Tyto roboti tvoří síť nazývanou jako botnet, která je spravována útočníkem pomocí příkazové řádky a řídícího serveru. Díky tomu může útočník koordinovat své útoky.* (24) Rozhodně není dobré DDoS útoky podceňovat, protože je velmi velký rozdíl, zda na nás útočí jeden počítač nebo botnet, který zahrnuje třeba sto počítačů. Zde si nyní

uvedeme několik konkrétních typů útoků, které DDoS útočníci využívají a je dobré je znát a být proti nim připraven.

Útok UDP Flood

Tento typ útoku zaplavuje cíl pakety UDP (User Datagram Protocol). Cílem tohoto útoku je zaplavit náhodné porty na vzdáleném hostiteli. To má za následek, že hostitel opakovaně kontroluje, zda aplikace naslouchá na tomto portu. Pokud na tomto portu není nalezena žádná aplikace, odpoví paketem ICMP „cíl nedosažitelný“. Tento proces spotřebovává síťové zdroje a znamená to, že může dojít až k nedostupnosti hostitele.

Útok ICMP (Ping) Flood

Tento typ útoku je podobný útoku UDP Flood. ICMP flood útočí, ale na cílový zdroj pakety ICMP Echo Request. Obvykle odesílá takové pakety co nejrychleji po sobě, aniž by čekal na odpověď. V případě tohoto typu útoku může dojít ke spotřebování jak odchozí, tak příchozí šířky pásma, protože servery oběti se často snaží reagovat na takový požadavek odezvou pakety ICMP Echo Answer. To má za následek výrazné celkové zpomalení systému.

Útok SYN Flood

Velmi rafinovaný typ útoku, který využívá three-way handshake komunikaci je právě SYN flood útok. Útok začíná odesláním požadavku SYN na zahájení TCP spojení s hostitelem, který musí být následně zodpovězen odpovědí SYN-ACK od tohoto hostitele a poté potvrzeno odpovědí ACK od žadatele. Útok, ale spočívá v tom, že se pošle více požadavků SYN, ale už se neodpoví zpětně na odpověď hostitele SYN-ACK nebo se odešle požadavek SYN z podvržené IP adresy. Výsledkem je, že hostitelský systém stále čeká u každého požadavku na potvrzení, dokud nebude možné navázat žádná nová spojení, až tím dojde k odmítnutí služby.

Útok Ping of Death

Útok Ping of Death spočívá v zasílání nesprávně naformátovaných nebo škodlivých pingů. *Maximální délka paketu IP paketu (včetně hlavičky) je 65 535*

bajtů. Datová vrstva však obvykle představuje omezení o maximální velikosti rámce – například 1500 bajtů po síti Ethernet. V tomto případě je velký paket rozdělen do několika fragmentů a příjemce znovu sestaví fragmenty do úplného paketu. (23) Výsledkem je tedy, že je sestaven paket, který je větší než 65 535 bajtů, což může vést k přetečení vyrovnávací paměti, která je přidělená pro paket, a to má za důsledek odmítnutí služby pro legitimní pakety.

Útok Slowloris

Dalším vysoce cíleným útokem je Slowloris. *Tento útok umožňuje jednomu webovému serveru odstranit jiný server, aniž by to ovlivnilo další služby nebo porty v cílové síti. Slowloris funguje tak, že drží co nejvíce aktivních spojení s cílovým webovým serverem tak dlouho, jak jen je to možné. Toho se dosáhne připojením k cílovému serveru, ale pouze s částečnou žádostí. Slowloris tedy neustále odesílá více hlaviček HTTP, ale nikdy nedokončí požadavek.* (23) Výsledkem je, že cílový server stále udržuje všechna tato spojení otevřená, až do doby, kdy nelze navázat další spojení, a další nová spojení pro legitimní klienty jsou odepřena.

Útok HTTP/S Flood

Při útoku HTTP flood útočník používá zdánlivě legitimní požadavky HTTP GET nebo POST k útoku na webový server nebo aplikaci. *Tento typ útoku probíhá na 7 vrstvě referenčního ISO/OSI modelu a nepoužívají se chybné pakety, spoofing nebo jiné reflexní techniky. Útočníci používají tento typ útoku, protože vyžaduje menší šířku pásma než jiné typy útoků, aby vyloučili cílovou síť z provozu.* (24)

4.6 Web Proxy Gateway

Podle (25) Web Proxy Gateway je zjednodušeně server, který zpracovává přenosy na a z webové stránky. Obvykle uživatel zadá adresu webové stránky, kterou si přeje zobrazit a prohlížeč odešle tento požadavek web proxy. Web proxy pak žádost prozkoumá a provede úkoly související se zabezpečením, jako je ověřování či autorizace a pokud neexistují žádné problémy, odešle žádost na server hostujícího webu. Web proxy také zkoumá požadovaný obsah pro malware a další hrozby, před odesláním do prohlížeče uživatele. *Web proxy v podstatě*

poskytuje službu „karantény“ pro webový provoz. Zkoumá 100 % provozu mezi uživateli a HTTP/HTTPS a kategorizuje všechny URL adresy tak, aby bylo možné identifikovat a blokovat škodlivé weby nebo stránky, zatímco dobré URL adresy zůstávají podle zásad pravidel přístupné. (25)

4.7 Další typy útoků

V této sekci jsou uvedeny další typy útoků, které nelze zařadit do žádné z již probraných kategorií výše.

Zero-Day útoky

Zero-Day útoky jsou útoky, které využívají zranitelnosti, které dosud nebyly objeveny. Zero-Day je všeobecný termín pro útoky, kterým by se mohlo v budoucnu čelit. Tyto typy útoků mohou být zvláště zničující, protože oběť nemá možnost se na takový útok připravit, než takový útok zažije na živo. Jediné, co pro to společnost může udělat je prevence.

Amplification útoky

Obecným principem amplification útoků je zneužití síťové služby k zesílení prováděného útoku. (26) Většinou jsou hlavně zneužívány především služby, u kterých vcelku malý dotaz vyvolá několikanásobně větší odpověď. Útočník potřebuje mít možnost podvrhnout zdrojovou IP adresu zasílaného požadavku tak, aby odpověď byla doručena na IP adresu cílené oběti. *Proto jak popisuje* (26) *jsou zneužívány služby, které ke své komunikaci využívají protokol UDP, u nějž tak na počátku neprobíhá trojcestný handshake, jako je tomu u TCP. V případě protokolu UDP jsou data z klientské aplikace rovnou zasílána cílovému serveru bez jakéhokoliv vzájemného ověření. Doručení paketů také není navzájem ověřováno a případnou ztrátu paketů během komunikace si musí řešit samotná aplikace.* A díky tomu, že UDP nemá mechanismus pro navazování komunikace, může útočník podvrhnout zdrojovou IP adresu a server pak považuje obdržený požadavek za skutečně odeslaný z podvržené IP adresy.

5 IDS – Intrusion Detection System

IDS (Intrusion Detection System) neboli systém detekce narušení je považován za monitorovací systém. IDS také nikterak nemění síťové pakety, ale pouze analyzuje a sleduje síťový provoz, kdy hlídá, zda útočníci nevyužívají známou kybernetickou hrozbu pro narušení systému a následné odcizení dat. *Definice IDS dle* (27) *říká, že cílem IDS je identifikovat, nejlépe v reálném čase neoprávněný přístup, zneužití a zneužití počítačových systémů, jak ze strany zasvěcených (insiders), tak ze strany externích narušitelů. Další definice dle* (28) *říká, že IDS je proces identifikace a reakce na škodlivou činnost zaměřenou na počítačové a síťové zdroje. Jedná se obvykle o zařízení, které je obvykle další počítač, který monitoruje aktivity k zajišťování škodlivých nebo podezřelých událostí v síti.* IDS jako takové přijímá surová data ze senzorů, analyzuje tyto data a poté až provede nějakou akci. IDS ve své podstatě detekuje narušení monitorováním sítě nebo systému a následně se musí nasbíraná data analyzovat za pomoci systému nebo člověka, kde se hledají stopy po škodlivém chování. IDS však není zcela žádnou novinkou a díky tomu existuje již mnoho útoků, které IDS umějí obejít. S ohledem právě na tyto útoky a na rostoucí převahu šifrované komunikace se staly populárnějšími IPS nebo kombinace IPS/IDS či propracovanější honeypoty.

6 IPS – Intrusion Prevention System

IPS (Intrusion Prevention System) neboli systém detekce narušení je kontrolním systémem, který analyzuje pakety, ale také může zastavit doručení paketů na základě detekce útoku. Pomáhá tím předcházet útokům na zabezpečení sítě, jako jsou brute force útoky DoS (Denial of Service), DDoS (Distributed Denial of Service) a zneužití zranitelnosti (29). Lze IPS také využít pro odepření použití nezabezpečených protokolů, jako jsou například dřívější verze SSL nebo protokoly, které používají slabé šifry. IPS a IDS systémy porovnávají síťové pakety s databází známých hrozeb, která obsahuje známé popisy kybernetických útoků a následně označí všechny pakety, které odpovídají popisu dle databáze. Databázi je nutné pravidelně aktualizovat na nové hrozby. Nachází se ve stejné oblasti jako firewall, což je v oblasti mezi vnější sítí a vnitřní sítí. IPS proaktivně zakáže síťový provoz na

základě bezpečnostního profilu, pokud paket představuje známou bezpečnostní hrozbu. *Podle* (30) *mnoho dodavatelů IDS/IPS integrovalo novější systémy IPS s firewallem, aby vytvořili technologií zvanou UTM (Unified Threat Management), která kombinuje funkčnost těchto dvou podobných systémů do jedné jednotky.* V praxi, ale tyto systémy nejsou hojně využívané. Účelem IPS je tedy zachytit nebezpečné pakety a vypustit je dříve, než dosáhnou svého cíle.

IDS		IPS
IDS je detekční a monitorovací systém		*IPS je kontrolní systém*
Nástroje IDS nedělají samy žádné akce	*Oba čtou síťové pakety a porovnávají obsah s databází známých hrozeb*	*Kontrolní systém přijímá nebo odmítá pakety na základě stanovených pravidel*
IDS vyžaduje, aby se na výsledky podíval člověk nebo jiný systém		*IPS potřebuje, aby databáze byla pravidelně aktualizována s novými daty o hrozbách*

Tabulka 1 Porovnání IDS a IPS

7 Honeypoty

Přesná definice honypotu je jasně stanovena, ale přesto se trochu od sebe liší jejich formou: *„Server, který je nakonfigurován tak, aby detekoval vetřelce zrcadlením skutečného produkčního systému. Vypadá to, že běžný server pracuje, ale všechna data a transakce jsou falešné. Honeypot, který se nachází uvnitř nebo vně brány firewall, se používá k seznámení se s technikami vetřelce a k určení zranitelností v reálném systému.“* (31) Další autor například popisuje honeypoty jako: *„Honeypot je prvek, jehož hodnota spočívá právě v jeho napadení a kompromitaci útočníkem“* (32). V praxi jsou honeypoty počítače, které se maskují jako nechráněné. Honeypot zaznamenává všechny akce a interakce s uživateli. Jelikož honeypoty neposkytují žádné legitimní služby, veškerá činnost je neoprávněná a možná i škodlivá. *Talabis představuje honeypoty jako analogické použití mokrého cementu pro detekci lidských vetřelců* (33). Velmi dobře také popisuje definici honeypoty stránka techopedia.com: „Honeypot je počítačový systém, který je návnadou a slouží pro zachycení hackerů, sledování nekonvenčních nebo nových hackerských metod. Honeypoty jsou navrženy tak, aby záměrně zapojovaly a oklamaly hackery a identifikovaly škodlivé činnosti prováděné přes internet.“ (34) Také je důležité zdůraznit, že honeypoty leží

v separované síti mimo firemní servery nebo koncové body a nemají ani konektivitu do takovéto sítě. IPS/IDS honeypoty pracují na L2/L3 ostatní, pokud provádí aplikační kontrolu tak pak operují na L7 ISO/OSI modelu.

7.1 Historie honeypotu

Jelikož historii honeypotů již popsal velmi dobře Lance Spitzner, budeme uvádět historii podle něho.

1990-1991: *Poprvé, co byla vydána studie o honeypotech, kterou vydali Clifford Stoll (The Cuckoo's Egg) a Bill Cheswick (An Evening with Berferd).*

1997: *Deception Toolkit verze 0.1 byl představen Fredem Cohen po předchozí publikaci z roku 1990-1991. Deception Toolkit dal díky tomu nápad, jak bude vypadat první struktura honeypotů.*

1998: *Byl vydán první komerční honeypot pod názvem CyberCop Sting.*

1998: *Byl představen BackOfficer Friendly honeypot. Konfigurace byla bezplatná a snadná. Fungoval pod operačním systémem Windows. Většina lidí si tento software vyzkoušela a díky tomu byl koncept honeypotů mezi lidmi stále více známý.*

1999: *Po BackOfficer Friendly se více lidí začalo zajímat o tyto nové technologie. V tomto roce byl také zahájen projekt Honeynet. Byly vydány také papíry s názvem Know Your Enemy. Díky všem těmto zprávám lidé začali lépe rozumět, k čemu vlastně honeypoty slouží.*

2000-2001: *Honeypoty začaly být používány pro zachycení škodlivého softwaru z internetu a pro uvědomování si nových hrozeb. Společnosti začaly honeypoty používat ve svých systémech pro zlepšení zabezpečení a sledování škodlivého provozu.*

2002: *Samotný koncept honeypotů se stal populární, což vedlo ke zlepšení jeho funkčností a také se tak staly honeypoty více užitečnými a zajímavými pro výzkumníky a společnosti. (32)*

7.2 Výhody a nevýhody honeypotů

Zde si zkusíme obecně říci v čem spočívá výhoda honeypotů a na co naopak honeypoty trpí, co je jejich slabinou.

7.2.1 Výhody

- Shromažďují menší datové sady s vyšší hodnotou, protože zaznamenávají pouze nezákonnou činnost
- Na rozdíl od IDS nevyžadují známé signatury útoků
- Jednoduché na nastavení
- Umí šifrovanou a IPv6 komunikaci
- Velmi malá chybovost
- Není potřeba velký výkon pro běh

7.2.2 Nevýhody

- Při neopatrném použití mohou být zneužity útočníkem na jiné systémy v síti (High-interaction)
- Sledují se pouze interakce, které jsou přímo s honeypotem – honeypot nemůže detekovat útoky mimo něj
- Útočník může potenciálně honeypot detekovat

7.3 Typy honeypotů

Rozdělení honeypotů je možní vzít z mnoha různých pohledů. Například v současné době máme známé zejména dva typy honeypotů a to High-interaction a Low-interaction. Další rozdělení honeypotů je založeno na směru interakce, zda je klientské nebo serverové. Honeypoty také můžeme dělit z hlediska, zda se jedná o honeypoty fyzické nebo virtuální. Poslední a asi i nejméně významné dělení honeypotů je pak dle jejich účelu, a to buď honeypoty produkční nebo výzkumné.

7.3.1 Dle míry interakce

Honeypoty s různou mírou interakce se rozdělují v zásadě do dvou skupin. High-interaction a Low-interaction avšak některé publikace uvádí ještě Medium-

interaction. Tyto kategorie jsou definovány na základě služeb nebo úrovně interakce, které honeypoty poskytují.

7.3.1.1 Low-interaction honeypoty (S nízkou mírou interakce)

Low-interaction honeypoty napodobují většinou jen určité aplikace, služby nebo jiné části reálného systému. Například to mohou být některé porty. *U low-interaction honeypotů je většinou k dispozici jedna nebo více jednoduchých služeb, které zaznamenávají všechny pokusy o komunikaci s konkrétními službami, jako je například web nebo SSH server* (35). *Pokud se podíváme na věc z pohledu útočníka, tak to znamená, že low-interaction honeypoty reagují pouze na dotazy, které byly tvůrcem honeypotu nasazeny. Obdobně je to tak i u zranitelností honeypotů.* (36)

Jelikož low-interaction honeypoty, jak již bylo řečeno výše, fungují pouze jako konkrétní služby nebo aplikace nejsou díky tomu náchylné k útokům a nelze je v podstatě zneužít k dalším útokům. *Díky tomu můžeme říci, že provozování těchto honeypotů je vcelku bezpečné. Další výhodou je jejich rychlé nasazení a jednoduchá správa. Stejně tak jsou i hardwarové požadavky vcelku nízké.* (37) *Nevýhodou low-interaction honeypotů je zejména omezený sběr dat o útočníkovi, takže většinou sběr dat je spíše kvantitativního rázu.* (38) Proto hlavním využitím honeypotů je sběr informací o škodlivém softwaru.

7.3.1.2 High-interaction honeypoty (S vysokou mírou interakce)

High-interaction honeypoty jsou mnohem komplexnější a složitější. *Útočníkovy je představen reálný operační systém, kde jsou provedeny pouze drobné změny, aby bylo možné jejich monitorování, a kde není nic omezeno nebo emulováno. Obvyklým cílem high-interaction honeypotů je, aby útočník získal kořenový přístup na zařízení, a pak jsme sledovali, jak a co útočník dělá při hackingu.* (35) Z tohoto důvodu je u high-interaction honeypotů nejvyšší riziko, že mohou být zneužity k dalším útokům, ale také mají zároveň největší potenciál pro shromažďování informací. *High-interaction honeypoty tak vyžadují neustálý dohled, aby se právě nemohly stát bodem pro další útoky.* (35) U high-interaction honeypotů může analýza nasbíraných dat zabrat i několik dní.

Velkou výhodou těchto honeypotů je jejich důvěryhodnost. Je však důležité dobře honeypot nastavit, jinak může být lehce rozpoznán, a to na základě například výchozích přihlašovacích údajů, minimum spuštěných procesů, neexistence dat nebo právě naopak až moc dobrá data. Další výhodou je, že high-interaction honeypot díky sběru dat může odhalit úplně nový typ útoků nebo škodlivého softwaru. (37)

Nevýhodou je větší hardwarový nárok na běh, jelikož reálný systém potřebuje více výkonu oproti pouhé emulaci. *Další nevýhodou je, jak již bylo zmíněno výše, že útočník může honeypot využít k dalším útokům. Může volně instalovat nový software nebo provádět jakékoliv změny. Tímto počínáním může ohrozit naši okolní síť a v ní ostatní stroje.* (38)

7.3.1.3 Medium-interaction honeypoty (Se střední mírou interakce)

Většina publikací vůbec medium-interaction honeypoty neuvádí, jelikož se jedná o honeypoty, které jsou něco mezi low-interaction a high-interaction honeypoty. Medium-interaction honeypoty by mohly být definovány takto: „Jsou to vlastně low-interaction honeypoty, které poskytují pouze částečnou implementaci služeb a neumožňují typickou plnou interakci se systémem jako high-interaction honeypoty.“ (39)

Interakce honeypotu	Nastavení a následná správa	Sběr dat	Škálovatelnost	Riziko zneužití honeypotu	Věrohodnost
Vysoká (High-interaction)	Obtížné	Vysoký	Nízká	Vysoké	Vysoká
Nízká (Low-interaction)	Jednoduché	Nižší	Vysoká	Nízké	Nízká

Tabulka 2 Porovnání vlastností honeypotů na míře jejich interakce

7.3.2 Dle směru interakce

Rozdělení honeypotů můžeme dělit na klientské a serverové. Níže je popsán rozdíl mezi klientským honeypotem a serverovým honeypotem.

7.3.2.1 Klientské

Klientský honeypot je vcelku nový typ, který se aktivně hlásí k různým vzdáleným serverům a jejich službám, aby ověřil, zda nebyla provedena nějaká změna v jeho systému (36). Cílem takového honeypotu je detekce útoků na

klientské aplikace. Většinou se jedná hlavně o detekci zvláštního chování serveru nebo jeho obsahu. Klientský honeypot tedy aktivně vyhledává servery a jejich služby, na které se poté připojí. Nejčastější klientskou aplikací, která je napadena, je webový prohlížeč včetně pluginů. *U klientského honeypotu je velká pravděpodobnost, že bude nakažen, proto je vhodné takový honeypot umístit do vlastní podsítě* (40).

7.3.2.2 Serverové

Serverové honeypoty jsou nejvíce rozšířené. *Můžeme takové honeypoty označit též jako pasivní, jelikož nevyvíjí žádnou aktivitu a čekají, až na ně bude zaútočeno. Velkou výhodou u serverových honeypotů je, že dokáží zpracovat velké množství dat a tím pádem jsou schopny odhalit nové hrozby.* (41) Takovým případem může být například pozorování služby SSH, kdy honeypot sleduje útočníky, kteří se pokouší získat přístup k systému.

7.3.3 Dle typu provozu

Zde se zaměříme na to, jaký je rozdíl mezi honeypoty fyzickými a virtuálními.

7.3.3.1 Fyzický

Fyzický honeypot, je takový honeypot, který funguje na svém vlastním hardwaru a má svoji vlastní IP adresu (38).

7.3.3.2 Virtuální

Kdežto virtuální honeypot je emulace jiným zařízením, které odpovídá na síťové požadavky, které jsou směřované na adresu honeypotu (38). Velkou výhodou, které mají virtuální honeypoty oproti fyzickým je možnost běhu více honeypotů na jednom fyzickém zařízení. Další výhodou jsou náklady, které by při běhu fyzických honeypotů byli velmi nákladné, ale zde stačí jeden výkonný fyzický stroj, který může právě provozovat více honeypotů naráz. *Poslední výhodou, kterou se virtuální honeypoty honosí je možnost tzv. snapshotu, což je možnost vrátit nastavení honeypotu do stavu, kdy byl vytvořen, pokud se stane, že byl nakažen útočníkem* (36).

7.3.4 Dle důvodu nasazení

Rozlišujeme pouze dva typy, kde nasazujeme honeypoty, a to produkční a výzkumné.

7.3.4.1 Produkční

Produkční honeypoty jsou nasazovány zejména za účelem ochrany organizace. Jsou zde z důvodu prevence, detekce útoků a také pro případnou zpětnou analýzu útoků a následné zlepšování bezpečnostních opatření v systému. (37) Většinou doporučované umístění honeypotu je mimo produkční síť. Cílem honeypotů totiž je schválně nechat neopravenou chybu a zjednodušit tím přístup do takového prostředí. Kdyby naopak honeypot byl umístěn v segmentu produkční sítě, došlo by k bypassu všech kontrol, které jsou na cestě. V takovém případě se pak využívají low-interaction honeypoty nebo host-based IDS na koncovém serveru. Snaží se vlastně odvrátit pozornost útočníka od reálných serverů, kde leží důležitá data a snaží se ho nasměrovat právě na honeypot.

7.3.4.2 Výzkumné

Hlavním cílem výzkumných honeypotů je sledování a studování metod útočníků, jejich technik a nástrojů, které k tomu používají. Podle Galetky *„sběr údajů má následně různé využití – sledování trendů a způsoby útoků, k preventivním varovným účelům, odhalováním nových typů útoků a podobně. Nepřináší přímý prospěch pro konkrétní firmu, jsou nejčastěji provozovány specialisty, vzdělávacími a vládními institucemi a dobrovolníky se zájmem o bezpečnost.“* (42)

7.4 Honeyfarms or Honeynets and Honeywall

Honeyfarmy a honeynety jsou v podstatě to samé, akorát honeyfarmy bývají více centralizované. Dále budeme mluvit pouze o honeynetech, jelikož jejich funkčnost je velmi podobná, až téměř totožná.

V souvislosti s honeyfarmami a honeynety existuje také pojem honeywall, který by neměl nikdy chybět.

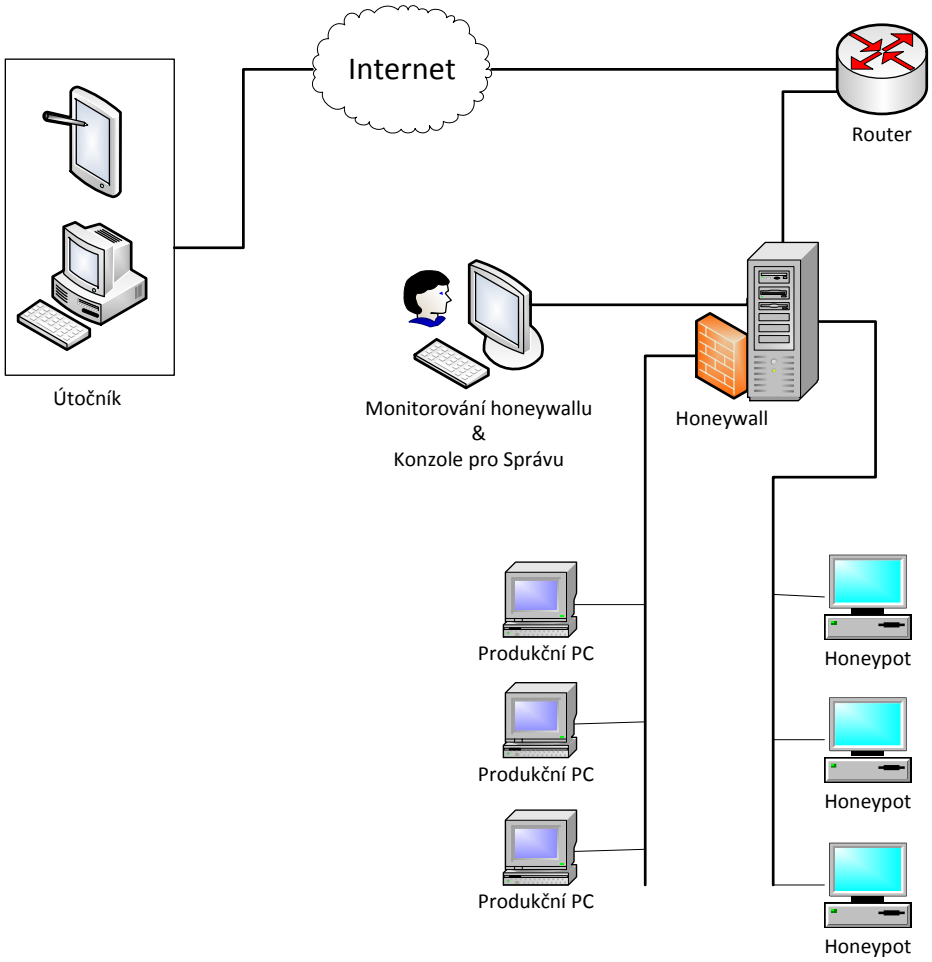
7.4.1 Honeynet

Honeynet je počítačová síť, která obsahuje jeden a více honeypotů, které sami o sobě nemají žádnou přidanou hodnotu a jsou striktně monitorované. (32) Seskupování honeypotů do honeynetu poskytuje mnoho výhod, kterými jinak samostatné honeypoty trpí. *Honeynet může být klidně celá napodobená produkční síť, kde běží různé služby a vše je striktně monitorováno. Díky honeynetu je tedy pro útočníky složitější se dostat právě k produkčním strojům.* (37)

Vhodný způsob, jak útočníka odlákat od produkční sítě je přesměrování do honeynetu, avšak do honeynetu by neměl být přesměrováván takový provoz, který je v pořádku. Z honeynetu nemůže útočník produkční síť napadnout, pokud se nezmocní nějakého honeypotu. *Proto by jednotlivé honeypoty měli být důkladně monitorovány. Díky tomu, že v honeynetu je více honeypotů, může každý honeypot sledovat něco jiného a tím si můžeme zajistit sběr širšího rozsahu dat.* (37)

7.4.2 Honeywall

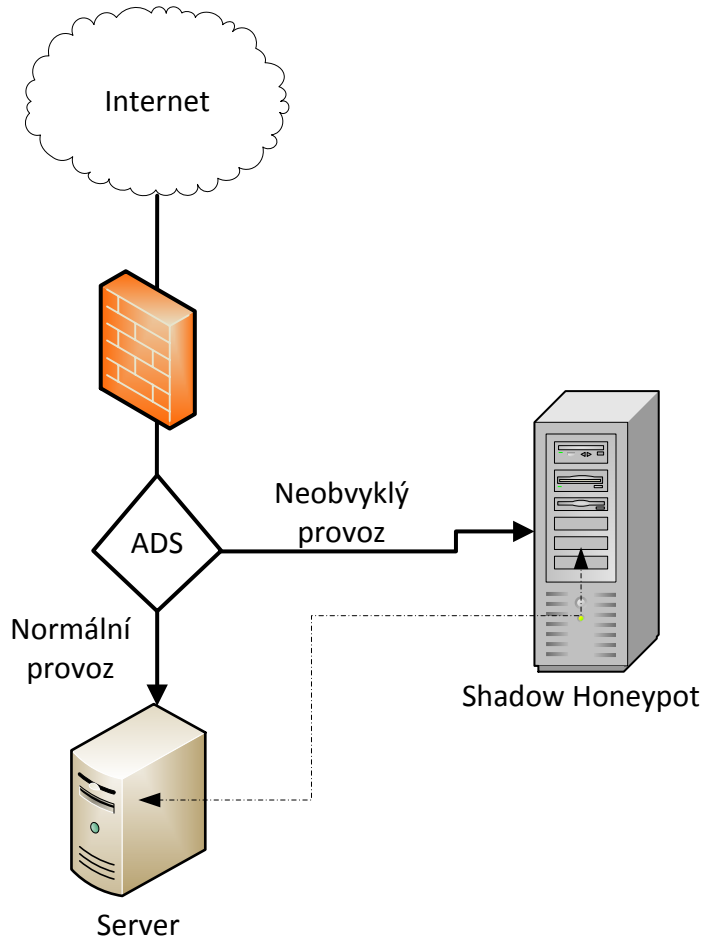
Honeywall je důležitý prvek v architektuře honeynetu. *Tento prvek odděluje honeynet od ostatní sítě viz. Obrázek 1. Veškerý provoz, který probíhá v síti, a to jak v odchozí nebo příchozí, proudí právě přes honeywall. Díky honeywallu a kontrole provozu, který vykonává, se snižuje riziko, že bude produkční sít napadena.* (43)



Obrázek 4 Umístění honeywallu a oddělení produkční sítě od honeynetu (vlastní zpraování)

7.5 Shadow honeypot

Architektura shadow honeypotu tkví v systémovém přístupu k řešení útoků na síť, kdy kombinuje filtrování, ADS (Anomaly Detection Systém) a honeypoty, který využívá ty nejlepší vlastnosti z jednotlivých mechanizmů a zároveň navzájem chrání nedostatky. (44) *Shadow honeypoty nejdříve oddělí anomální provoz od toho běžného. Pokud je detekován shadow honeypotem útok, tak jsou všechny změny zamítnuty. Pokud ne, komunikace a změny probíhají správně a jsou zpracovány.* (39)



Obrázek 5 Schéma zapojení Shadow honeypotu (vlastní zpracování)

7.6 Distributed honeypot

Aby byl honeypot účinný, musí zabírat velkou část adresního prostoru. A to, protože honeypot musí být přímo napadnut, aby mohl provádět monitorování.

Díky tomu se začali provozovat tzv. distribuované honeypoty. Distribuované honeypoty poskytují distribuovaný rámec pro výpočetní síť, ve které jednotliví hosté, kteří se zapojili přeposílají podezřelou komunikaci na jeden honeypot. Klientský provoz je přesměrováván anonymně například skrze Tor síť na centrální honeypoty. (39)

7.7 Wireless honeypot

Nyní se podíváme na bezdrátové honeypoty. *Jak už název napovídá, jedná se o honeypoty, které zachycují škodlivé chování v bezdrátové oblasti. Nejedná se pouze o wifi, ale i například bluetooth a další podle standardu IEEE 802.11.* (45)

8 Právní problémy s honeypoty

V této kapitole se budeme zabývat bezpečnostními problémy honeypotů souvisejících se zákonem.

8.1 Je používání honeypotů nezákonné?

Při nasazování a používání honeypotů by měl člověk vědět o některých právních problémech. Každá země má však jiné zákony týkající se používání honeypotů a shromažďování informací. Tyto předpisy se týkají tří částí, a to zabezpečení dat, shromažďování dat a způsobu využívání dat. Všechny tyto zákony se odvíjejí od toho, kolik informací honeypoty samotné sbírají a na osobě, která honeypoty nasazuje a zodpovídá za ně. Je složité tedy říci, zda je využívání honeypotů zákonné nebo nikoliv. Jak jsme si již řekli, záleží na záměru a použití shromažďovaných informací. Musíme tedy přemýšlet před nasazením o několika věcech, které je potřeba promyslet. Existuje také několik otázek, které bychom si měli položit před nasazením honeypotu. Jednou otázkou je, zda honeypot, který nasazujeme je pro firmu nebo pro domácí využití. Nejprve bychom se měli také zamyslet nad zákony jednotlivých zemí, které se k honeypotům vztahují a poté na zákony, které si stanovuje společnost sama, pokud se jedná o firemní nasazení. Před nasazením ve firmě by měl administrátor kontaktovat odpovědné osoby, které mu tyto otázky zodpoví, aby se administrátor ujistil, že nejedná v rozporu s právními zákony firmy nebo státu.

8.1.1 Ochrana osobních údajů

Zde se podíváme konkrétněji na ochranu osobních údajů. Podíváme se na to z hlediska člověka ve firmě, zda má právo shromažďovat informace od ostatních uživatelů společnosti. V souladu s tím je to stejná logika jako u hackera. Má hacker právo shromažďovat tyto informace? Můžeme se však na to podívat i opačně, máme právo shromažďovat informace o hackerovi? Soukromí je zde relativní. Jelikož existuje několik úrovní interakce honeypotů, jak jsme si již řekli v kapitolách výše, jsou získávané informace také relativní. Vyšší úroveň interakce znamená více informací, které můžeme zachytit. Otázkou je, jaké informace z honeypotu můžeme vzít, aniž bychom porušili zákony. Lance Spitzner odkazuje

na některé užitečné body, které pocházejí z ministerstva spravedlnosti. Těmi tedy jsou:

Lidé, kteří napadají tyto systémy nemají oprávnění je používat a pokud na ně umístí jakékoliv soubory (pokud nemají legitimní účty nebo oprávnění), vzdali se svých práv na ochranu osobních údajů k těmto souborům a údajům, které dali na honeypot.

Používáním honeypotů pro komunikaci se vzdali svých práv pro tuto komunikaci. Honeypoty obecně neposkytují veřejné účty, proto nejsou poskytovatelem služeb a nejsou vázáni požadavky na ochranu soukromí, které je jinak určené pro poskytovatele služeb. (46)

9 Rešerše praktická část

S neustále navyšujícím počtem metod a taktik, které používají útočníci k útoku na sítě se musí také neustále zlepšovat zabezpečení sítě. Zatímco tradiční metody, jako jsou systémy IDS / IPS, různé penetrační testy a další nástroje mohou vytvořit velmi bezpečnou síť, je ale vhodné předpokládat, že chyby zabezpečení vždy budou existovat a dříve nebo později budou využity útočníky. *Musíme tedy neustále hledat inovativní způsoby, jak čelit hrozbám a jedním z takových způsobů jsou právně naše honeypoty, které se nasazují nad standardní bezpečnostní mechanismy* (32). Honeypotů jako takových je mnoho druhů, podle toho, co mají zachytávat a detekovat. Co všechno nám, ale takové honeypoty mohou poskytnout a k čemu mohou sloužit? Podle článku od autora pod pseudonymem 6c2e6e2e, který pracuje jako bezpečnostní IT specialita, honeypoty slouží a poskytují následující:

- *Poskytují komunitě seznam IP adres, které se účastní pokusů o útoky a hackování nebo jsou zdroji jiného škodlivého chování a také pomáhají dalším kolegům v oblasti IT security po celém světě.*

- *Slouží pro shromažďování malwaru a klientů botnetů, což slouží následně určitým komunitám, aby zabránili dalším takovým útokům.*

- *Slouží pro obranu produkční sítě a přidávají čas, který útočník ztrácí díky honeypotu namísto páchání škod v produkční síti. (47)*

My se budeme zabývat honeypoty Dionaea, Cowrie a správcem pro některé určité typy honeypotů nazývaným MHN (Modern Honey Network). *Jak říká Jason Trost, honeypoty nebyli přijaty v širokém spektru jako podniková obrana hlavně proto, že nasazení a správa byli komplikované a díky tomu mohli honeypoty nastavovat dříve většinou jen společnosti zabývající se bezpečností nebo bezpečnostními výzkumníky. MHN je však projekt s otevřeným zdrojovým kódem, který umožňuje využívat honeypoty v širším měřítku pro podnikové bezpečnostní týmy, protože je jednodušší a přehlednější.* (48) MHN je tedy centralizovaný server pro správu a sběr dat z honeypotů. MHN mimo jiné podporuje honeypot Dionaea i Cowrie.

Dionaea nástupcem Nepenthes je honeypot, který zachycuje malware a byl vyvinut v rámci projektu Google Summer of Code. Hlavním cílem honeypotu Dionaea je zachytit malware využívající zranitelná místa služeb, které komunikují přes síť, a nakonec získat samotnou kopii malwaru. *Stejně jako jakýkoliv jiný software bude pravděpodobně i honeypot Dionaea obsahovat zneužitelné chyby. Aby se minimalizoval dopad, běží Dionaea v omezeném prostředí bez oprávnění správce.* (49) Tan také popisuje z jakých protokolů Dionaea zachytává malware.

- *Server Message Block (SMB)*
- *Hypertext Transfer Protocol (HTTP)*
- *File Transfer Protocol (FTP)*
- *Trivial File Transfer Protocol (TFTP)*
- *Microsoft SQL Server (MSSQL)*
- *Voice over IP (VoIP) (49)*

Avšak tento seznam se může lišit v průběhu vývoje honypotu. Následnou analýzou útoků se pak zabývá článek od Austina Ponténa, který honeypot nasadil do

univerzitní sítě a zachycoval útoky. *Jak uvádí ve své práci zachytil přes 40 000 útoků z 12 000 různých IP adres, ale nepodařilo se mu zachytit jediný malware. Podle jeho práce byl nejvíce napadaným portem port 23.* (50) *Analýzou honeypotu Dionaea se také zabýval Davide Bove, který ve své práci prezentuje počet útoků na jednotlivé protokoly. Nejvíce napadanými protokoly u něj byli SipSession, SipCall, mssqld, mysqld a smbd. SipSession a SipCall protokoly dohromady dělali 87 % celkových útoků. Dále popisuje, že útok smbd neboli SMB protokol, který je stále relevantní v oblasti informační bezpečnosti již od roku 2017.* (51) *Velmi detailně se zabýval analýzou útoků ještě autor Rasmi Vlad Mahmoud, který zaznamenával útoky a nasbíral přes 2800 útoků. Mezi hlavní útočící země podle jeho analýzy patřila Čína, která útočila hlavně na protokol mssqld a to v poměru 58,93 % z celkových útoků, následně Irsko, které také útočilo na protokol mssqld v poměru už pouhých 3,42 % a stejně tak třetí země, kterou bylo USA s pouhým 1,84 %. Mezi hlavní protokoly, které byli zaznamenány Dionaeou, tím pádem patřili mssqld útoky s 60,49 %, následně pak mysqld a httpd.* (52) Ještě se také zabývali stejnou problematikou Gary Kelly a Diane Gan, kteří přišli s následující analýzou. *Nejvíce útoků pocházelo z Pakistánu, Číny a zbytek byl v poměru k těmto dvěma zemím velmi malý. Následně také vyhodnotili, jaké porty, tudíž i protokoly byli nejvíce napadané. Podle jejich výzkumu to byl protokol SSH s portem 22 na který dopadlo 90,8 % všech útoků.* (53)

Také je nutné si něco říci o útocích na honeypot Cowrie, který v naší praktické části simulujeme jen okrajově. *Honeypot Cowrie je středně interakční SSH a Telnet honeypot, který může logovat útoky hrubou silou a interakci útočníka se shellem. Cowrie je open source projekt vyvíjený Michaelem Oosterhofem.* (54) Pokud tedy potřebujeme cokoliv ohledně Cowrie, je zde online stále aktualizovaná dokumentace, kde nalezneme i různé problémy a jak je řešit. Například můžeme najít dokumentaci na tomto odkaze: https://readthedocs.org/projects/cowrie/downloads/pdf/latest/. Detailní analýzou útoků se například zabýval autor William McCann. *Během 16 září až 6 října byl honeypot vystaven útokům, kdy na honeypot útočilo 1754 různých IP adres. Zobrazuje ve své analýze také nejčastější uživatelská jména, které byli použity útočníky pro přístup. První bylo admin, druhé root, třetí support.* (55) Většinou se jednalo o výchozí hesla různých zařízení. *Stejně tak zobrazuje nejčastější hesla,*

které byli použity pro útok. První heslo bylo ' ', druhé heslo bylo password, třetí 123456, čtvrté admin, páté 1234. Většinou se tedy jednalo o číselnou řadu nebo kombinace admin + číslo. (55) Jeho analýza však zahrnuje mnoho dalších zajímavých výsledků. Jedním z nich je ještě počet útoků, které provedl člověk a který bot. *Popisuje, že procentuální útok od člověka je pouhých 5 % a zbylých 95 % provádí bot.* (55)

10 Stanovení hypotéz

Nyní si stanovíme hypotézy, které budeme v průběhu testovat a na konci si tyto hypotézy vyhodnotíme. Důležité je říci si, proč jsme si tyto hypotézy stanovili. Hypotéza 1 byla stanovena, na základě tvrzení z rešerše od Davida Bova, který uváděl, že protokoly SipSession a SipCall obsadili v jeho výzkumu první pozice, kdy SipSession protokol měl 45,1 % útoků a SipCall protokol 41,9 % útoků. Tudíž tyto dva protokoly celkově měli 87 % podíl ze všech typů útoků. Hypotéza 2 byla stanovena na základě výzkumu od Rasmi Vlad Mahmouda, který ve svém výzkumu zjistil, že útoky pocházeli zejména z Číny, která leží v Asii a tvořila sama o sobě 58,93 % celkových útoků na honeypot. Proto jsme si stanovili hypotézu, kdy na náš honeypot s jiným geografickým umístěním bude dopadat minimálně 40 % celkových útoků, a to z celé Asie. Důvod, proč jsme si stanovili menší procentuální zastoupení, přestože budeme brát v potaz celou Asii je odlišné geografické umístění serveru s honeypotům na rozdíl od zkoumání Rasmi Vlad Mahmouda.

10.1 Hypotéza 1

Typ útoku SipSession na honeypot Dionaea tvoří víc jak 40 % celkových útoků všech typů.

10.2 Hypotéza 2

Útoky na honeypot Dionaea s umístěním serveru v Německu pochází výhradně z Asie a zda tyto útoky tvoří více jak 40 % celkových útoků.

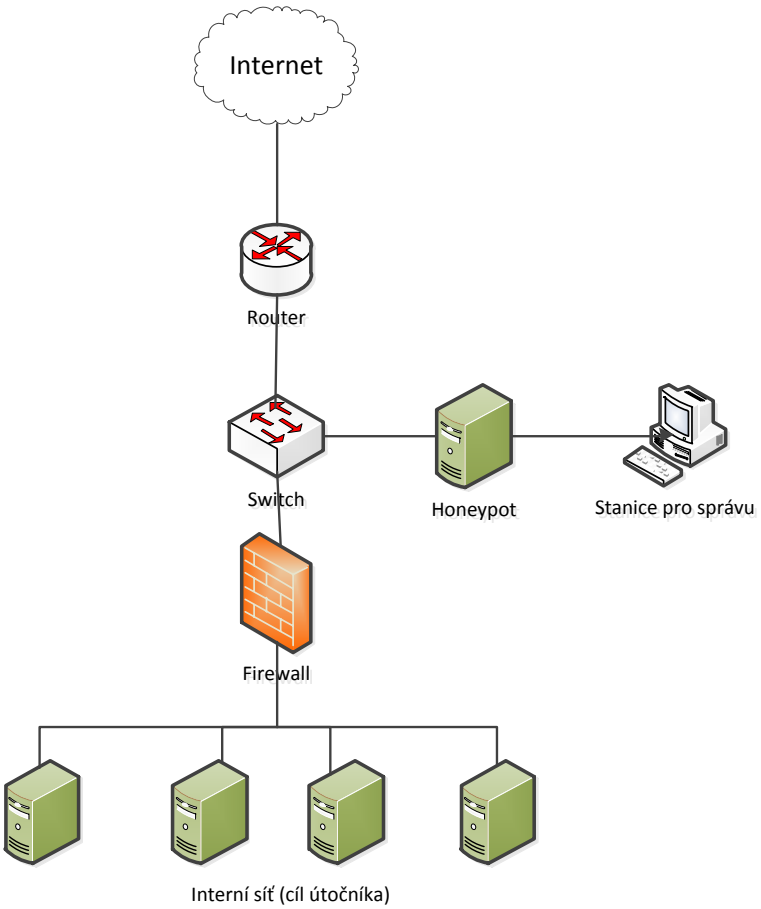
11 Konfigurace honeypotů

V této kapitole se zaměříme, jak nainstalovat a nakonfigurovat grafické prostředí pro honeypoty nazývané Modern Honey Network (MHN). Následně si nainstalujeme a nakonfigurujeme honeypot Dionaea a ten následně propojíme do MHN. V MHN si ukážeme, co je zde vše možné sledovat a analyzovat.

V další části si představíme, jak nainstalovat a nakonfigurovat honeypot Cowrie s nadstavbou IDS, a to konkrétně Snortu. Výsledkem by měl být funkční honeypot s IDS, kde si ukážeme, jak takový útok na tento typ honeypotu vypadá.

Veškeré testování bude prováděno skrze službu Digital Ocean, která poskytuje cloudovou infrastrukturu po celém světě a díky tomu dosáhneme více útoků na naše honeypoty. Zároveň pro přístup k jednotlivým dropletům (strojům) bude využito externího programu Putty.

Ještě, než přejdeme k samotným honeypotům, je důležité si ukázat, jak vypadá topologie zapojení honeypotu do síťové infrastruktury.



Obrázek 6 - Topologie pro umístění honeypotu (vlastní zpracování)

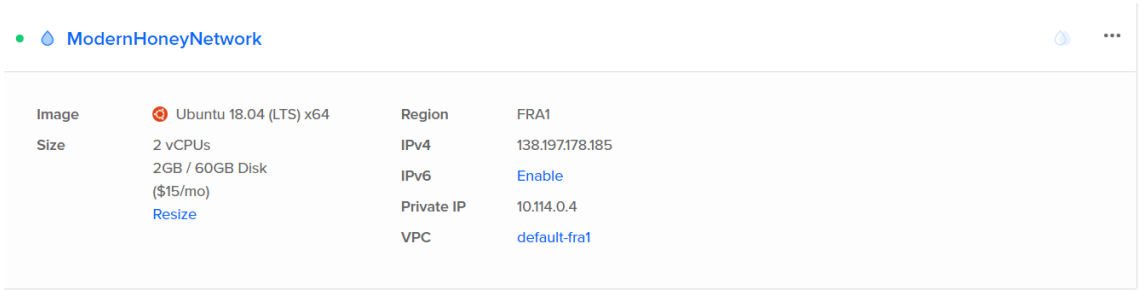
11.1 Modern Honey Network (MHN)

Modern Honey Network je centralizovaný server pro správu a sběr dat z honeypotů. MHN umožňuje rychlý sběr dat, který lze zobrazit z přehledného webového rozhraní. MHN je podporováno na operačním systému Ubuntu verze 18.04 nebo verze 16.04 případně Centos 6.9. Pro naši ukázku budeme využívat operační systém Ubuntu verze 18.04.

11.1.1 Instalace

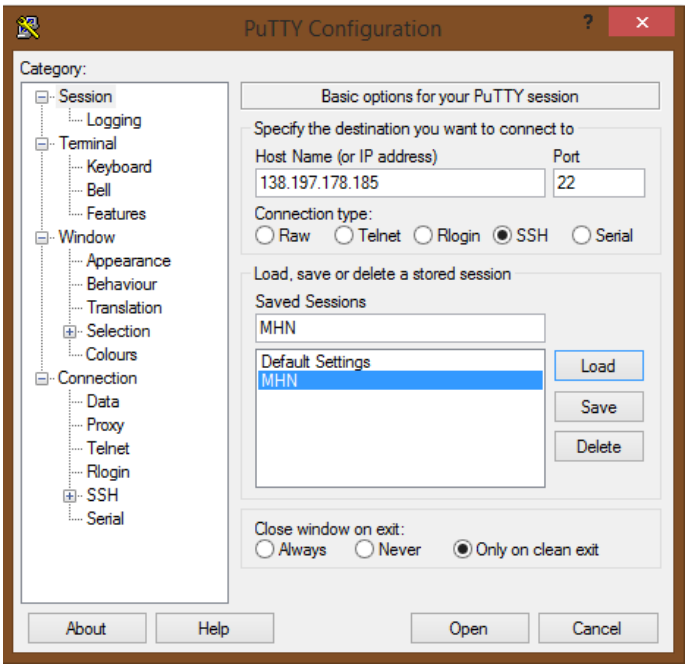
Jak již bylo zmíněno výše, veškeré testování bude prováděno skrze službu Digital Ocean, kde si nejprve vytvoříme droplet s operačním systémem Ubuntu verze 18.04 s x64 architekturou. Jelikož budeme potřebovat více místa, kvůli větším logům, které nám bude následně honeypot Dionaea generovat, vybereme možnost většího uložiště. Budeme používat specifikaci 2GB/2CPU a 60GB SSD. Další parametr, který musíme vybrat je umístění datového centra, které má Digital Ocean po celém světě. Zvolíme si nejbližší možné a tím je pro nás Frankfurt v Německu. Poslední částí při tvorbě dropletu je heslo pro přístup, které následně slouží i jako heslo pro uživatele root.

Nyní se nám nainstaloval stroj s předem námi stanovenými specifikacemi, kde je možný přístup pouze skrze konzoli, bez grafického rozhraní. Na obrázku níže, můžeme vidět, pro rekapitulaci naši specifikaci stroje a přidělenou veřejnou IP adresu.



Obrázek 7 - Specifikace vytvořeného dropletu pro MHN

V dalším kroku se připojíme skrze Putty na náš vytvořený droplet a začneme se samotnou instalací MHN.



Obrázek 8 – Připojení pomocí Putty k MHN dropletu

Jako první se přihlásíme pod uživatelem root, zkontrolujeme dostupné aktualizace pro naše nainstalované balíky a následně je zaktualizujeme, abychom předešli případným chybám během instalace.

$ apt-get update && apt-get upgrade

Pokračujeme instalací Gitu.

$ sudo apt install git -y

Nyní si zkopírujeme instalaci z Git repozitáře do našeho stroje a zahájíme samotnou instalaci pomoci již vytvořeného skriptu.

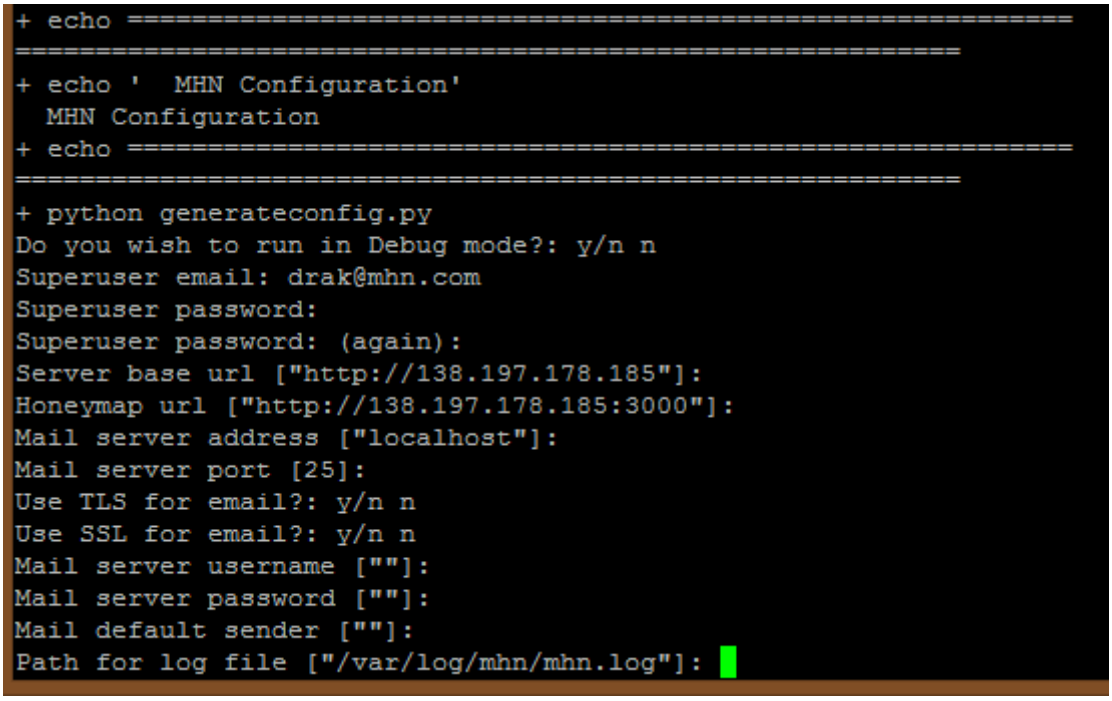
$ cd /opt/

$ sudo git clone https://github.com/pwnlandia/mhn.git

$ cd mhn/

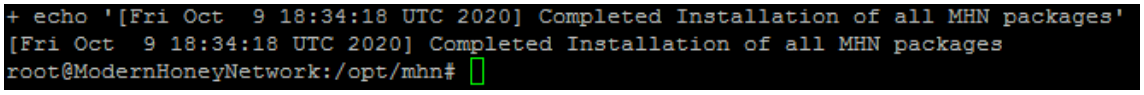
$ sudo ./install.sh

Během instalace budeme vyzváni ke konfiguraci MHN, kde je potřeba vytvořit uživatele, pod kterým se budeme do grafické správy MHN připojovat a samozřejmě i heslo. Další důležitý údaj, který musíme nastavit je IP adresa, kde má MHN běžet a IP adresa s portem, kde bude běžet tzv. Honeymap. Poslední, co je dobré si nastavit, kam se bude ukládat log MHN, ale v našem případě můžeme nechat výchozí umístění /var/log/mhn.log. Ostatní údaje pro nás nyní nejsou důležité, a proto ponecháme jejich výchozí nastavení.



Obrázek 9 - Konfigurace MHN

Ke konci instalace budeme dotázáni, zda chceme instalovat Splunk, ELK a pravidla pro UFW. Nic z toho v našem případě nechceme, proto všude zvolíme ne. Výsledná instalace by měla skončit hláškou viz. obrázek 10.



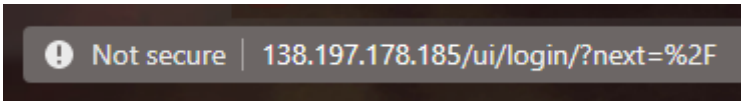
Obrázek 10 - Instalace MHN dokončena

Nyní si zkontrolujeme, zda nám všechny potřebné služby běží pomoci příkazu.

$ supervisorctl status

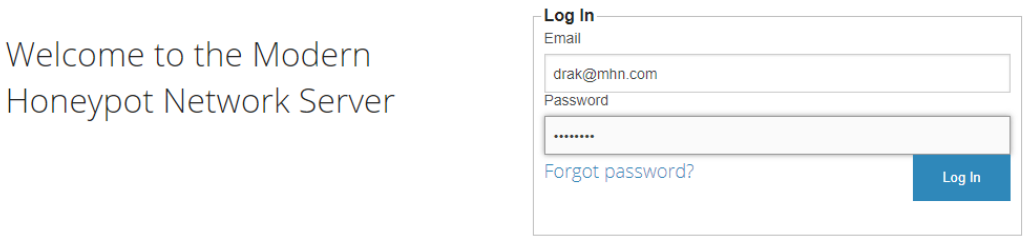
Obrázek 11 - Kontrola procesů pro běh MHN

Instalace proběhla v pořádku a můžeme si to vyzkoušet zadáním IP adresy do jakéhokoliv webového prohlížeče.



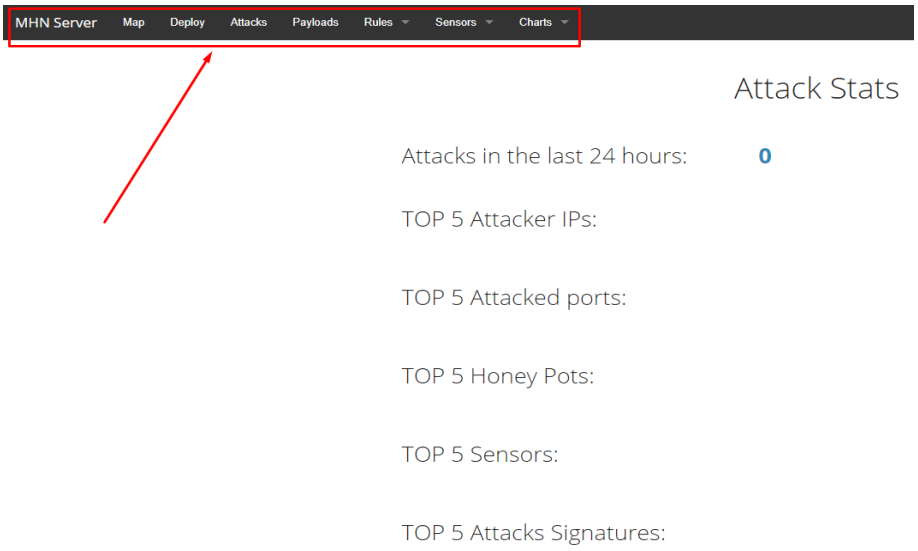
Obrázek 12 - URL MHN

Mělo by se načíst grafické rozhraní MHN, kde se musíme nejprve přihlásit pomoci údajů, které jsme zadali při konfiguraci MHN. V našem případě login: drak@mhn.com a heslo.



Obrázek 13 - Přihlašovací stránka MHN

Po přihlášení bychom se měli již dostat do přehledu MHN, které vypadá jako na obr. 14. Ve vyznačeném červeném rámečku je hlavní menu MHN serveru.



Obrázek 14 - Přehled MHN serveru

11.2 Honeypot Dionaea

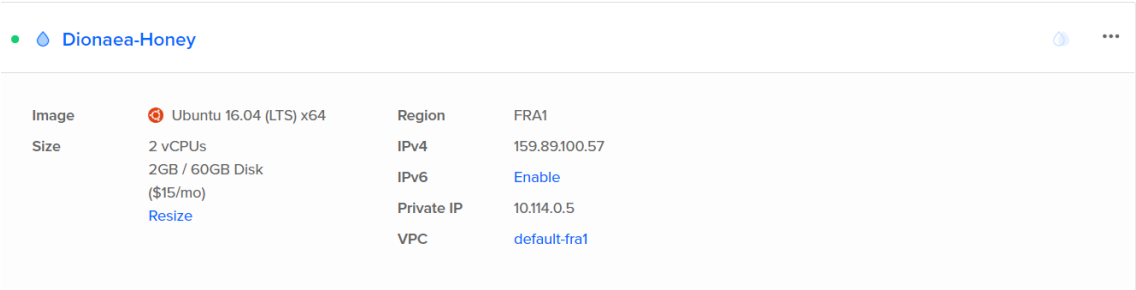
Záměrem honeypotu Dionaea je zachytit malware využívající zranitelná místa, která jsou vystavená službami nabízenými v síti. Konečným cílem je pak získat kopii malwaru. Dionaea jako software pravděpodobně bude mít chyby, které mohou být zneužity, ale snaží minimalizovat dopad napadení tím, že může zrušit oprávnění a chroot. Aby mohla Dionaea spustit určité akce, které vyžadují oprávnění, poté co je zruší, vytvoří při spuštění podřízený proces a požádá

podřízený proces, aby spustil právě tyto akce, které vyžadovali zvýšená oprávnění. To však nic nezaručuje, ale mělo by být mnohem těžší získat přístup root oprávnění do systému od neprivilegovaného uživatele v prostředí chroot.

Honeypot Dionaea by měl být podporován na Ubuntu verze 14.04 a 16.04. My si zvolíme verzi 16.04, protože ve verzi 14.04 byli již označeny nějaké bugy, díky vývoji všech potřebných knihoven.

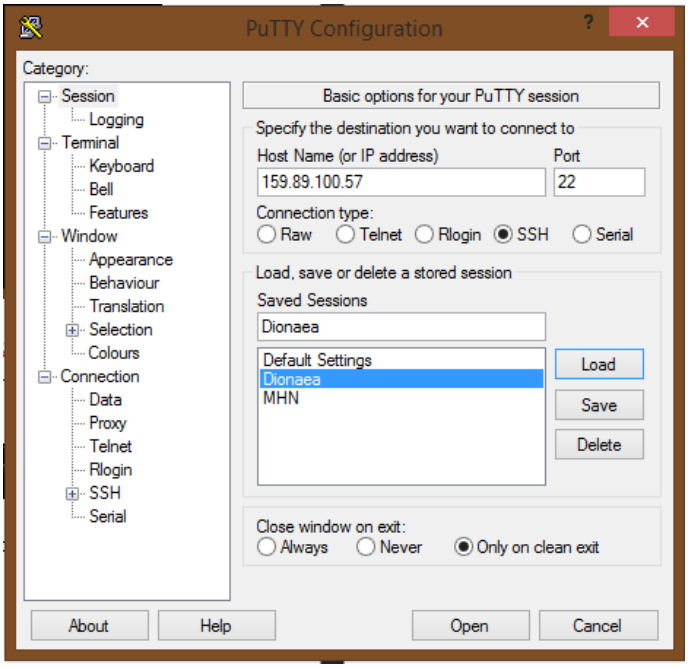
11.2.1 Instalace

Opět si vytvoříme droplet na Digital Ocean s Ubuntu verze 16.04 s architekturou x64, kdy postup a specifikace stroje je identická jako pro MHN server. Jedná se tedy o 2GB/2CPU a 60GB SSD. Opět zvolíme server ve Frankfurtu. Získáme tedy stroj se specifikacemi viz. Obrázek 15.



Obrázek 15 - Specifikace vytvořeného dropletu pro Dionaea honeypot

Stejně jako u MHN se budeme připojovat ke stroji skrze Putty pomocí SSH.



Obrázek 16 - Připojení pomocí Putty k Dionaea dropletu

Stejně jako předtím se přihlásíme jako root a provedeme prvně update a upgrade, kterým zkontrolujeme dostupné aktualizace pro naše nainstalované balíky a následně je i zaktualizujeme, abychom opět předešli případným chybám při instalaci honeypotu Dionaea.

$ apt-get update && apt-get upgrade

Nyní se vrátíme do grafického rozhraní MHN a vybereme požadovaný honeypot v našem případě honeypot Dionaea. MHN nám vytvoří skript, který následně vložíme do našeho stroje pro Dionaeu. Skript vypadá následně.

$ wget "http://138.197.178.185/api/script/?text=true&script_id=2" -O deploy.sh && sudo bash deploy.sh http://138.197.178.185 qMl1ntmo

Po dokončení skriptu, vidíme ve výpisu echo viz obrázek 17, které nám říká, že senzor byl vytvořen a spárován s naším MHN. Pro ujištění, zda Dionaea opravdu běží můžeme použít příkaz:

$ supervisorctl status

```
++ echo 'Created sensor: ' 4289760a-0b02-11eb-86d3-ba808b7139e5
Created sensor:  4289760a-0b02-11eb-86d3-ba808b7139e5
+++ echo http://138.197.178.185
+++ sed 's#^http://##; s#^https://##; s#/.*$##; s/:.*$//'
++ export HPF_HOST=138.197.178.185
++ HPF_HOST=138.197.178.185
++ export HPF_PORT=10000
++ HPF_PORT=10000
+++ python -c 'import json;obj=json.load(file("/tmp/deploy.json"));print obj["id
entifier"]'
++ export HPF_IDENT=4289760a-0b02-11eb-86d3-ba808b7139e5
++ HPF_IDENT=4289760a-0b02-11eb-86d3-ba808b7139e5
+++ python -c 'import json;obj=json.load(file("/tmp/deploy.json"));print obj["se
cret"]'
++ export HPF_SECRET=66keXv1Slz6rrR8m
++ HPF_SECRET=66keXv1Slz6rrR8m
+ cat
+ mkdir -p /opt/dionaea/var/log/dionaea/wwwroot /opt/dionaea/var/log/dionaea/bin
aries /opt/dionaea/var/log/dionaea/log
+ chown -R nobody:nogroup /opt/dionaea/var/log/dionaea
+ mkdir -p /opt/dionaea/var/log/dionaea/bistreams
+ chown nobody:nogroup /opt/dionaea/var/log/dionaea/bistreams
+ cat
+ supervisorctl update
dionaea: added process group
root@Dionaea-Honey:~# supervisorctl status
dionaea                          RUNNING   pid 4797, uptime 0:01:01
root@Dionaea-Honey:~#
```

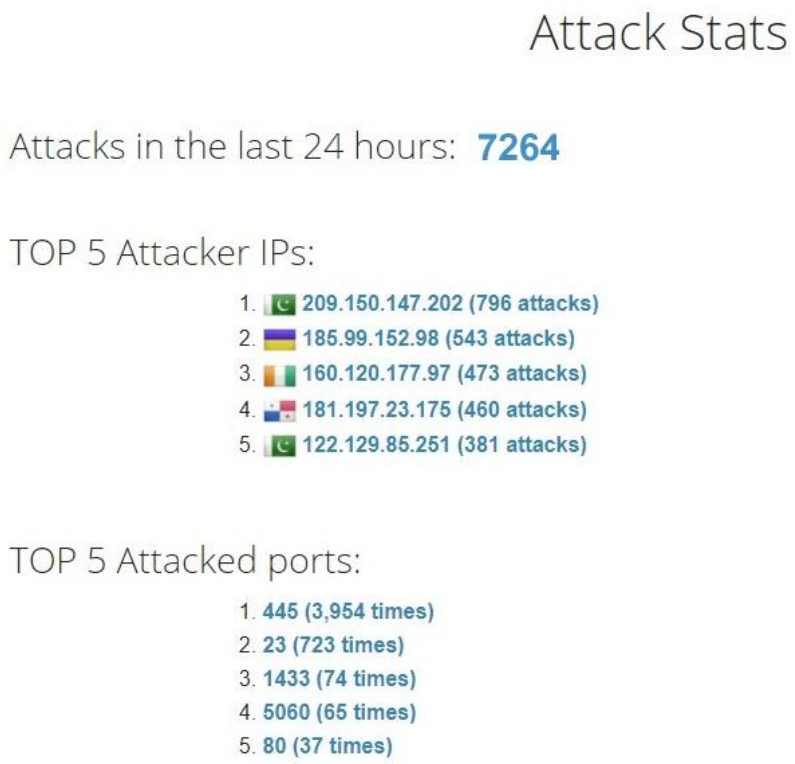
Obrázek 17 - Dokončení skriptu - deploy Dionaea

Pokud se nyní podíváme do grafického rozhraní MHN do sekce Sensors → View sensors, měli bychom vidět přidaný honeypot i zde. Stejně jako je na obrázku 18. Můžeme také vidět, že už na honeypot od doby depoly stihlo dopadnout 26 útoků.

Obrázek 18 - Sensors v MHN

11.2.2 Analýza útoků

V této kapitole se zaměříme na následnou analýzu útoků, které byli na honeypot Dionaea provedeny. Zaznamenávání útoků probíhalo po dobu 24 hodin. Za tuto dobu se nám stihlo nastřádat neuvěřitelných 7264 útoků. Veškeré útoky byli zaznamenávány do logu, který jsme si následně vyexportovali do json souboru a ten převedli do excelu. Díky tomu jsme schopni udělat o trochu detailnější analýzu útoků, než je uvedena na přehledové stránce MHN. Nejprve se však podíváme na výstupy z přehledové stránky MHN. Při pohledu na přehled našeho MHN, vidíme počet celkových útoků, TOP 5 IP adres odkud bylo útočeno a TOP 5 portů, které byli napadeny. Přehled vypadá stejně jako je na obrázku 19.



Obrázek 19 - Přehled útoků na honeypot Dionaea

IP adresy, odkud bylo útočeno patří Pakistánu, následně Ukrajině, Pobřeží Slonoviny, Panamě a nakonec opět Pakistánu.

Pokud se podíváme na porty, na které bylo útočeno, tak mezi favority patří TCP port 445, který využívá společnost Microsoft pro přímý přístup k síti bez nutnosti vrstvy NetBIOS. Druhým portem je port 23, který však už má pouze 1/5 útoků oproti portu 445. Tento port je využíván pro Telnet spojení. Třetím portem je port 1433 a ten je využíván pro komunikaci s Microsoft SQL serverem. Následuje ještě port 5060, který slouží pro SIP (Session Initiation Protocol), což je určeno pro přenos signalizace v internetové telefonii. Většinou funguje na UDP, ale může být používán i na TCP. A posledním portem je klasický port 80, který je využíván pro HTTP neboli pro komunikaci s www servery. Jak je ale vidět, první dva porty jsou mnohem častěji pod útokem než zbylé tři, kde zastoupení útoků je mnohem menší.

Nyní se podíváme na detailnější analýzu z logu, který jsme si stáhli z MHN. Díky němu jsme schopni zjistit na jaké všechny protokoly byly cíleny útoky a v jakém zastoupení viz. tabulka 3.

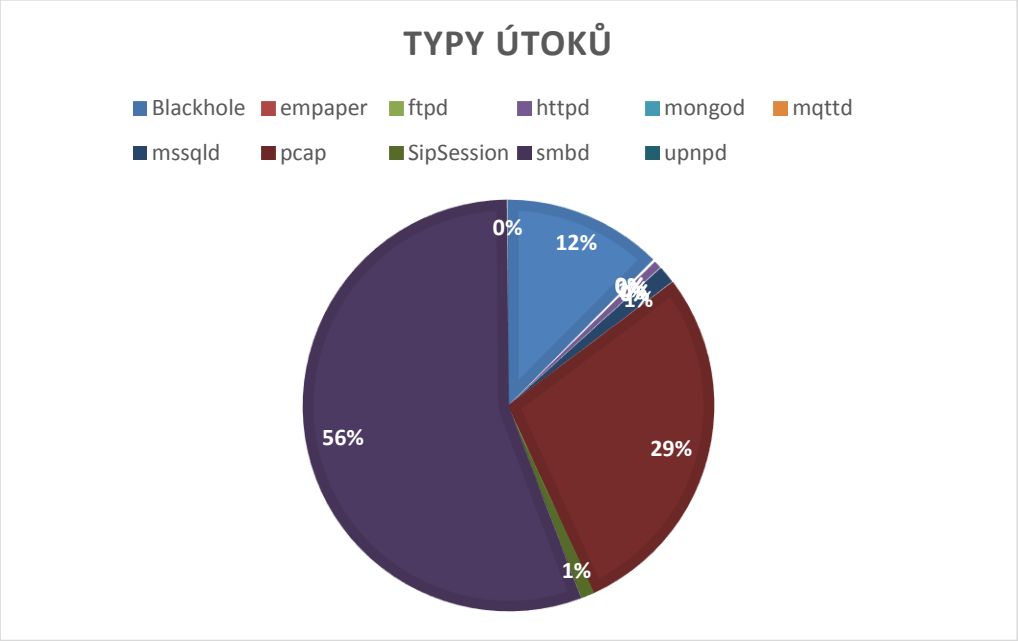
Typ útoku	Počet	Procentuální zastoupení útoků %
smbd	4036	55,56
pcap	2068	28,47
Blackhole	904	12,44
mssqld	101	1,39
SipSession	79	1,09
httpd	44	0,61
upnpd	11	0,15
ftpd	8	0,11
empaper	7	0,10
mongod	5	0,07
mqttd	1	0,01
	7264	**100**

Tabulka 3 - Přehled typů útoků s procentuálním zastoupením

Nejčastěji se jednalo o útoky na protokol smbd, který slouží ke sdílenému přístupu k souborům, tiskárnám a další komunikaci mezi zařízeními v síti. Tento protokol se využívá hlavně na počítačích s operačním systémem Windows. Druhým nejčastěji napadaným protokolem byl protokol pcap, který slouží jako aplikační rozhraní pro odchytávání síťové komunikace. Třetí typ útoku je odlišný od předchozích a jedná

se o útok na port 23 tedy útok o navázání telnet spojení. Dionaea však umožňuje black hole, což je místo v síti v našem případě náš honeypot, kde je příchozí nebo odchozí provoz zahozen nebo zrušen, aniž by byl zdroj informován, že data nedosáhla cíle.

Pro lepší přehlednost je zastoupení vyobrazeno na koláčovém grafu včetně procentuálního zastoupení.

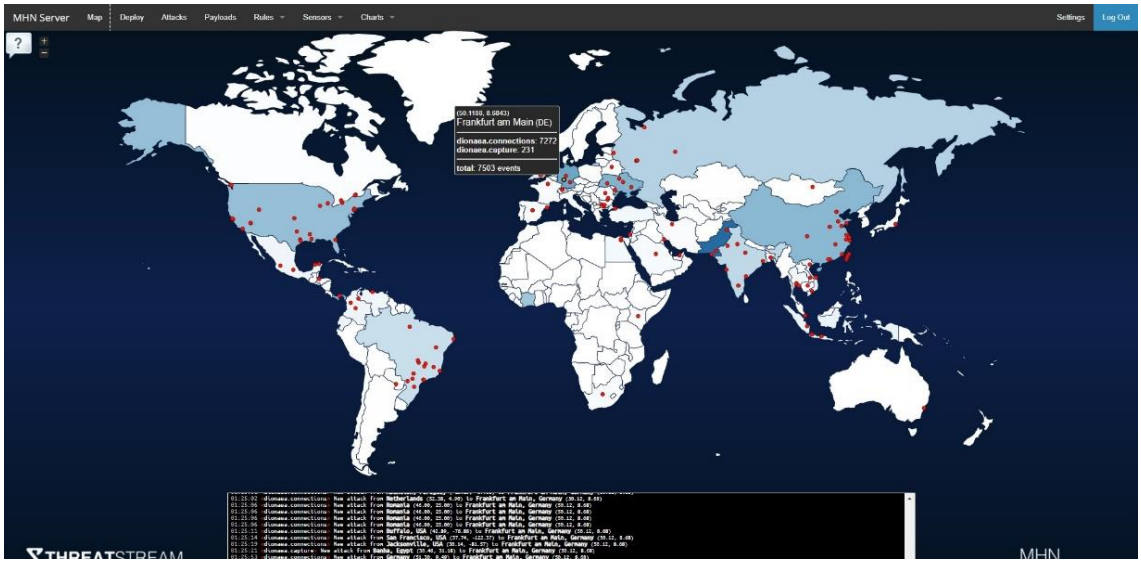


Obrázek 20 - Přehled typů útoků

Také jsme z logu schopni zjistit z jakých zdrojových portů bylo útočeno, ale variabilita je natolik velká, že nenajdeme v 7264 útocích téměř žádnou shodu. Většina těchto zdrojových portů využívá velmi vysoká čísla v rozmezí 10000 – 70000.

11.2.3 Honeymap

Díky MHN je možné vytvořit mapu útoků, kde je krásně graficky znázorněno, odkud útoky směřují a kolik útoků právě bylo provedeno z daného místa. Mapa je velmi detailní, protože umožňuje zobrazit počet útoků z daného státu, ale dokonce umožňuje zobrazit počet útoků z daného města. Naše mapa útoků vypadá přesně, jak je uvedeno na obrázku 21.



Obrázek 21 - Mapa útoků (ThreatStream)

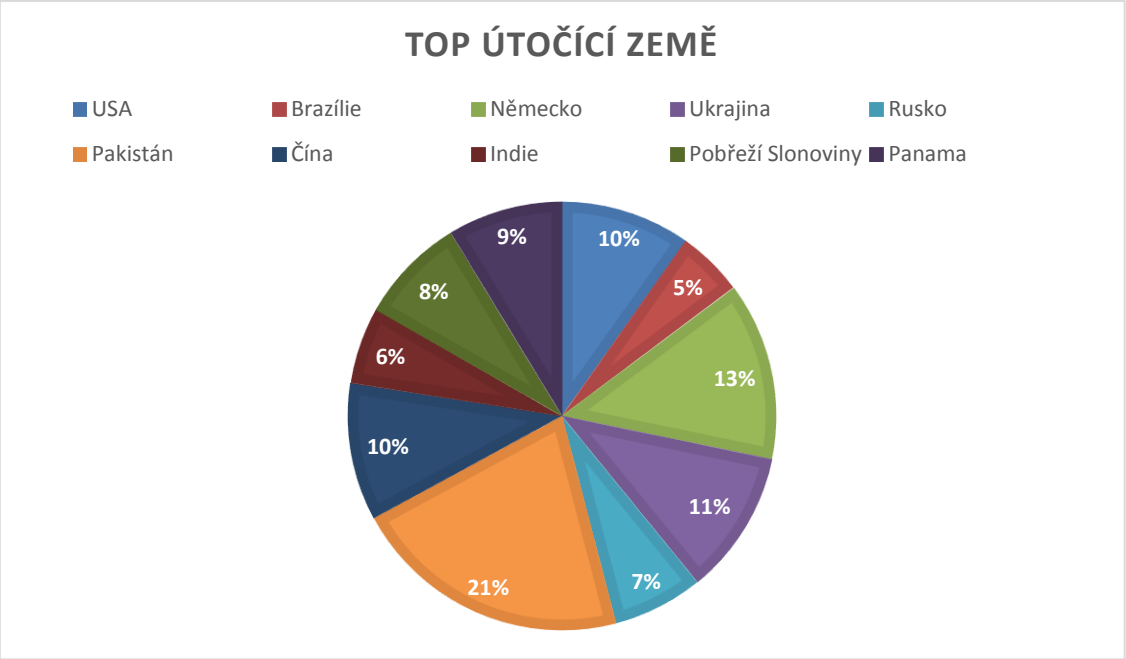
Pokud se podíváme na mapu detailněji, vidíme tabulku, která ukazuje umístění našeho serveru a je označena žlutým puntíkem. Ve spodní části jsme mohli vidět během zaznamenávání, aktuální útoky.

Díky této mapě a logu jsme mohli vytvořit přehled států, které mají největší podíl útoků na náš honeypot. Výběr těchto států dokonce tvoří 80,77 % z celkového počtu útoků.

Stát	Počet
Pakistán	1242
Německo	789
Ukrajina	638
Čína	607
USA	576
Panama	508
Pobřeží Slonoviny	473
Rusko	400
Indie	342
Brazílie	292
	5575

Tabulka 4 - Přehled útoků podle států

Pro lepší přehlednost tato tabulka byla zpracována opět do koláčového grafu, kde je vidět procentuální zastoupení hlavních útočících zemí s velkým podílem útoků.



Obrázek 22 - Přehled hlavních útočících zemí

Jak je na grafu vidět, největší podíl útoků připadá Pakistánu. Druhé místo obsadilo Německo, kde je server hostován a díky tomu byl počet útoků vyšší.

Pro detailnější přehled jsme si vytvořili tabulky, které nám ukazují zastoupení jednotlivých zemí v počtů útoků, včetně rozdělení do jednotlivých světadílů.

Stát	Počet
Amerika	
USA	576
Brazílie	292
Panama	508
Paraguay	28
Venezuela	12
Kolumbie	46
Mexiko	52
Guatemala	5

Tabulka 5 - Detailnější přehled počtu útoků (Amerika)

Afrika	
Keňa	9
Egypt	3
Jižní Afrika	15
Pobřeží Slonoviny	473

Tabulka 6 - Detailnější přehled počtu útoků (Afrika)

Oceánie	
Austrálie	12

Tabulka 7 - Detailnější přehled počtu útoků (Oceánie)

Asie	
Pakistán	1242
Čína	607
Indie	342
Japonsko	22
Mongolsko	7
Vietnam	128
Thajsko	88
Izrael	4
Irán	16
Saudská Arábie	27
Kuwait	9
Spojené Arabské Emiráty	25
Bangladéš	32
Kambodža	84
Malajzie	78
Taiwan	223

Tabulka 8 - Detailnější přehled počtu útoků (Asie)

Evropa	
Německo	789
Ukrajina	638
Rusko	400
Anglie	34
Francie	30
Španělsko	76
Rumunsko	112
Bělorusko	29
Turecko	19
Bulharsko	93
Rakousko	79
	7264

Tabulka 9 - Detailnější přehled počtu útoků (Evropa)

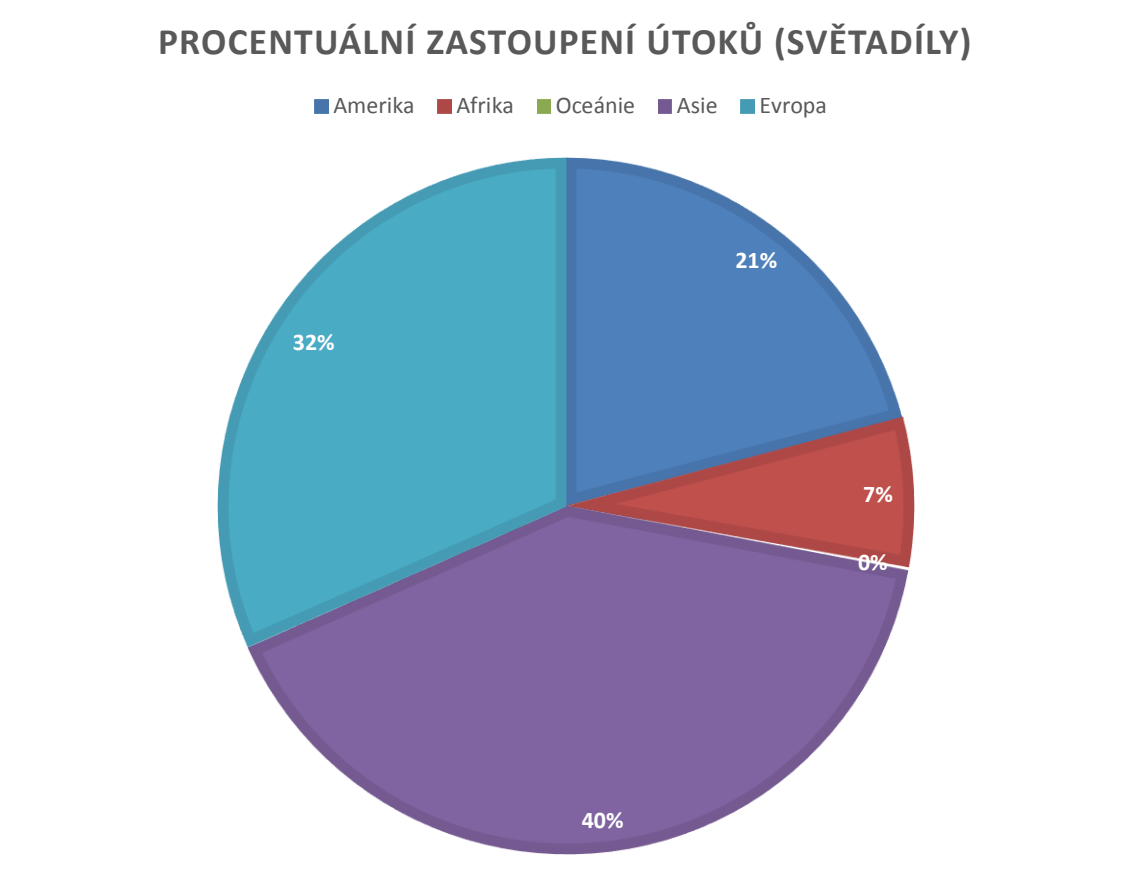
Abychom měli jasnou představu, kolik útoků směřuje z jednotlivých světadílů, sestrojili jsme si na základě našich nasbíraných dat tento přehled níže. Jak můžeme vidět největší procentuální zastoupení útoků je z Asie, následuje Evropa a jako třetí je Amerika. Počet útoků z Afriky a Oceánie je velmi malý. Afrika dosáhla 6 %,

protože hlavním útočníkem bylo Pobřeží Slonoviny s celkem vysokým počtem útoků. Pobřeží Slonoviny obsadilo celkově sedmé místo v top útočících zemích.

Světadíl	Počet	Procentuální zastoupení útoků %
Asie	2934	40,39
Evropa	2299	31,65
Amerika	1519	20,91
Afrika	500	6,88
Oceánie	12	0,17
	7264	**100,00**

Tabulka 10 - Přehled procentuálního zastoupení útoků jednotlivých světadílů

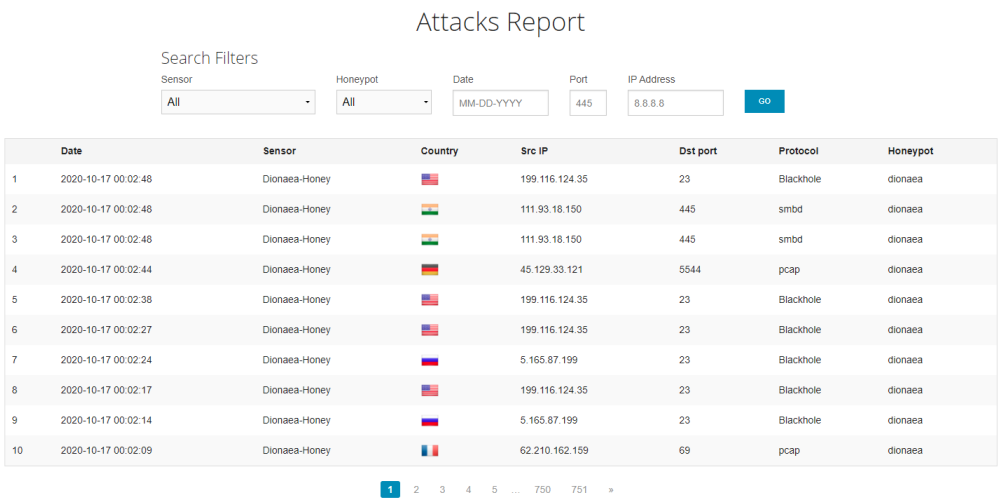
Pro lepší přehlednost jsme si opět sestavili koláčový graf, kde je krásně vidět procentuální zastoupení jednotlivých světadílů.



Obrázek 23 - Přehled procentuálního zastoupení útoků jednotlivých světadílů

11.2.4 Přehled útoků s možností filtrace

Ještě bychom si měli ukázat sekci Attacks v grafickém prostředí MHN, kde je možné jednotlivé útoky vidět a filtrovat na základě datumu, portu a zdrojové IP adresy.

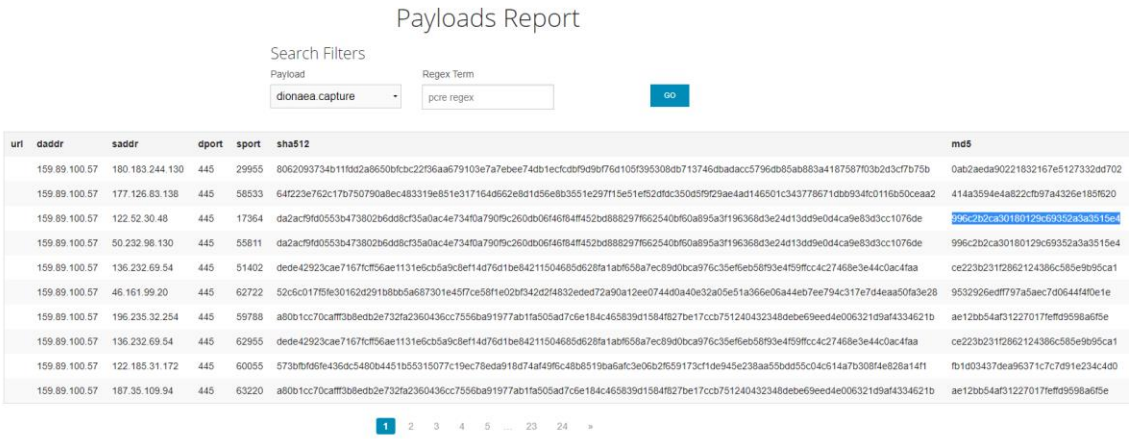


Obrázek 24 - Přehled MHN - sekce útoky (filtrace)

Díky tomu je snadné najít požadované typy útoků nebo konkrétního útočníka z určité IP adresy. Možné je filtrovat i senzory, které naše MHN má, ale my jsme si nastavili pouze jeden a tím je honeypot Dionaea, takže v našem případě tento filtr nemá význam, ale je dobré vědět, že lze mezi jednotlivými senzory přepínat.

11.2.5 Přehled zachyceného malwaru

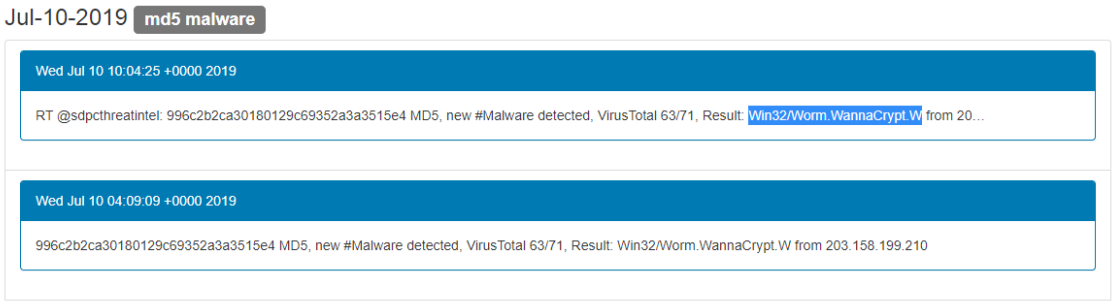
Pokud se v grafickém prostředí MHN přepneme do sekce Payloads, uvidíme tabulku, kde je vidět vždy cílová adresa, zdrojová adresa, cílový port, zdrojový port a md5 kód malwaru. Díky md5 kódu můžeme zjistit o jaký typ malwaru se jedná. Toto lze zjistit jak z grafického prostředí MHN, ale i z vytvořeného logu.



Obrázek 25 - Přehled Payloads

Vezmeme si například označený md5 kód, který je **996c2b2ca30180129c69352a3a3515e4**. Nyní na základě tohoto md5 kódu jsme

schopni vyhledat co reprezentuje za odpovídající malware. V našem případě se jedná o Win32/Worm.WannaCrypt.W. Na přiloženém obrázku níže vidíme detailní informace, které jsou spojeny s tímto md5 kódem a zároveň pro tento typ malwaru.



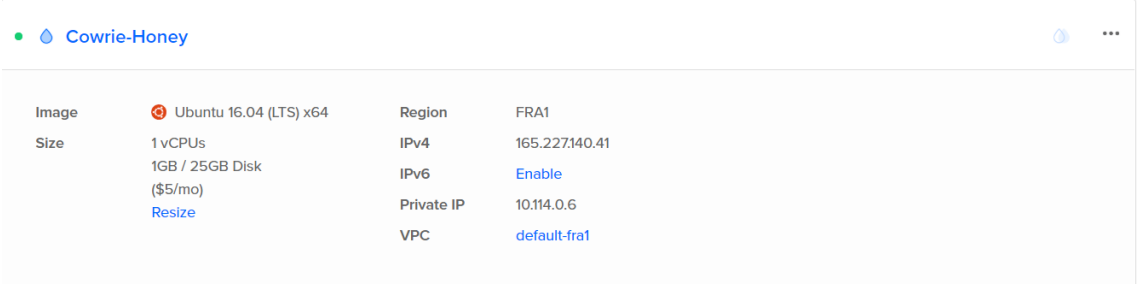
Obrázek 26 - md5 typ malwaru

11.3 Honeypot Cowrie (Standalone verze)

Cowrie je středně až vysoce interakční SSH a Telnet honeypot určený k zaznamenávání útoků hrubou silou neboli (brute-force útokům) a interakcím prováděné útočníkem v shellu. Honeypot Cowrie byl vyvinut z honeypotu Kippo. My si budeme instalovat Cowrie na Ubuntu verze 16.04, na kterém by mělo být podporováno. Podle dokumentace k Cowrie budeme také potřebovat nainstalovat Python verze 3.5+ a nástroj python-virtualenv.

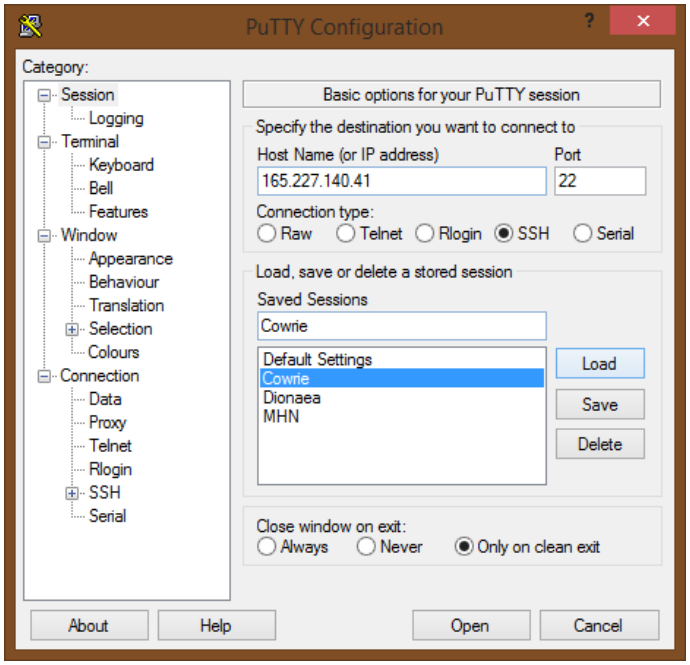
11.3.1 Instalace

Vytvoříme si stejně jako u MHN a u honeypotu Dionaea droplet na Digital Ocean a nyní s verzí Ubuntu 16.04 s architekturou x64. Specifikace není potřeba tak náročná jako tomu bylo u MHN a Dionaea honeypotu, proto si zvolíme 1GB / 1CPU a k tomu 25GB SSD. Stejně jako u ostatních vybereme nejbližší server ve Frankfurtu. Na obrázku níže, můžeme vidět celou specifikaci včetně přidělené IP adresy.



Obrázek 27 - Specifikace vytvořeného dropletu pro honeypot Cowrie

Připojení ke stroji funguje úplně stejně jako tomu bylo u MHN a honeypotu Dionaea. Budeme se připojovat přes již zmiňovaný software Putty, kde nám byla přidělena IP adresa 165.227.140.41.



Obrázek 28 - Připojení pomocí Putty k Cowrie dropletu

Po přihlášení pomocí oprávnění root provedeme již známý příkaz pro update a upgrade, kterým zkontrolujeme dostupné aktualizace pro naše nainstalované balíky a následně je i zaktualizujeme, abychom se vyvarovali případných chyb.

$ apt-get update && apt-get upgrade

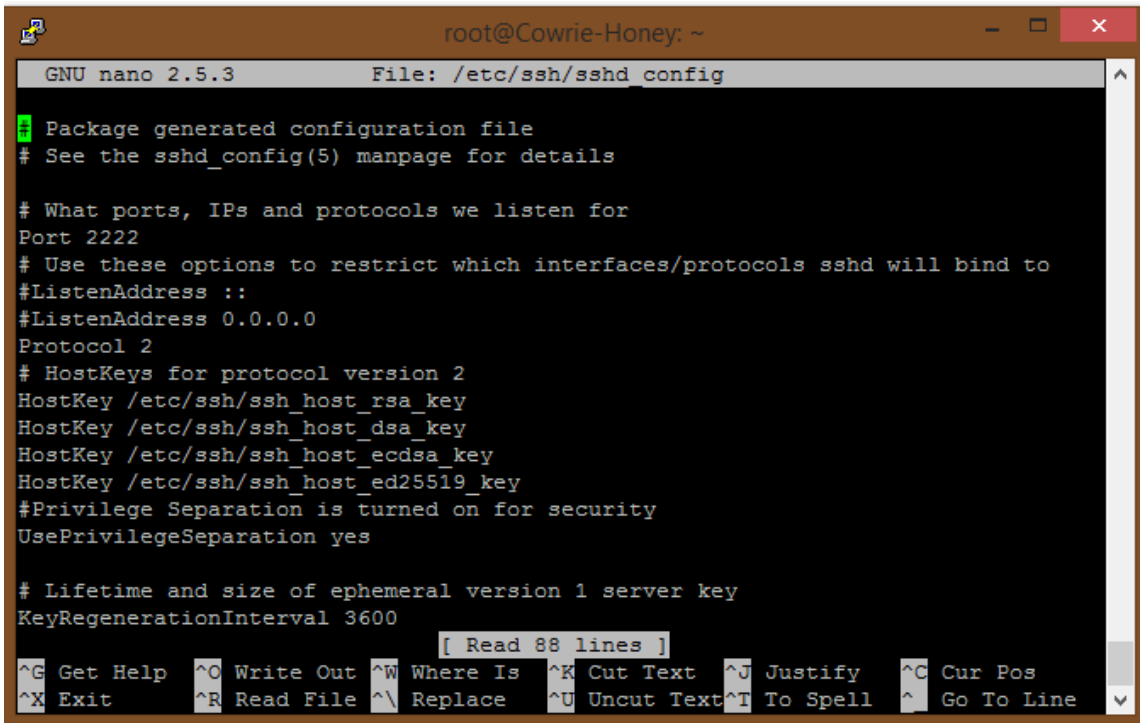
Ještě, než začneme s instalací Cowrie honeypotu, tak je nutné si říci, že náš SSH démon poběží na vysokém portu 2222, zatímco Cowrie poběží na portu 22. Díky tomu útočník útočící na port 22, bude přesměrován na náš honeypot.

Změníme si tedy port 22 (výchozí SSH port) na port 2222, takže bot nebo útočník si bude myslet, že je připojen na reálném SSH portu. Pro tuto změnu musíme změnit parametry v souboru sshd_config, který najdeme v

/etc/ssh/sshd_config. Použijeme pro to textový editor nano a příkaz tedy bude vypadat následovně.

$ sudo nano /etc/ssh/sshd_config

Po spuštění tohoto příkazu se otevře textový editor, kde vidíme obsah souboru sshd_config. Najdeme si položku port 22 a změníme ji na port 2222. Uložíme a zavřeme.



Obrázek 29 - Obsah souboru sshd_config

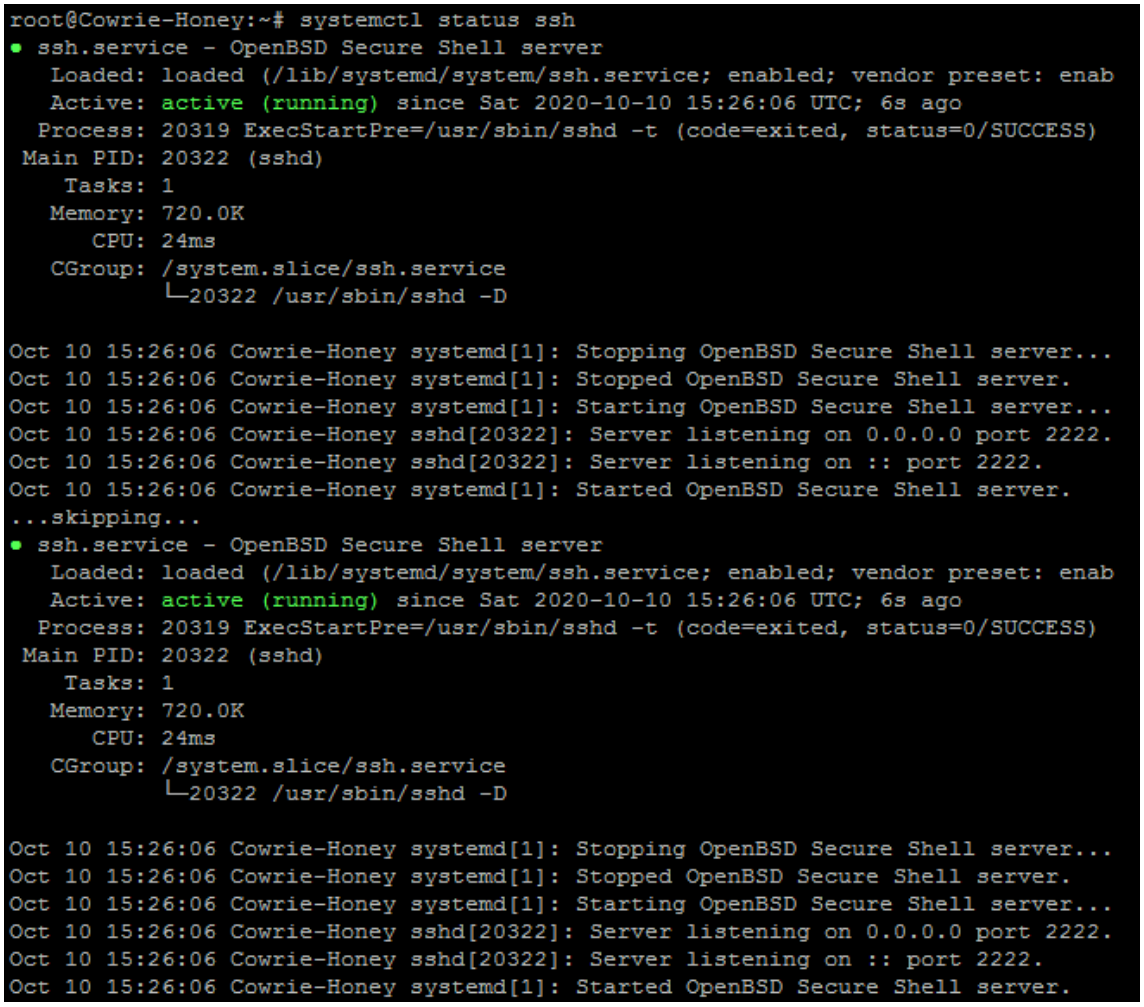
Po uložení je nutné restartovat SSH, aby se změny provedli. Toho docílíme pomocí příkazu:

$ systemctl restart ssh

Následně si můžeme zkontrolovat, zda nám služba SSH běží správně pomoci příkazu:

$ systemctl status ssh

Správně běžící SSH, by mělo vypadat jako na obrázku 30.



Obrázek 30 - Kontrola běžícího SSH na portu 2222

Nyní se dostáváme už k samotné instalaci Cowrie honeypotu. Jako první si musíme nainstalovat všechny závislosti (dependencies).

$ sudo apt-get install git python-virtualenv libssl-dev build-essential libpython-dev python2.7-minimal authbind

Následně si vytvoříme uživatele, který bude použit pro přístup útočníkovi. Zvolíme si například účet bob s heslem bob, aby bylo heslo pro útočníka lehce prolomitelné.

$ sudo adduser bob

```
root@Cowrie-Honey:~# sudo adduser bob
Adding user `bob' ...
Adding new group `bob' (1000) ...
Adding new user `bob' (1000) with group `bob' ...
Creating home directory `/home/bob' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bob
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@Cowrie-Honey:~#
```

Obrázek 31 - Vytvoření uživatele bob

Když je účet vytvořen, přepneme se na tento účet a stáhneme si kód pro Cowrie.

$ su bob

$ cd

$ git clone http://github.com/micheloosterhof/cowrie

Úspěšné stažení kódu by mělo vypadat jako na obrázku 32.

```
bob@Cowrie-Honey:~$ git clone http://github.com/micheloosterhof/cowrie
Cloning into 'cowrie'...
remote: Enumerating objects: 13786, done.
remote: Total 13786 (delta 0), reused 0 (delta 0), pack-reused 13786
Receiving objects: 100% (13786/13786), 8.79 MiB | 3.12 MiB/s, done.
Resolving deltas: 100% (9522/9522), done.
Checking connectivity... done.
bob@Cowrie-Honey:~$
```

Obrázek 32 - Stažení kódu Cowrie

Po stažení musíme vytvořit virtuální prostředí pro Python a Cowrie a aktivovat jej. Prostředí si pojmenujeme Cowrie-env.

$ cd cowrie/

$ virtualenv cowrie-env

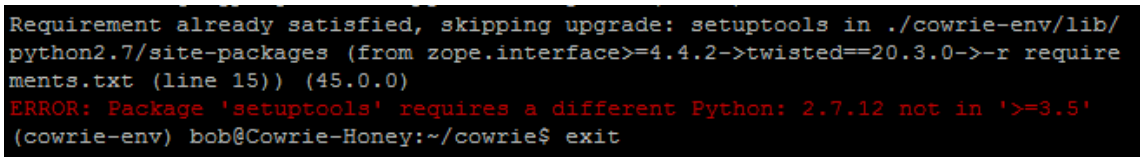
$ source cowrie-env/bin/activate

Po zadání těch příkazů bychom měli mít vytvořené a aktivované virtuální prostředí. Poté potřebujeme nainstalovat balíčky Pythonu, které Cowrie potřebuje pro spuštění.

$ pip install --upgrade pip

$ pip install --upgrade -r requirements.txt

Jak můžeme vidět na obrázku 33. Skončili jsme chybovou hláškou, že balíček setuptools vyžaduje jinou verzi Pythonu, než máme nyní nainstalovanou.

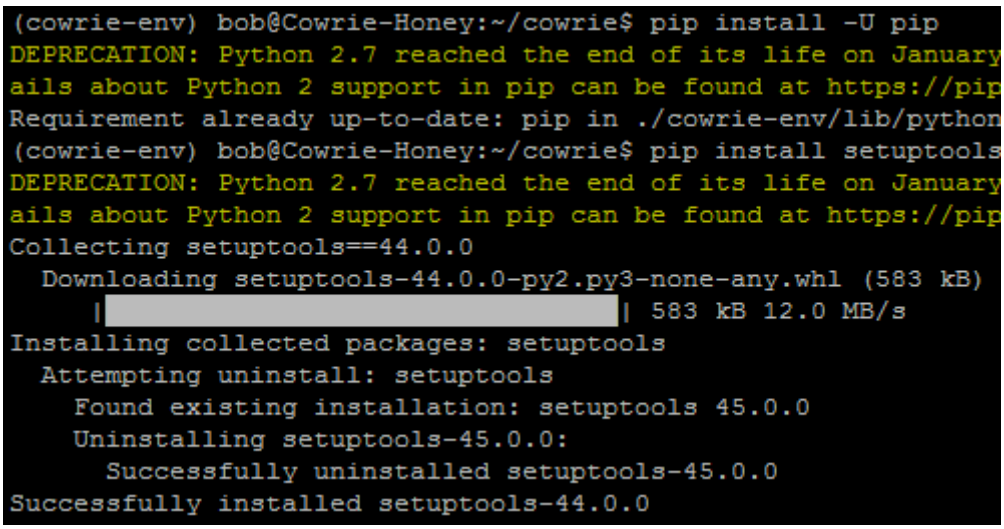


Obrázek 33 - Chybová hláška setuptools

Hláška je trochu zavádějící, protože potřebujeme pouze starší verzi setuptools 44.0.0. Stačí tedy tuto verzi nainstalovat a příkaz pro instalaci requirements zopakovat. Pro instalaci setuptools v 44.0.0 provedeme následující příkazy.

$ pip install -U pip

$ pip install setuptools==44.0.0



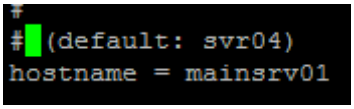
Obrázek 34 - Instalace setuptools v44.0.0

Jako další potřebujeme nakonfigurovat démona Cowrie. Musíme proto vytvořit kopii výchozího nastavení souboru cowrie.cfg.dist a pojmenovat ji cowrie.cfg, protože Cowrie ve výchozím nastavení hledá právě takto pojmenovaný soubor, pokud nenajde soubor cowrie.cfg, bude se řídit nastavením souboru cowrie.cfg.dist. Soubor nalezneme ve složce etc/.

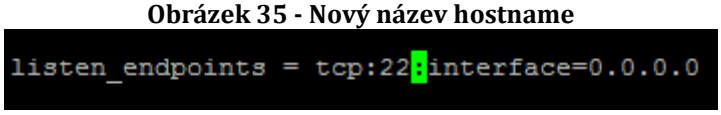
$ cd etc/

$ cp cowrie.cfg.dist cowrie.cfg

Následně si soubor otevřeme a provedeme změny. Vhodné je změnit hostname, aby jméno serveru vypadalo pro útočníka zajímavě a také port, na kterém bude honeypot poslouchat. Náš název bude mainsrv01 a port si změníme na požadovaný port 22, jak jsme si definovali již na začátku.



Obrázek 35 - Nový název hostname



Obrázek 36 - Poslech honeypotu pro SSH na portu 22

Po každé změně konfigurace je nutné restartovat Cowrie, aby se provedli změny. Můžeme to provést pomoci rebootu celého stroje. Použijeme tedy příkaz reboot jako root.

$ sudo reboot

Jako poslední krok je nutné zadat následující příkazy, aby neroot uživatel mohl naslouchat na portu 22, protože ve výchozím nastavení nemůžeme spustit Cowrie jako root.

$ sudo apt-get install authbind

$ sudo touch /etc/authbind/byport/22

$ sudo chown bob:bob /etc/authbind/byport/22

$ sudo chmod 770 /etc/authbind/byport/22

Tímto je instalace a základní nastavení Cowrie hotové. V další části si ještě nainstalujeme Snort na Cowrie, abychom si mohli nechávat vypisovat aktuální útoky a ukážeme si, jak si na Cowrie zaútočit.

11.4 Snort – IDS

Snort je jedním z nejčastěji používaných síťových IDS, jelikož je podporován na mnoho platformách a má otevřený zdrojový kód. Monitoruje data, které byli odeslané a přijaté skrze konkrétní síťové rozhraní. NIDS může zachytit hrozby zaměřené na zranitelnosti systému pomocí technologií detekce a analýzy protokolů založených na podpisu. Pokud je NIDS nainstalován a nakonfigurován správně, dokáže identifikovat nejnovější útoky, malware, napadené systémy a porušení zásad v síti.

11.4.1 Instalace

Nejprve si musíme nainstalovat veškerý potřebný software na naši stanici předtím, než budeme instalovat samotný Snort. To zařídíme tímto příkazem:

**$ sudo apt install -y gcc libpcre3-dev zlib1g-dev libluajit-5.1-dev **
**libpcap-dev openssl libssl-dev libnghttp2-dev libdumbnet-dev **
bison flex libdnet autoconf libtool

Pokud máme splněny předpoklady pro instalaci díky předchozímu příkazu, můžeme začít s instalací Snortu, který lze stáhnout a nainstalovat ručně přímo ze zdroje Snortu. Samotná instalace se skládá z několika kroků, kterými si nyní projdeme. Nejprve musíme stáhnout kód, provést konfiguraci, kompilaci kódu, instalace do příslušného adresáře, a nakonec konfigurace pravidel detekce, kterými se Snort řídí. Ještě je důležité zmínit, že před samotnou instalací Snortu je potřeba nainstalovat také Data Acquisition Library (DAQ), který potřebuje.

Prvním krokem je tedy vytvoření dočasné složky pro stažení a přepnutí se do této složky.

$ mkdir ~/snort_src && cd ~/snort_src

Druhým krokem je stažení Data Acquisition Library (DAQ), které Snort používá, aby provedl abstraktní volání knihoven pro zachycování paketů. DAQ stáhneme také přímo z webu Snortu, pomocí příkazu wget.

$ wget https://www.snort.org/downloads/snort/daq-2.0.7.tar.gz

Stahování většinou trvá několik vteřin. Když je stahování dokončené, extrahujeme zdrojový kód a přepneme se do vytvořené složky s kódem.

$ tar -xvzf daq-2.0.7.tar.gz

$ cd daq-2.0.7

Poslední verze DAQ vyžaduje další krok pro automatickou rekonfiguraci DAQ, před spuštěním konfigurace. To zařídíme pomoci následujícího příkazu. Důležité je ještě zmínit, že potřebujeme mít nainstalovaný software libtool a autoconf, který jsme si však nainstalovali ihned na začátku.

$ autoreconf -f -i

Následně spustíme konfigurační skript s jeho výchozími hodnotami, poté program zkompilujeme pomocí příkazu make a nakonec nainstalujeme samotný DAQ.

$ ./configure && make && sudo make install

Nyní, když máme nainstalovaný DAQ, tak můžeme začít s instalací Snortu. Vrátíme se zpět do složky, kam jsme si stahovali DAQ v našem případě snort_src.

$ cd ~/snort_src

Konečně stáhneme z oficiálního webu pomocí wget aktuální verzi Snortu. Aktuální verze je 2.9.16.1.

$ wget https://www.snort.org/downloads/snort/snort-2.9.16.1.tar.gz

Opět rozbalíme stažený soubor a přepneme se do něj.

$ tar -xvzf snort-2.9.16.1.tar.gz

$ cd snort-2.9.16.1

A nyní provedeme instalaci s povoleným sourcefire. Takový příkaz pro instalaci vypadá následovně.

$ ./configure --enable-sourcefire && make && sudo make install

11.4.2 Konfigurace

Jak již bylo řečeno, nyní si nakonfigurujeme Snort, pro náš systém, což zahrnuje editaci několika konfiguračních souborů. Začneme s aktualizací sdílených knihoven.

$ sudo ldconfig

Snort na Ubuntu se nám nainstaluje do umístění /usr/local/bin/snort. Je dobrou zvyklostí vytvořit symbolický odkaz do /usr/sbin/snort

$ sudo ln -s /usr/local/bin/snort /usr/sbin/snort

Pro bezpečné spuštění Snortu na Ubuntu bez root oprávnění musíme vytvořit nového neprivilegovaného uživatele a novou skupinu uživatelů, aby démon mohl běžet pod ním. Vytvoříme si tedy uživatele a skupinu s názvem snort.

$ sudo groupadd snort

$ sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort

Následně vytvoříme strukturu složek pro uložení konfigurace Snortu.

$ sudo mkdir -p /etc/snort/rules

$ sudo mkdir /var/log/snort

$ sudo mkdir /usr/local/lib/snort_dynamicrules

Poté si nastavíme odpovídající oprávnění k nově vytvořeným složkám.

$ sudo chmod -R 5775 /etc/snort

$ sudo chmod -R 5775 /var/log/snort

$ sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules

$ sudo chown -R snort:snort /etc/snort

$ sudo chown -R snort:snort /var/log/snort

$ sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules

Ještě si musíme vytvořit soubory pro whitelist a blacklist, stejně jako i pro místní pravidla.

$ sudo touch /etc/snort/rules/white_list.rules

$ sudo touch /etc/snort/rules/black_list.rules

$ sudo touch /etc/snort/rules/local.rules

Nakonec si nakopírujeme konfigurační soubory ze stažené složky se Snortem.

$ sudo cp ~/snort_src/snort-2.9.16.1/etc/*.conf /etc/snort

$ sudo cp ~/snort_src/snort-2.9.16.1/etc/*.map /etc/snort

11.4.3 Pravidla a další konfigurace

Dále budeme potřebovat stáhnout pravidla detekce, která bude Snort dodržovat, aby identifikoval potenciální hrozby. Snort poskytuje tři úrovně sad pravidel a to komunitní, registrovaná a předplatitelská. My budeme využívat pravidel komunitních, protože jsou volně dostupná na rozdíl od pravidel registrovaných a předplatitelských. Pokud by se však člověk potřeboval zabývat Snortem více do hloubky, určitě je vhodné zvážit minimálně pravidla registrovaná, ke kterým se dostaneme po základní registraci.

Jak tedy na instalaci komunitních pravidel. Nejprve si je stáhneme pomocí nástroje wget z oficinálního webu.

$ wget https://www.snort.org/rules/community -O ~/community.tar.gz

Následně rozbalíme a nakopírujeme do naší vytvořené složky pro konfiguraci.

$ sudo tar -xvf ~/community.tar.gz -C ~/

$ sudo cp ~/community-rules/* /etc/snort/rules

Ještě potřebujeme změnit konfigurační soubor Snortu, kde ve výchozím nastavení Snort na Ubuntu očekává, že najde řadu různých pravidel, které nejsou zahrnuty v komunitních pravidlech. Proto nepotřebné řádky můžeme snadno zakomentovat pomocí následujícího příkazu.

$ sudo sed -i 's/include \$RULE_PATH/#include \$RULE_PATH/' /etc/snort/snort.conf

Po nastavení konfiguračních souborů a pravidel ještě musíme upravit soubor snort.conf a v něm několik parametrů. Otevřeme si tedy konfigurační soubor v nějakém textovém editoru. Já rád používám nano, takže si to ukážeme skrze nano.

$ sudo nano /etc/snort/snort.conf

Uvnitř si najdeme položku ipvar HOME_NET any a změníme any za požadovanou ip adresu našeho stroje, kde chceme, aby Snort detekoval případné útoky. Stejně tak položku ipvar EXTERNAL_NET místo any změníme na !$HOME_NET viz. obrázek 37.

```
###################################################
# Step #1: Set the network variables.  For more information, see README.variables
###################################################

# Setup the network addresses you are protecting
ipvar HOME_NET 165.227.140.41/32

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET !$HOME_NET
```

Obrázek 37 - Konfigurace1 snort.conf

Ještě je potřeba nastavit správné cesty k pravidlům a kde je náš whitelist a blacklist. Naše nastavení vypadá stejně jako na obrázku 38.

```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# If you are using reputation preprocessor set these
# Currently there is a bug with relative paths, they are relative to where snort is
# not relative to snort.conf like the above variables
# This is completely inconsistent with how other vars work, BUG 89986
# Set the absolute path appropriately
var WHITE_LIST_PATH /etc/snort/rules
var BLACK_LIST_PATH /etc/snort/rules
```

Obrázek 38 - Konfigurace2 snort.conf

V souboru nastavíme také jak má vypadat output pro logy a odkomentujeme tento řádek, aby byl parametr brán v potaz.

```
# unified2
# Recommended for most installs
output unified2: filename snort.log, limit 128
```

Obrázek 39 - Konfigurace3 snort.conf

A ještě nakonec si odkomentujeme parametr include $RULE_PATH/loacl.rules a přidáme si navíc include $RULE_PATH/community.rules viz. obrázek 40.

```
# site specific rules
include $RULE_PATH/local.rules
include $RULE_PATH/community.rules
```

Obrázek 40 - Konfigurace4 snort.conf

11.4.4 Validace

V této fázi by měl být Snort plně funkční, ale pro jistotu si ho ještě zvalidujeme, zda se zde nenachází nějaká chyba, kterou je potřeba opravit.

$ sudo snort -T -c /etc/snort/snort.conf

Po dokončení validace bychom měli dostat následující zprávu. Pokud se však zpráva liší, pravděpodobně je chyba v snort.conf nebo nějaký chybějící adresář či soubor, který jsme mohli zapomenout vytvořit během konfigurace.

```
        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.16.1 GRE (Build 140)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2020 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.38 2015-11-23
           Using ZLIB version: 1.2.8

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>

Snort successfully validated the configuration!
Snort exiting
root@Cowrie-Honey:~/snort_src/snort-2.9.16.1#
```

Obrázek 41 - Validace

11.4.5 Testování

Nakonec si vyzkoušíme přidat pravidlo pro příchozí ICMP, abychom zjistili, zda opravdu funguje vše, jak má. Potřebujeme tedy otevřít soubor local.rules.

$ sudo nano /etc/snort/rules/local.rules

Do souboru přidáme následující pravidlo.

alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)

Toto pravidlo obsahuje následující:

- Akce pro provoz odpovídající pravidlu v tomto případě alert
- Komunikační protokol jako TCP, UDP nebo ICMP jako v našem případě
- Zdrojovou adresu a port, pro zjednodušení používáme any pro zahrnutí všech adres a portů
- Cílová adresa a port, pro nás předefinované $HOME_NET, které jsme definovali v konfiguraci a any pro jakýkoliv port
- Nějaké další bity
    - Zpráva logu
    - Unikátní ID pravidla (sid), které musí být 1000001 nebo vyšší pro místní pravidla
    - Číslo verze pravidla

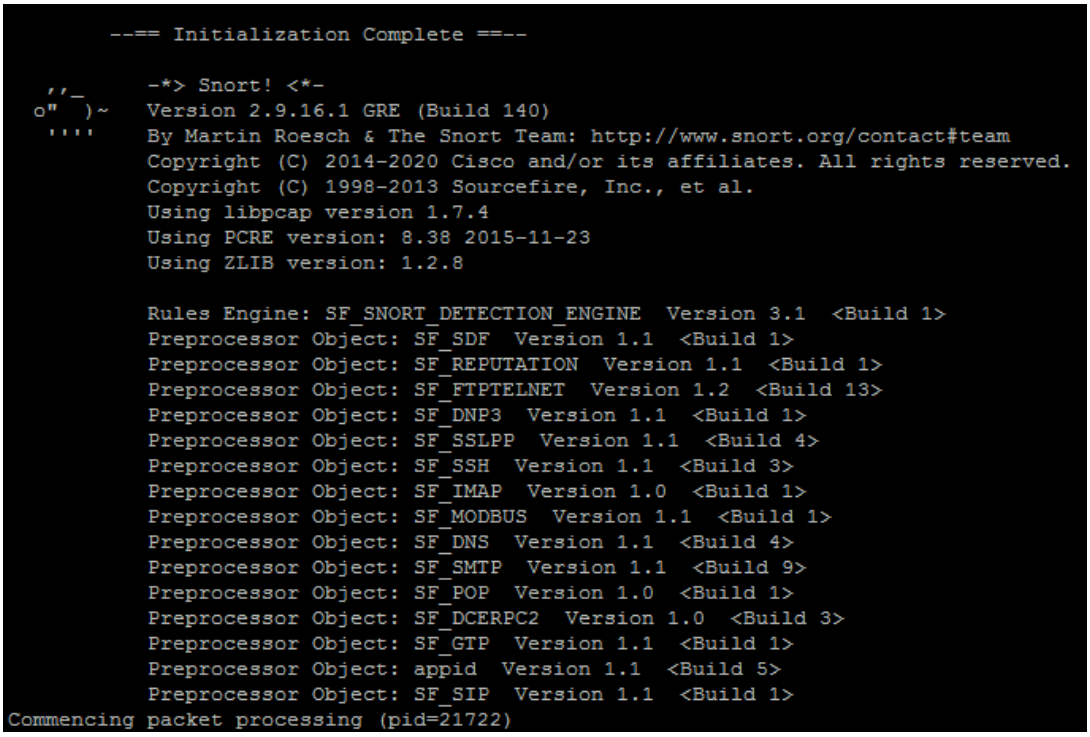
Pravidlo uložíme a spustíme Snort, aby hlídal pravidlo, které jsme si nastavili v local.rules pomoci následujícího příkazu.

$ sudo snort -A console -i eth0 -u snort -g snort -c /etc/snort/snort.conf

V příkazu je nutné zadat správné rozhraní. To si můžeme zkontrolovat pomoci příkazu:

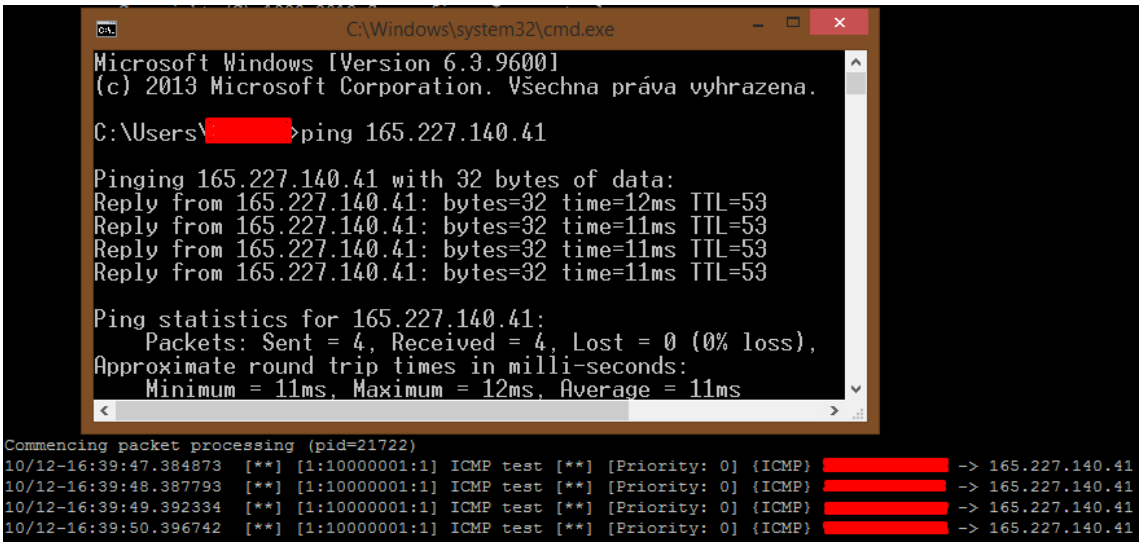
$ ip address

Pokud Snort naslouchá měl by vypadat stav jako na obrázku 42.



Obrázek 42 - Úspěšné spuštění Snortu

A na závěr si vyzkoušíme, zda nám opravdu vše funguje, jak má jednoduchým pingem například z naší stanice, protože Cowrie je hostováno na veřejném serveru a je díky tomu na dané IP adrese dostupné.



Obrázek 43 - Ověření pravidla pomocí ICMP

Jak vidíme na obrázku 43. tak Snort ping zaznamenal, takže vše funguje správně. Veškeré tyto zprávy jsou zaznamenávány do logu v umístění /var/log/snort/snort.log. Log si můžeme zobrazit následujícím příkazem.

$ snort -r /var/log/snort/snort.log

11.4.6 Útoky na Cowrie za pomoci detekce Snortu

V této kapitole se podíváme, jak detekovat útoky, které směřují na náš honeypot Cowrie. Budeme detekovat konkrétně SSH útoky na portu 22. Musíme si proto vytvořit nové pravidlo, které bude detekovat SSH útoky a přidat ho do souboru, který je umístěn v této lokaci /etc/snort/rules/local.rules. Soubor si otevřeme například pomocí nano editoru.

$ sudo nano /etc/snort/rules/local.rules

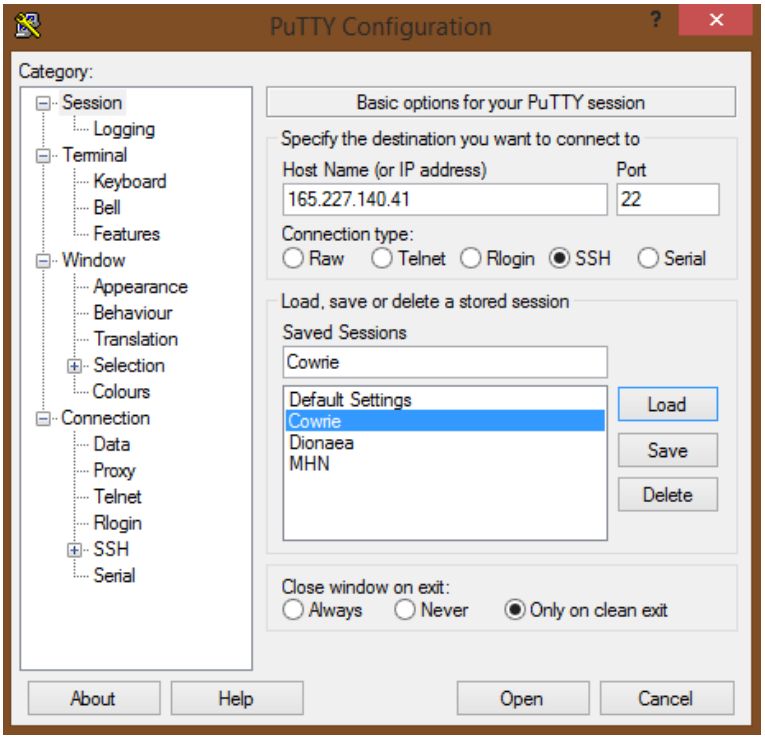
Tvorbu pravidel jsme si ukazovali v kapitole 11.4.5, kdy jsme si tvořili pravidlo pro ICMP pakety. Nyní si podle stejných zásad vytvoříme právě pravidlo pro SSH spojení na portu 22. Pravidlo by mělo vypadat takto:

alert tcp any any -> $HOME_NET 22 (msg:"SSH login attempt"; sid:10000002; rev:001;)

Pravidlo uložíme a spustíme Snort, který začne detekovat obě pravidla, jak ICMP pakety, tak SSH spojení. Příkaz pro spuštění je stejný.

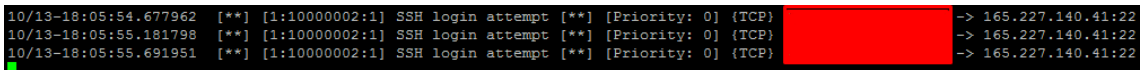
$ sudo snort -A console -i eth0 -u snort -g snort -c /etc/snort/snort.conf

Nyní si zkusíme na náš honeypot Cowrie zaútočit. Otevřeme si program Putty a zkusíme navázat spojení na portu 22.



Obrázek 44 - Nastavení Putty pro SSH komunikaci na portu 22

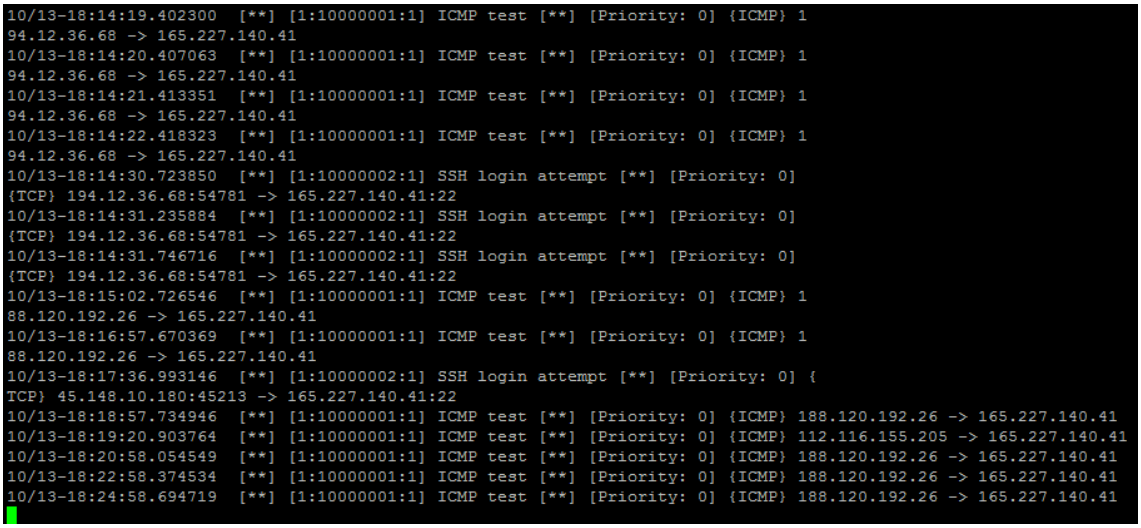
Po otevření komunikace na portu 22 nám náš honeypot Cowrie s nadstavbou Snortu detekuje tento pokus se přihlásit na port 22.



Obrázek 45 - Záznam o pokus navázání SSH spojení na portu 22

Jak můžeme vidět na obrázku 45, Snort ihned detekoval náš pokus se přihlásit. Stejným způsobem pak můžeme vytvářet jakákoliv pravidla. V našem případě by mělo ještě smysl pravidlo pro připojení přes Telnet na portu 23, protože Cowrie je SSH a Telnet honeypot.

Pokud necháme po nějakou dobu takto spuštěný Snort na Cowrie, zachytíme i jiné útoky, než které jsme demonstrovali samy.



Obrázek 46 - Výpis útoků

Jak demonstruje obrázek 46, jedná se většinou o sken sítě, kdy nám Snort zachycuje hlavně ICMP pakety.

12 Vyhodnocení hypotéz

Nyní si vyhodnotíme již dříve stanovené hypotézy na základě provedeného testování honeypotů.

12.1 Hypotéza 1

Hypotézu zamítáme, protože v našem případě na honeypot Dionaea dopadlo pouze 79 útoků typu SipSession, což činí pouhých 1,09 % celkového počtu všech typů útoků.

12.2 Hypotéza 2

Přijímáme hypotézu, protože v našem případě na honeypot Dionaea dopadly útoky zejména z Asie, a to v procentuálním zastoupení 40,39 % z celkového počtu všech útoků z celého světa.

13 Závěry a doporučení

Hlavním cílem této diplomové práce bylo představit, jak se dají využít honeypoty pro síťové zabezpečení. Na základě toho jsme si právě ukázali, jak nainstalovat MHN, Cowrie a Dionaeu a to včetně jejich konfigurace a nasazení.

V práci jsme si stanovili dvě hypotézy, které byli stěžejní při našem testování. Hypotézy se zaměřovali hlavně na honeypot Dionaea, kde jsme prováděli detailnější testování v praxi.

První hypotéza byla, zda typ útoku SipSession na honeypot Dionaea tvoří víc jak 40 % celkových útoků všech typů, které na honeypot dopadli. Tato otázka byla velmi důležitá, protože jiní autoři, jak bylo uvedeno v rešerši před praktickou částí, dosahovali různých výsledků. Proto jsme na základě Bova, stanovili právě tuto hypotézu. Bove uváděl, že na jeho honeypot dopadlo okolo 45 % útoků tohoto typu. Bohužel se nám, ale tato hypotéza nepotvrdila a museli jsme ji zamítnout. Na náš honeypot dopadlo pouze 1,09 % SipSession útoků. Naopak jsme dosáhli až 55,56 % útoků na protokol smbd, který Bove zmiňoval také, ale u něho to činilo pouze 1,87 %. Na základě tohoto zkoumání jsme zjistili, že musíme tedy systémy bránit proti všem typům útoků, protože není vždy jasné, jaký typ protokolu bude nejvíce napadán.

Druhá hypotéza byla, zda útoky na honeypot Dionaea s umístěním serveru v Německu pochází výhradně z Asie a zda tyto útoky tvoří více jak 40 % celkových útoků. Tuto otázku jsme si stanovili, jelikož ostatní autoři, kteří se zabývali stejnou problematikou výzkumu, vždy prováděli výzkum z jiných částí světa. Proto nás tedy zajímalo, zda umístění serveru s honeypotem má vliv na počet útoků z jednotlivých světadílů. Většinou autoři uváděli, že útoky pocházeli z Číny a Pakistánu, které právě spadají do Asie. Nám se díky testování tato hypotéza potvrdila, protože mezi hlavními útočníky na náš server byli také Čína a Pakistán.

Celkově tedy útoky z Asie u nás představovali 40,39 % všech útoků. Samozřejmě, zde bylo vcelku vysoké zastoupení útoků i z Evropy, kdy na server dopadlo 31,64 % všech útoků. Hlavní země z Evropy, které utočili byli Německo, kde byl server hostovaný, Ukrajina a Rusko. Na základě tohoto testování jsme zjistili, že je jedno, kde je server umístěn, protože útoky jsou celosvětové a je tedy nutné systém dobře zabezpečit, ať je umístěn kdekoliv.

Na závěr si shrneme ještě důležité aspekty, během našeho testování. Na náš honeypot dopadlo za celých 24 hodin neskutečných 7264 útoků, což je vcelku velký počet útoků. Díky zachytávání jednotlivých útoků, můžeme analyzovat typy útoků, IP adresy, ze kterých nejčastěji útoky přicházejí a díky tomu následně efektivněji bránit náš systém. Jelikož honeypot Dionaea zachytává útoky na jednotlivé protokoly, které se využívají k šíření malwaru do jiných systémů, můžeme tak narazit na úplně nový typ malwaru, který ještě není znám tzv. zero-day útok. Závěrem tedy plyne, že využívání honeypotů má rozhodně své opodstatnění, a to zejména ve společnostech, kde je kladen velký důraz na zabezpečení jako jsou velké korporace nebo například různé bankovní sektory, kde to platí dvakrát tolik. Honeypoty však mohou být nasazeny i v menších společnostech a díky nim mohou být účinně detekovány hrozby a prováděny následující bezpečnostní opatření. Honeypoty mohou být zejména dobrým sluhou, pokud je administrátor umí správně nakonfigurovat a mohou tak pomáhat proti kybernetickým útokům. Dobré je pak kombinovat různé honeypoty s dalšími zabezpečovacími systémy, které jsme si zde prezentovali také, jako například IDS a IPS či firewally. V naší praktické části jsme si ukázali, jak funguje jeden z IDS, a to konkrétně Snort.

Doporučení, které plyne z této diplomové práce je nebát se používání honeypotů a konfigurace s nimi spojenou, ačkoliv to může na začátku vypadat složitě. Důležité je, vždy kontrolovat, zda honeypot funguje přesně jak chceme, než ho vypustíme do ostrého systému, kde chceme útoky zaznamenávat, protože se může stát, že špatně nastavený honeypot může být zdrojem pro útok na náš reálný systém, pokud není správně oddělen od produkční sítě. Z toho lze tedy dojít k závěru, že honeypoty mohou organizaci poskytnout užitečné informace a stojí za to je nasadit a využívat.

14 Citovaná literatura

1. **TechDifferences.** Difference between TCP/IP and OSI model. *Tech Differences.* [Online] 25. Březen 2016. https://techdifferences.com/difference-between-tcp-ip-and-osi-model.html.

2. **Mendelova Univerzita Brno.** Referenční model ISO/OSI. *Referenční model ISO/OSI.* [Online] https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=10010.

3. **Beal, Vangie.** The 7 layers of the OSI model. *WeboPedia.* [Online] 23. Duben 2019. https://web.archive.org/web/20200305212605/https://www.webopedia.com/quick_ref/OSI_Layers.asp.

4. **Internet a Jeho Služby.** Internet a Jeho Služby. *Referenčn model ISO/OSI.* [Online] http://ijs.8u.cz/index.php/standardizace-v-pocitacovych-sitich/referencni-model-iso-osi.

5. **Shaw, Keith.** The Osi model explained: How to understand (and remeber) the 7 layer network model. *Network World.* [Online] 22. Říjen 2018. https://www.networkworld.com/article/3239677/the-osi-model-explained-how-to-understand-and-remember-the-7-layer-network-model.html.

6. **Study Tonight.** The OSI Model - Features, Principles and Layers. *Study Tonight.* [Online] 2020. https://www.studytonight.com/computer-networks/complete-osi-model.

7. **Java T point.** OSI Model. *Java T point.* [Online] 2018. https://www.javatpoint.com/osi-model.

8. **Cloudflare.** What is the OSI model? *Cloudflare.* [Online] 2020. https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/.

9. **Geeks for Geeks.** Layers of OSI Model. *Geeks for Geeks.* [Online] 2020. https://www.geeksforgeeks.org/layers-of-osi-model/.

10. **Mojidra, Nilesh.** Stateful vs. Stateless Firewalls. *Cybrary.* [Online] 29. Červenec 2016. https://www.cybrary.it/blog/0p3n/stateful-vs-stateless-firewalls/.

11. **Faculty of Electrical Engineering - Czech Technical University in Prague.** Komponenty síťového bezpečnostního systému. *Firewall.* [Online] http://techpedia.fel.cvut.cz/html/frame.php?oid=76&pid=1018&finf=.

12. **Solarwinds msp.** Stateful vs. Stateless Firewall Differences. *Solarwinds msp.* [Online] 12. Červenec 2019. https://www.solarwindsmsp.com/blog/stateful-vs-stateless-firewall-differences.

13. **Pescatore, John a Young, Greg.** Defining the Next-Generation Firewall. [Online] 12. Říjen 2009. http://img1.custompublish.com/getfile.php/1434855.1861.sqqycbrdwq/Defining+the+Next-Generation+Firewall.pdf.

14. **Zhang, Ellen.** What is Data Loos Prevention (DLP)? A Definition of Data Loss Prevention. *DataInsider - Digital Guardian.* [Online] 27. Leden 2020. https://digitalguardian.com/blog/what-data-loss-prevention-dlp-definition-data-loss-prevention.

15. **Autocont.** Data Loss Prevention. *Autocont.* [Online] 2019. https://www.autocont.cz/aktuality/openspace/data-loss-prevention/jak-dlp-pomaha.

16. **Rouse, Margaret.** data loss prevention (DLP). *WhatIs.com.* [Online] Říjen 2014. https://whatis.techtarget.com/definition/data-loss-prevention-DLP.

17. **Cisco.** What is a VPN? - Virtual Private Network. *Cisco.* [Online] 2020. https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html.

18. **Skoler, Ben.** Remote-access VPN vs Site-to-site VPN. *VPN Mentor.* [Online] 26. 12 2019. https://www.vpnmentor.com/blog/remote-access-vpn-vs-site-to-site-vpn-full-guide/.

19. **Keogh, Frank.** What is Advanced Malware Protection (AMP)? *TEC Communications.* [Online] 2019. https://tec4it.com/what-is-advanced-malware-protection-amp/.

20. **Gartner, Inc.** *Gartner 2019.* [Obrázek] místo neznámé : Microsoft.com, Říjen 2019. https://www.microsoft.com/security/blog/wp-content/uploads/2019/10/Gartner-2019.png.

21. **Check Point.** Threat Extraction pro 100% bezpečné dokumenty v reálném čase. *Check Point.* [Online] 13. Březen 2015. https://www.dns.cz/sites/default/files/150313-check-point-tex-tz.pdf.

22. **DNS better way.** Check Point představuje technologii Threat Extraction. *DNS better way.* [Online] 2015. https://www.dns.cz/aktuality/check-point-predstavuje-technologii-threat-extraction-pro-100-bezpecne-dokumenty-v-realnem.

23. **Imperva.** DDoS Attacks. *Imperva.* [Online] 2019. https://www.imperva.com/learn/application-security/ddos-attacks/.

24. **Keary, Tim.** DoS vs DDoS Attacks: The Differences and How To Prevent Them. *Compari Tech.* [Online] 21. Listopad 2018. https://www.comparitech.com/net-admin/dos-vs-ddos-attacks-differences-prevention/.

25. **Broadcom.** Symantec Secure Web Gateway. *Broadcom.* [Online] 2017. https://docs.broadcom.com/docs/next-generation-secure-web-gateway-the-cornerstone-of-your-security-architecture-en.

26. **Duračinská, Zuzana.** Amplification útoky obecně. *csirt.cz.* [Online] 10. Duben 2015. https://csirt.cz/cs/kyberbezpecnost/pro-administratory/amplification-utoky-obecne/.

27. **Mukherjee, Biswanath, Heberlein, L. Todd a Levitt, Karl N.** *Network Intrusion Detection.* 1994.

28. **Singhal, Anoop.** *Intrusion Detection Systems.* 2007.

29. **CheckPoint.com.** What is IPS. *www.checkpoint.com.* [Online] 2020. https://www.checkpoint.com/definitions/what-is-ips/.

30. **Petters, Jeff.** IDS vs. IPS. *Varonis.* [Online] 23. 12 2018. https://www.varonis.com/blog/ids-vs-ips/.

31. **PC Magazine.** Definition of honeypot. *PC Mag.* [Online] https://www.pcmag.com/encyclopedia/term/44335/honeypot.

32. **Spitzner, Lance.** *Honeypots: Tracking Hackers.* místo neznámé : Addison-Wesley Professional, 2002. ISBN 0-321-10895-7.

33. **Talabis, Ryan.** *Honeypots 101: Brief History of Honeypots.* 2007.

34. **Techopedia.com.** Honeypot. *Techopedia.* [Online] 23. Březen 2012. https://www.techopedia.com/definition/10278/honeypot.

35. **Gibbens, Mathias a Rajendran, Harsha vardhan.** Honeypots. [Online] 22. Duben 2012. https://www2.cs.arizona.edu/~collberg/Teaching/466-566/2013/Resources/presentations/2012/topic12-final/report.pdf.

36. **Göbel, Jan Gerrit.** *Large-Scale Detection and Measurement of Malicious Content.* místo neznámé : Sudwestdeutscher Verlag Fur Hochschulschrifte, 2011. 9783838127200.

37. **Týma, Jiří.** Moderní honeypoty a honeynety. [Online] 2018. https://is.muni.cz/th/buzbs/moderni-honeypoty-honeynety.pdf.

38. **Niels, Provos a Thorsten, Holz.** *Virtual Honeypots.* 2007. 978-0-321-33632-3.

39. **Peter, Eric a Schiller, Todd.** A Practical Guide to Honeypots. [Online] 15. Duben 2008. https://www.cse.wustl.edu/~jain/cse571-09/ftp/honey.pdf.

40. **Enisa.** Proactive detection of security incidents II - Honeypots. [Online] 2012. https://www.enisa.europa.eu/publications/proactive-detection-of-security-incidents-II-honeypots.

41. **Osmík, Lukáš.** *Použití honeypotu ve školní síti.* 2011.

42. **Galetka, Josef.** Analýza síťových útoků pomocí honeypotů. [Online] 2010. https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=117452.

43. **The Honeynet Project.** Know your Enemy: Honeynets. *The Honeynet Project.* [Online] 2006. http://old.honeynet.org/papers/honeynet/.

44. **Anagnostakis, K G, a další.** Detecting Targeted Attacks Using Shadow Honeypots. [Online] 9. 10 2010. https://www.usenix.org/legacy/events/sec05/tech/full_papers/anagnostakis/anagnostakis.pdf.

45. **Akkaya, Deniz a Thalgott, Fabien.** Henypots in Network Security. *Linnaeus University.* [Online] 29. Červen 2010. http://www.diva-portal.org/smash/get/diva2:327476/fulltext01.pdf.

46. **Spitzner, Lance.** The Value of Honeypots, Part Two: Honeypot Solutions and Legal Issues. *Broadcom.* [Online] 23. Říjen 2001. https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=e12cd08d-413d-4a07-9b2a-d502ba248abd&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments.

47. **6c2e6e2e.** Honeypots with Modern Honey Network (MHN). *Medium.* [Online] 27. Únor 2018. https://medium.com/@6c2e6e2e/honeypots-with-modern-honey-network-mhn-6acd1a04d4a9.

48. **Trost, Jason.** Modern Honey Network. *Anomali.* [Online] 19. Červen 2014. https://www.anomali.com/blog/mhn-modern-honey-network.

49. **Tan, Emil.** Dionaea - A malware capturing honeypot. *Edgis Security.* [Online] 13. Únor 2014. http://www.edgis-security.org:80/honeypot/dionaea/.

50. **Pontén, Austin.** *Evaluation of Low-Interaction Honeypots on the University Network.* [Dokument] 2017.

51. **Bove, Davide.** *Using Honeypots to Detect and Analyze Attack Patterns on Cloud Infrastructures.* [Dokument] Erlangen : autor neznámý, 30. Říjen 2018.

52. **Mahmoud, Rasmi Vlad.** *Honeypots on AAU's Network.* 6. Červen 2019.

53. **Kelly, Gary a Gan, Diane.** *Analysis of Attacks Using a Honeypot.* Červen 2014.

54. **French, David.** How to Setup "Cowrie" - An SSH Honeypot. [Online] 1. Říjen 2018. https://medium.com/threatpunter/how-to-setup-cowrie-an-ssh-honeypot-535a68832e4c.

55. **McCann, William.** Cowrie Honeypot Analysis Results. *William McCann.* [Online] 6. Říjen 2017. https://wjmccann.github.io/blog/2017/10/06/Cowrie-Honeypot-Analysis.

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor: **Bc. Ondřej Líbal**

Studium: I1800140

Studijní program: N6209 Systémové inženýrství a informatika

Studijní obor: Informační management

Název diplomové práce: **Principy honeypotů a jejich využití pro zabezpečení síťového provozu**

Název diplomové práce AJ: Principles of honeypots and their use for network security

Cíl, metody, literatura, předpoklady:

Cílem práce je podrobně představit principy fungování honaypotů, jejich rozdělení a využití pro zabezpečení síťového provozu. V teoretické části autor představí principy fungování honeypotů, jejich typy a možnosti nasazení. V praktické části autor představí modelové nasazení vybraných typů honeneypotů, jejich konfiguraci a otestuje jejich efektivitu s důrazem na open source řešení.

1. Úvod
2. Co je to honeypot
3. Jak funguje honeypot
4. Rozdělení honeypotů
    1. Low-interaction
    2. High-interaction
    3. Pure
5. Typy honeypotů
6. Instalace a nastavení honeypotů
7. Testování honeypotu
8. Závěr

SPITZNER, Lance. *Honeypots: tracking hackers. Boston: Addison-Wesley, c2003. ISBN 9780321108951.*

MOHAMMED, Mohssen a Habib-ur REHMAN. Honeypots and routers: collecting internet attacks. Boca Raton, FL: CRC Press, [2016]. ISBN 9781498702195.

Garantující pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: Mgr. Josef Horálek, Ph.D.

Datum zadání závěrečné práce: 21.10.2014