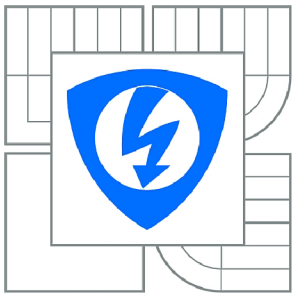
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

SYSTÉMY DETEKCE A PREVENCE PRŮNIKU

INTRUSION DETECTION AND PREVENTION SYSTEMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

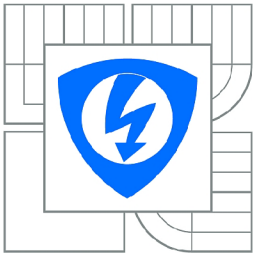
AUTOR PRÁCE — Bc. MICHAL ČERNÝ

AUTHOR

VEDOUCÍ PRÁCE — Ing. RADIM PUST

SUPERVISOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Michal Černý ***ID:*** 83083
Ročník: 2 ***Akademický rok:*** 2009/2010

NÁZEV TÉMATU:

Systémy detekce a prevence průniku

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je popsat principy činnosti systémů pro detekci a prevenci průniků do počítačové sítě. Dále navrhnout řešení pro detekci a prevenci s koncepcí host a network base systému. Přičemž koncepce network base by měla být založena na platformě Linux a host base na platformách Linux a Windows. Dále navržené řešení realizovat tj. popsat instalaci a nastavení všech užitých nástrojů. A následně realizované řešení ověřit.

DOPORUČENÁ LITERATURA:

[1] TOXEN, Bob. Bezpečnost v Linuxu : Prevence a odvracení napadení systému. [s.l.] : Computer Press, 2003. 876 s. ISBN 80-7226-716-7.
[2] KABELOVÁ, Alena, DOSTÁLEK, Libor. Velký průvodce protokoly TCP/IP a systémem DNS . [s.l.] : Computer Press, 2008. 488 s. ISBN 978-80-251-2236-5.

Termín zadání: 29.1.2010 ***Termín odevzdání:*** 26.5.2010

Vedoucí práce: Ing. Radim Pust

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Systémy detekce a prevence průniku mohou být realizovány jako samostatný hardware nebo nasazeny v podobě software na hostiteli. Primárním účelem těchto bezpečnostních prvků je odhalení nežádoucí aktivity, jako například narušení integrity souborů, neplatné pokusy při připojení ke vzdálené službě nebo získávání informací o vnitřní síti. Na událost reagují na základě akce, definované ve vnitřními pravidly. Mezi možná protiopatření lze zařadit vydání výstrahy nebo blokování komunikace.

V rámci diplomové práce jsou popsány základní principy systému detekce i prevence průniku. Zmíněny jsou různé typy analýz zachycených dat, postupy při tvorbě vlastních pravidel nebo formáty výstrah. Zvažovány jsou také varianty jejich umístění, včetně výhod pro vybrané situace. Popsána je instalace a nastavení jednotlivých prvků realizované sítě i bezpečnostních systémů. K ověření funkčnosti a míry poskytované ochrany bylo provedeno několik vybraných typů útoků.

KLÍČOVÁ SLOVA

HIDS IDS IPS NIPS Snort inline OSSEC

ABSTRACT

The detection and intrusion prevention systems could be realized as independent hardware or set in the software form on to the host. The primary purpose of these protective elements is the undesirable activity detection such as integrity intrusion of the files, invalid attempts while connecting to the remote service or acquisition of the local network data. The systems react to the event on the basis of the action that is defined by internal rules. We can include the caution sending or communication blocking among possible counteractions.

The base principals of the detection and intrusion prevention systems are described in the dissertation. Various types of captured data analyses and processes of the inhere rules creation and further more caution formats are mentioned in the dissertation. There are also considered the alternatives of their location including advantages of selected situations. There is described the installation and setting up of particular elements of the realized network and security systems. In order to the verification of functionality and factor of the protection providing there was realized several selected types of attacks.

KEYWORDS

HIDS IDS IPS NIPS Snort inline OSSEC

ČERNÝ, M. *Systémy detekce a prevence pruniku*: diplomová práce. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010. 85 s. Vedoucí práce Ing. Radim Pust

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Systémy detekce a prevence pruniku“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

BRNO ....

(podpis autora)

Děkuji vedoucímu diplomové práce Ing. Radimu Pustovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

Bezpečnost je velice často diskutovaným tématem, oblast informačních technologií nevyjímaje. Obzvláště dnes, kdy je většina počítačů propojena sítí. Nejčastěji jí bývá Internet. Společně s jeho příchodem došlo ke značnému rozšíření komunikačních možností. Množství provozovaných služeb je celá řada, jako například elektronická pošta, sdílení různorodého typu dat, internetová telefonie, sociální sítě a mnoho dalších. S rostoucím množstvím připojených subjektů a jistým množstvím anonymity však také výrazně vzrostlo riziko napadení subjektu a kompromitace uložených dat nebo poskytovaných služeb. Z tohoto důvodu by měla být bezpečnost jednotlivých objektů vnímána jako velice důležitá. Protože se však jedná o stav, nikoliv o vlastnost, jak se by se mohlo na první pohled zdát, je nutné pravidelné provádění příslušných kroků k zachování systémů bezpečným.

Velkým problémem může být útok vedený na slabiny provozovaných protokolů. Obzvláště na ty, kterým je povolen průchod firewallem. Situace, kdy dojde k napadení jedné entity v dané síti, může být velice nebezpečnou. Kompromitovaný server může sloužit jako „odrazový můstek“ k útokům vedeným na další servery v autonomním systému. Pokud se podaří na server nainstalovat škodlivý software typu „rootkit“ nebo „backdoor“, cracker má odkudkoliv zajištěný neustálý přístup k danému stroji.[13]

Protože žádný systém nemůžeme považovat za naprosto bezpečný, je vhodné provádění monitorování nestandardní chování či nežádoucí síťové aktivity. K tomuto účelu byly navrženy nástroje spadající do kategorie IDS a IPS.

1 SYSTÉMY DETEKCE A PREVENCE

Systémy detekce a prevence představují softwarové nebo hardwarové řešení, pomocí něhož je možné detekovat a následně blokovat nežádoucí činnost, jako tomu zpravidla bývá u detekčních systémů. Obecně je však vhodnějším řešením blokování nežádoucí aktivity neprodleně, což je významným rysem systémů prevence průniku. Díky těmto vlastnostem se čím dále častěji stávají součástí obranné strategie především v sektoru středních a velkých podniků.

Stále se však tyto systémy nacházejí ve svých počátcích. Jejich vývoj a rozšíření lze tedy očekávat obzvláště dnes, kdy je na bezpečnost nahlíženo s vyšším důrazem. Systémy detekce mají však také svá úskalí, jakými mohou být falešná hlášení útoků, problémy se stabilitou u vysokorychlostních systémů nebo s detekcí neznámých hrozeb. Většina problému je ale zapříčiněna nehodnou implementací a nepochopením možností dané technologie.

1.1 Architektura

Architekturu systémů detekce i prevence průniku je možné rozdělit do tří základních skupin. Hlavním kritériem pro řazení jsou především závislosti jednotlivých prvků. Každý bývá zpravidla určen pro vykonávání konkrétní činnosti.

1.1.1 Jednovrstvá

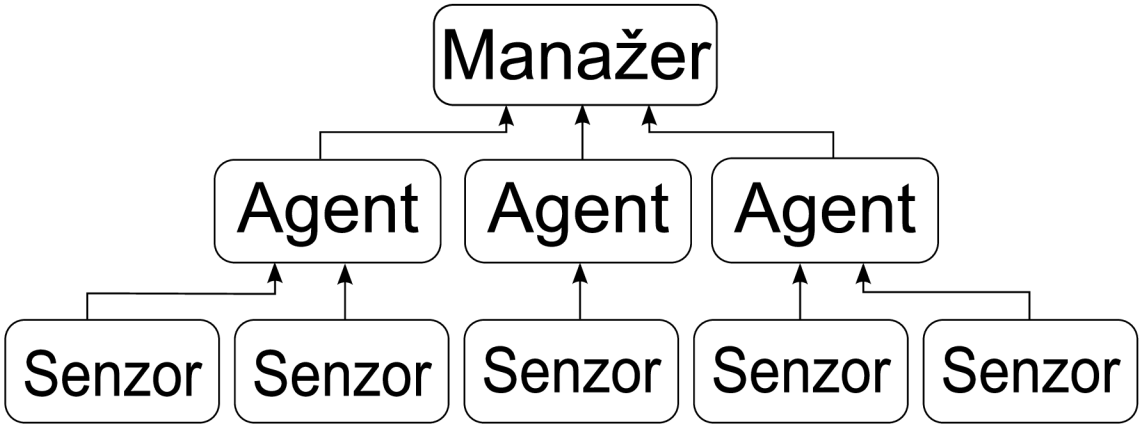
Jednovrstvá architektura je nejjednodušším možným případem, kdy je IDS nebo IPS tvořen pouze jednou komponentou, která obstarává všechny potřebné funkce. Získaná data tedy nejsou přeposílána na výstup k zpracování nadřazenou entitou. Operace, jako například zachytávání dat, analýza nebo informování o zachycené hrozbě, jsou tedy prováděny v rámci jedné komponenty bez spolupráce s ostatními prvky, jak je naznačeno na obrázku 1.1. Příkladem jednovrstvé architektury může být nástroj procházející záznamy o činnosti služeb a programů, které porovnává s databází.

Obr. 1.1: Struktura jednovrstvé architektury.

Mezi výhody této architektury bezesporu patří jednoduchost a nízká cena daného řešení. Jistým přínosem může také být nezávislost na dalších součástech, které se mohou stát terčem kompromitace. Nevýhodná však může být situace, kdy umístíme více takovýchto systémů, které dokáží pracovat pouze samostatně bez vědomí o dalších systémech. Není zde možnost komunikace jednotlivých entit a tedy jejich efektivnější a sofistikovanější využití.

1.1.2 Vícevrstvá

Odlišností vícevrstvých systémů je především existence vertikální komunikace mezi entitami. Ty jsou uspořádaných hierarchicky do stromové struktury. U dnešních IDS a IPS to bývá zpravidla ze tří vrstev, jak je naznačeno na obrázku 1.2.



Obr. 1.2: Struktura vícevrstvé architektury.

Základem systémů jsou senzory, které získané údaje o síťovém provozu předávají dále ke zpracování agentům. V rámci jednoho systém může být umístěno několik a každý může provádět analýzu jiného protokolu či služby. Nejvyšším prvkem je manažer, který přebírá hlášení od analyzátorů a může případně provádět definovaná opatření. Slouží také jako přístupový bod pří obsluze a správě systému.

Výhody vícevrstvé architektury spočívají ve vyšší efektivitě a hloubce analýzy. Také dokáže poskytnout mnohem kompletnější obraz stavu bezpečnosti sledované sítě, nežli je tomu u jiných architektur. Na druhou stranu tato šířka pohledu s sebou přináší také vyšší náklady.

Senzory

Hlavní funkcí těchto entit je sběr dat, které jsou následně předávány nadřazené komponentě. Data je možné získávat sledováním provozu na síti nebo také z dalších zdrojů, jakými mohou být výpisy z programů, služeb nebo zařízení, jako například firewallů nebo TCP Wrappers. Každý senzor je konfigurován a přizpůsoben k běhu na specifickém operačním prostředí, ve kterém je umístěn.

Agent

Obvykle bývají komponenty agentů jednostranně specializovány. Příkladem může být situace, kdy jeden agent slouží pouze ke sledování TCP provozu, zatímco další je určen pouze ke sledování FTP provozu. Výhodné může být také použití softwaru třetích stran určených k monitorování síťové aktivity nebo nástrojů zobrazujících komunikační cestu. Pokud je agentem obsah spojení vyhodnocen jako útok, provádí informování manažera.

Manažer

Manažer je nejvýše postavenou entitou v této hierarchii. Hlavní funkcí je koordinace agentů, přijímání zpráv a v neposlední řadě zajišťování činnosti celého systému. Na přijeté zprávy od agentů může reagovat vykonáním některé z následujících funkcí:

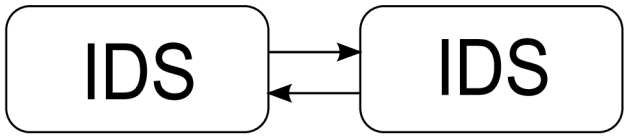
- Seskupení a zobrazení varování v konzoli.
- Upozornění na pager nebo mobilní telefon.
- Uložení informací o útoku do databáze.
- Získávání hlubších informací o incidentu.
- Zaslání informací k hostiteli, který zastaví komunikaci.
- Zaslání příkazu firewallu nebo routeru ke změně pravidel.
- Poskytuje uživatelské rozhraní k manažerské komponentě.

Výhody

Centralizovaný systém sběru dat umožňuje jednodušší analýzu záznamů, protože jsou dostupná na jednom místě. Další výhoda tohoto uspořádání vyniká v případě, kdy se útočníkovi podaří modifikovat nebo smazat nějaký záznam. Originál bývá uložen na centrálním serveru u manažerské komponenty. Pomocí ní je možné měnit pravidla bezpečnostní politiky IDS i IPS, odstraňovat záznamy, po té, co byli na centrálním serveru archivovány nebo provádět ověřování senzorů, agentů nebo vzdálených systémů.

1.1.3 Peer–to–peer

V případě vícevrstvé architektury obecně dochází k zachycení informace, následnému zpracování a odeslání na výstup směrem k nadřazené komponentě. Peer–to–peer architektura takto uspořádána není. Výměna informací probíhá horizontálně mezi jednotlivými entitami, ležícími na stejné úrovni a vykonávajícími stejnou nebo obdobnou funkci, jak naznačuje obrázek Obr.1.3. Tato architektura je často využívána ve spolupráci s firewally. Pokud systém získá informace o nových událostech, které se na síti vyskytují, provede jejich přeposlání a změnu ve stávajících pravidlech nebo reaguje přidáním nového pravidla. Stejné chování je očekáváno i od ostatních prvků. Není tu tedy žádný nadřazený řídící či nadřazený prvek.



Obr. 1.3: Struktura architektury peer–to–peer.

Hlavní výhodou peer-to–peer architektury je jednoduchost. Každá entita může být umístěna ve skupině, které přináší prospěch sběrem informací. Nevýhodou je naopak nedostatek sofistikované funkčnosti kvůli absenci specializovaných součástí. Avšak funkčnost dosahuje vyšší úrovně, nežli je tomu u jednovrstvé architektury.

1.2 Analýza

Analýza prováděná systémy detekce a prevence narušení je založena na rozboru dat shromážděných sondami za účelem rozpoznání nestandardního obsahu přenášených informací. Analýzou, probíhající v reálném čase, rozumíme rozbor dat v časovém úseku definovaném dobou jejich přijetí a odeslání na výstup. Z tohoto je patrné, že je žádoucí, aby analýza probíhala co nejrychleji a nedocházelo tím k nežádoucímu zpoždění dat. Proces, kterým sledovaná data procházejí lze rozdělit do čtyř základních částí:

- Předzpracování
- Analýza
- Odezva
- Doladění

Předzpracování

Je klíčovou funkcí, zahrnující především sběr dat z IDS a IPS senzorů. Dále dochází k jejich zařazení podle struktury příchozích dat.

Analýza

Po dokončení předzpracování následuje analýza. Získaná data jsou porovnávána se znalostní databází. Pokud je vyhodnocen pokus o vniknutí, bývá spojení zrušeno a pakety zahozeny. Záleží však na konkrétním schématu.

Odezva

V tomto bodě se od sebe systémy IDS a IPS liší. Kdy IDS mívá zpravidla omezené schopnosti prevence. Dochází pouze k informování vhodné entity probíhajícím útoku nebo nestandardním chování. Tato informace je uložena a teprve poté může být inicializována obslužná rutina odezvy. Naopak je tomu u IPS systémů, kde je senzor umístěn in–line, což mu společně s prováděním analýzy v reálném čase umožňuje nechtěnou komunikaci blokovat neprodleně, případně provézt jinak definovanou akci. Přidání pravidla může být prováděno automaticky nebo ručně po té, co je vyhodnocena daná situace.

Doladění

Tato konečná fáze není sice nutná, ale neměla by chybět u profesionálních detekčních systémů. Složí k jemnému doladění pravidel, kdy hlavní výhodou je snížení úrovně falešných poplachů a zvýšení preciznost bezpečnostního nástroje. Pro efektivní běh IDS či IPS systému je tedy doladění velice důležitou fází, kterou je vhodné provézt pro každé konkrétní prostředí.

1.2.1 Detekce signatur

Systémy využívají metody detekce vzoru, provádějí porovnání datového provozu na síti s databází známých typů útoků. Tato metodou je řazena mezi přesné a jednoduché. Má však svá úskalí, jako například špatná přizpůsobivost při různých modifikacích známých útoků. Pokud jednotlivé mutace nejsou zaneseny v databázi, systém hrozbu nezaznamená. Přidání modifikace nebývá příliš často automatizováno. Mezi další nevýhody patří omezení porovnávání sítového provozu, omezeného na konkrétní paket. Nemá tedy možnost komplexnějšího pohledu.

1.2.2 Stavová detekce signatur

Je rozšířením předchozí metody detekce signatur o možnost kontroly spojení jako celku. To je výhodné v případě, kdy je signatura sítového útoku rozdělena do více paketů.

1.2.3 Detekce založená na pravidlech

Při této detekční metodě je využito databáze pravidel definovaných pro odhalení útoku. Pomocí několika pravidel je možné pokrýt velké množství typů útoků, i odhalení dosud neznámých. Nevýhodou může být nutnost aktualizace databáze pravidel, kterou však není nezbytné provádět tak často. Databáze má dále také menší velikost, nežli je tomu u databáze signatur.

1.2.4 Detekce na základě profilu

Detekce anomálií není snadné jednoznačně definovat. Lze ji přibližně popsat jako jistý detekční profil, do kterého je uloženo standardní chování systému. Klíčovou je definice aktivit nestandardního chování. Tedy činností, které jsou zakázané a které povolené. Pomocí vhodného nástroje možné snazší vytvoření detekčního profilu i následných statistik. Příkladem může být situace, kdy není známo, aby daný protokol komunikoval mimo danou síťovou oblast nebo překročil jistou kapacitu linky.

Schéma analýzy anomálií lze rozdělit do tří následujících kategorií:

- Změny chování
- Typ provozu
- Protokoly

Analýza změny chování vyhledává odchylky od běžného stavu, jako například souvislosti mezi pakety nebo jejich obsah. U typu provozu je prováděno vyhledávání specifických vzorků v síťovém provozu. Protokolová analýza sleduje porušení či nesprávné užití protokolů, oproti jejich stavu definovaného normou. Tato analýza má také výhodu v možnosti odhalení dosud nezveřejněných typů útoků. Nejsou tedy ještě zaneseny v databázi signatur.

1.2.5 Monitorování objektu

Monitorování slouží k informování o změnách, jako jsou nahrazení nebo modifikace. Realizována je porovnáváním otisků každého sledovaného objektu v různém čase. Otisk je pořizován pomocí kryptografického algoritmu. Na základě této změny poté

sledující systém může reagovat na změny objektu například zasláním varování nebo odepření přístupu.

1.2.6 Skrytý monitoring

Tento typ analýzy je prováděn pomocí korelace zachytávaných informací. To umožňuje detekci pomalých útoků, vyznačují se dlouhou dobou mezi jednotlivými kroky. Použití jiného typu analýzy není na odhalení takovéhoto útoku účinné. Analýza je založena na sběru dat od několika senzorů, která jsou následně porovnávána na existenci vzájemných vazeb.

1.2.7 Hybridní analýza

Tento typ analýzy je kombinací dvou nebo více výše uvedených metod. Hlavní výhodou poskytující takováto kombinace je komplexnost a robustnost takovéhoto řešení.

2 IDS

Systém detekce průniku je možné definovat jako soubor nástrojů, prostředků a metod, kterými je možné identifikovat, klasifikovat a zaznamenávat nežádoucí síťovou aktivitu. Detekční systém bývá typicky pouze částí celkové ochranné strategie.[20] Není tedy vhodné jeho samostatné nasazení pro dosažení efektivní ochrany. Systémy IDS lze rozdělit do několika základních skupin:

- Umístěné na síti – V tomto systému provádějí monitoring na úzkém místě sítě, kterým může být například hranici sítě .
- Systémové – PIDS a APIDS slouží k monitorování neplatné komunikace nebo neodpovídající protokolové sémantice. Příkladem může být podvržení příkazu SQL protokolu vedoucí k poškození databáze nebo viry šířené elektronickou poštou.
- Umístěné na hostiteli – Senzor obvykle provádí monitoring volených aktivit hostitele, na kterém je nainstalován. Nejčastěji kontroluje pokusy o modifikaci MBR, výpisy programů nebo přistupování k souborům.
- Hybridní – Je kombinací dvou nebo více typů. Například IDS umístěný na hostiteli přijímající informace o síti je schopen poskytnout mnohem komplexnější pohled na celou síť.

2.1 Umístěný na síti

NIDS je nezávislou platformou, která slouží k odhalení průniku pomocí monitorování síťového provozu. Zároveň je však možný dohled nad síťovým chováním velkého množství objektů. Systém bývá do sítě připojen rozbočovačem nebo přepínačem nastaveným na zrcadlení portů. Další možností je pasivní připojení pomocí síťové odbočky. Mezi zástupce IDS umístěných na síti patří například software Snort.

2.2 Monitorující protokol

PIDS bývá nejčastěji realizovaný systémem, umístěným na straně serveru. Slouží k monitorování a analýze komunikačního protokolu mezi připojovanými zařízeními a serverem. Pro webový server jsou typickými objekty monitorování protokoly HTTPS a HTTP. Při nasazení HTTPS, je nutné sledování komunikace dříve, než je odeslána prezentační vrstvě, kde dochází k jejímu šifrování.

2.3 Monitorující specifický protokol

Umístění APIDS je obdobné, jako u PIDS. Rozdíl spočívá v monitorování protokolu, který je však specifický pro konkrétní aplikaci. Příkladem může být webový server s databází, kdy je prováděno sledování sémantiky jednotlivých operací protokolu SQL.

2.4 Umístěný na hostiteli

HIDS provádí identifikaci vetření na základě analýzy systémových volání, záznamů z programů, modifikací souborového systému i ostatních aktivit a stavů hostitele. Zástupcem HIDS může být například software OSSEC nebo Snort.[7]

2.5 IDS senzory

Velice důležitou a nedílnou součástí systémů detekce průniku jsou senzory. Zpravidla leží na nejnižší úrovni v rámci architektury daného systému, jak je naznačeno na obrázku 1.2. Nedisponují tedy většinou žádnými sofistikovanějšími funkcemi, protože jsou navrženy především k získávání potřebných informací.[20] Oproti IPS jsou IDS senzory více flexibilní. Poskytují však menší možnosti. Nicméně v některých případech je možné jejich použití jako náhrada IPS senzorů.[2]

2.5.1 Umístění

IDS senzory lze rozčlenit do dvou základních skupin podle jejich umístění:

- Na síti
- Na hostiteli

Na síti

Umístění IDS senzorů je vhodné na přepínaných páteřních sítích, kde dochází ke kontrole velkého množství dat. Možné je nasazení několika málo senzorů, případně je možné užití pouze jediného.[2] Tímto krokem je možná úspora finančních prostředků, která však nedílně souvisí s nižší úrovní ochrany.

Na hostiteli

Umístění IDS senzorů je vhodné na hostiteli, kde je možné sledování mnoha aspektů, týkajících se chování hostitelského systému, síťové komunikace v místě, kde je možné provádět její kontrolu. Například není v daném místě ještě šifrována.

2.5.2 Sběr dat

U senzorů umístěných na síti s přepínáním paktů se obvykle používají dvě hlavní techniky sběru dat:

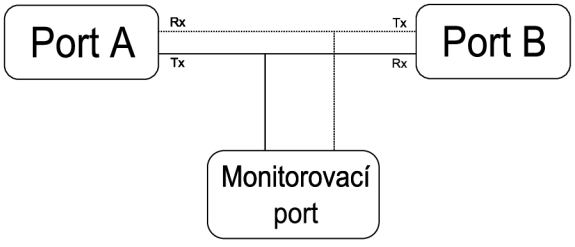
- Zrcadlení portů
- Síťovou odbočkou

Zrcadlení portů

Při zrcadlení portů, nazývané také jako „spanning“, dochází k předávání přicházejících dat ze vstupního na výstupní port, ale zároveň také ke kopírování na další port, který data odesílá dále k analýze. Realizováno bývá například rozbočovačem nebo konfigurovatelným přepínačem.

Síťovou odbočkou

Síťová odbočka je hardwarové zařízení, které umožňuje pasivní sledování síťové komunikace v počítačových sítích. Nevyžaduje tedy žádné aktivní zařízení, jako je tomu u techniky zrcadlení portů. Nasazení je vhodné při sledování síťového provozu mezi zvolenými dvěma body. Zařízení bývá realizováno pomocí dvou vstupně – výstupních portů, mezi které je zařazen třetí, sloužící k monitorování provozu, jak naznačuje obrázek 2.1.



Obr. 2.1: Schéma pasivní síťové odbočky.

Obě zmíněné metody sběru dat mají své výhody i jistá omezení, která je nutné vzít v úvahu při jejich nasazení.

2.5.3 Omezení IDS

- Šum – Šum může omezit identifikační efektivitu systému. Špatné pakety generované například neodladěným software nebo chybnými DNS údaji mohou způsobit falešný poplach vysoké priority.
- Málo časté útoky – Není obvyklé, aby počet reálných útoků byl nižší, než počet falešných poplachů. Odhalení reálného útoku je poté obtížné, protože může být systémem ignorován.
- Aktualizace Signatur – Mnoho útoků je vedeno proti slabinám specifické verze softwaru. V následující verzi ale mohou být tato slabá místa ošetřena. Může avšak nastat situace, kdy existují nové či stále neopravené slabiny. Z důvodu snížení rizika hrozeb, především u nových typů útoků, je vhodná pravidelná aktualizace databáze signatur.

3 IPS

Nedostatky spojené s IDS systémy, především nemožnost okamžité blokace síťové komunikace, vedly k vývoji nových produktů, označovaných jako IPS. Jsou to vlastně proaktivní systémy, navržené k detekci nežádoucí aktivity v běžné síťové komunikaci s možností zásahu proti ní. Umožňují provádění totožných analýz jako IDS, ale díky umístění senzorů přímo mezi síťovými zařízeními, kdy datový tok prochází přímo senzorem je možné rozhodnout, zda bude komunikace povolena či zakázána. Což je hlavním a velice důležitým rozdílem oproti IDS systémům, kde IDS není schopen komunikace neprodleně blokovat.

Některé dnešní firewally, routery nebo multimediální brány mají technologii prevenci průniku implementovánu. Jsou však implementovány ve formě skromného základu opravdového systému prevence průniku.[14] IPS lze je rozdělit do dvou hlavních kategorií:

- IPS umístěný na hostiteli.
- IPS umístěný na síti.

3.1 Umístěný na hostiteli

Podobně jako u systémů IDS, také IPS umístěný na hostiteli má své senzory a agenty instalované přímo na daném hostiteli.Je tedy těsně svázán s jádrem operačního systému, kde je snadný přístup ke službám, systémovým voláním jádru nebo API. Toto umístění je značně výhodné nejen při předcházení útoku ale také při generování zpráv o události.

Umožněno je snadné monitorování chování konkrétní aplikace včetně její komunikace s okolím. V dalším případě může ochránit aplikaci samotnou od běžně prováděných útoků, které však aktuálně nejsou součástí databáze signatur.

Těsné svázání s hostitelským operačním systémem s sebou přináší také jednu potencionální nevýhodu. Vyloučit totiž nelze případ, kdy aktualizace operačního systému způsobí vzájemnou nekompatibilitu s IPS systémem, což v krajním případě může způsobit omezení či úplný výpadek jeho funkčnosti. Naopak operačním systém přijímá od IPS žádosti. Aby bylo umožněno správné fungování obou systémů, je nutné splnění základních předpokladů IPS systému:

- Musí být spolehlivý.
- Nesmí negativně ovlivňovat výkon.

- Nesmí blokovat legitimní komunikaci

Jakýkoliv HIPS nesplňující tyto základní požadavky by nikdy neměl být instalován ani provozován bez ohledu na to, jak efektivně by dokázal zabránit útokům.[14]

3.2 Umístěný na síti

IPS umístěný na síti je spojením rysů standardního IDS systému, firewallu, a IPS systému. Často bývá také označován jako in–line IDS nebo IDS pracující jako výchozí brána (GIDS). NIPS vykazují jistou podobnost s nadcházející generací firewallů disponujících hlubší kontrolou. Tato zařízení jsou však zatím v počátcích svého vývoje.

NIPS disponuje také minimálně dvěmi síťovými rozhraními, obdobně jako je tomu typicky u síťového firewallu. Příchozí pakety jsou kontrolovány, zda nesená data nepředstavují nějakou hrozbu. Při zjištění škodlivé komunikace provede systém záznam o nalezení hrozby. Dále označí přenášený paket jako závadný a dojde k vyřazení z komunikace nebo ukončení celého spojení. Naopak pakety, které prošly v pořádku kontrolou jsou předány výstupnímu rozhraní a odeslány k zamýšlenému cíli. Užitečný postranní efekt některých NIPS produktů mimo detekce příchozích útoků také odhalování zneužití slabin v komunikačních protokolech, jako například u TCP/IP. Příkladem může být přijetí některých paketů mimo pořadí nebo s přesahující velikostí fragmentu pro protokol IP. Pokud se jedná o podvržené nebo poškozené pakety, jsou zahozeny ihned. U zbylých je provedeno přeskupení do správného pořadí a jejich následné odeslání příjemci.[14]

3.3 Požadavky

Pro správnou funkci celé sítě s nasazeným IPS prvkem je dobré dodržet následujících požadavky. V opačném případě může být ohrožena její spolehlivost, bezpečnost nebo dostupnost služeb.[13]

- In–line umístění senzoru – Toto umístění senzoru umožňuje zahazování škodlivých paketů a okamžitou blokaci spojení.

- Spolehlivost a dostupnost – Pokud zařízení umístěné in–line selže, může to mít za následek výpadek důležité trasy spojený s odepřením služeb. Proto je prioritní minimalizování takovýchto výpadků. Důležitá je také minimalizace doby, po kterou je prvek odpojen z důvodu aktualizace databáze signatur nebo informací o protokolech. Zařízení musí být schopné aktualizaci přijmout a začít používat bez nutnosti restartu. Při umístění prvku in–line se prostoj

a nefunkčnost linky způsobená jeho restartem negativně projeví v rámci části nebo celé sítě.

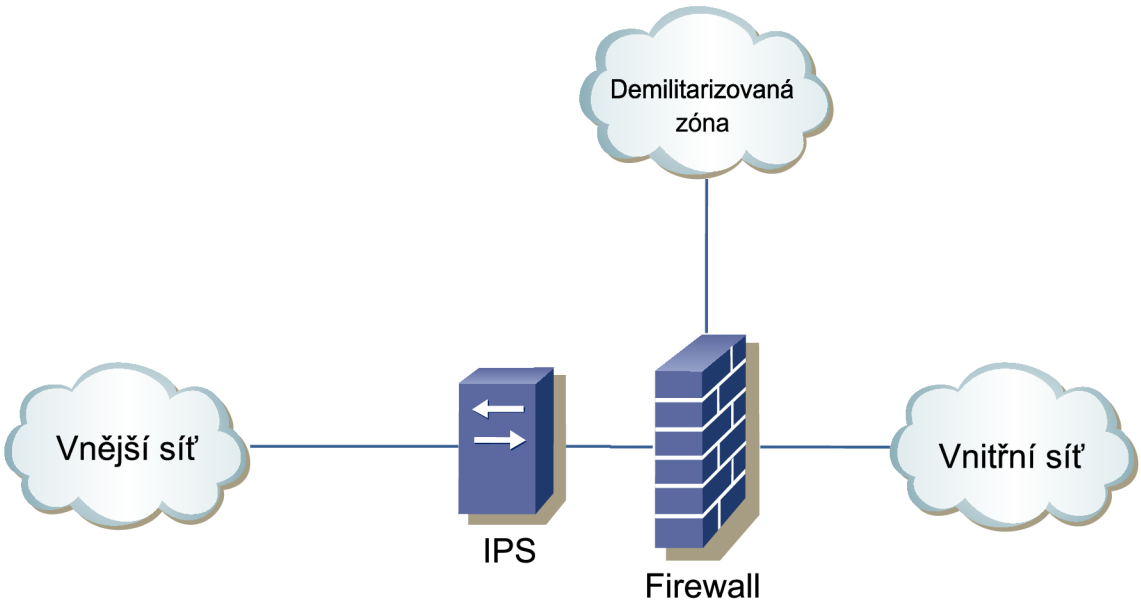
- Nízké zpoždění – Výkonnost zařízení umístěno in–line má významný dopad na propustnost a výkon celé sítě. Pakety by měli být zpracovány tak rychle, jak je to jenom možné. Typické zpoždění způsobené tímto prvkem obvykle dosahuje zpoždění zařízení, pracující na druhé nebo třetí vrstvě síťového modelu. Nemělo by však přesahovat zpoždění vznikající na prvcích čtvrté vrstvy, jakými jsou například firewally.

- Vysoká rychlost – Zpracování paketů je prováděno v reálném čase a tedy nutné disponovat dostatkem výpočetních prostředků k provádění potřebných operací. Zařízení by měla být navrhována s výkonovou rezervou, aby s rostoucím počtem kontrolované komunikace nedocházelo ke zvyšování zpoždění, způsobené bezpečnostním prvkem. V ideálním případě by velikost databáze signatur ani množství prováděných operací neměla způsobovat vyšší zatížení prvku.

- Identifikační přesnost – Bez pochyb vysoký vliv na kvalitu detekce má databáze signatur. Na základě vyhodnocení útoku dochází k blokaci spojení a odepření služby. Je proto vhodné požívat takové zařízení, kterému můžeme důvěřovat, že blokuje pouze nežádoucí komunikaci. Neprovádí tedy odepření žádoucí komunikace na základě chybného vyhodnocení. Nové signatury by měli být pravidelně dostupné a jejich přidání ke stávajícím pravidlům by mělo být rychlé, nejlépe prováděné u všech senzorů v síti najednou, pomocí centrální konzole.

3.4 Umístění senzorů

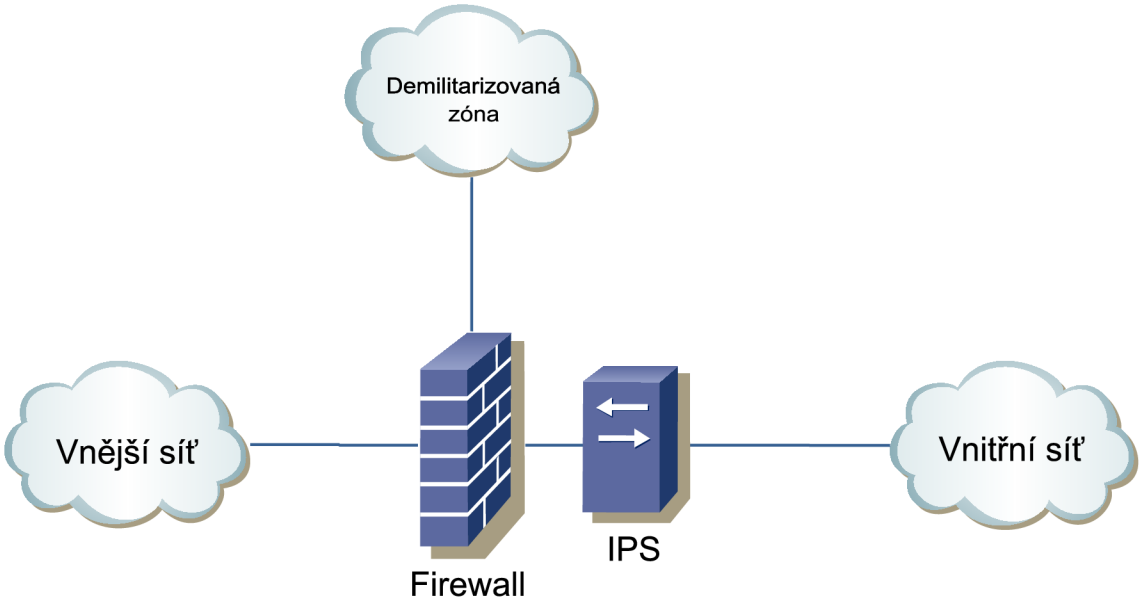
Umístění senzorů systémů IPS je rozhodující pro jejich správnou funkci. Zařazovány bývají především mezi komponenty infrastruktur různých sítí, kde se nalézají takzvaná úzká místa. Pro správnou funkci senzoru je vyžadováno, aby senzor byl umístěn in–line a komunikace tedy procházela skrze něho. Příkladem takového úzkého místa může být Internetová brána. V takovém případě je možné umístění senzoru před bránu, jak naznačuje obrázek 3.1, kdy je možné pomocí jednoho senzoru chránit celou vnitřní síť včetně demilitarizovaných zón, umístěných za firewallem. Hlavní nevýhodou tohoto řešení je však jeho obtížné a zdlouhavé nastavování. Provádí ale také kontrolu provozu, který nemusí být do vnitřní sítě propuštěn pravidly firewallu. V takovém případě dochází ke zbytečnému zabrání systémových prostředků a je také vhodné zakázání generování výstrah vzniklých na základě této komunikace. Současně se přepokládá, že není třeba informovat o každém útoku, směřujícím dovnitř. Takováto bezpečnostní politika je uplatňována nejčastěji, nikoliv však vždy. V některých prostředích jsou naopak tato varování vyžadována.



Obr. 3.1: IPS umístěný před firewallem.

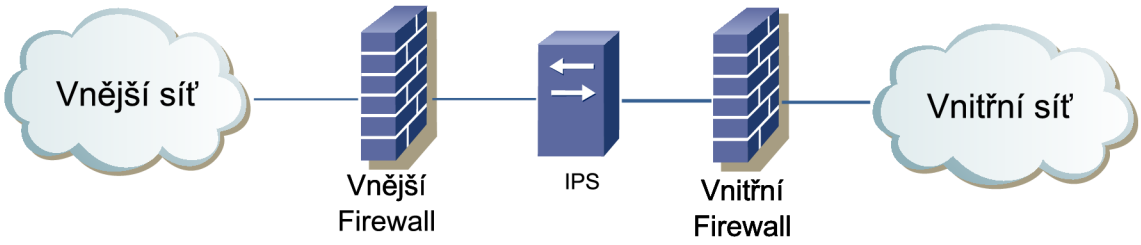
Obdobným případem je umístěním senzoru na stranu vnitřní sítě nebo přímo za firewall, jak naznačuje obrázek 3.2. Zde není třeba kontrolovat již tak velké množství dat, protože je zde již nežádoucí komunikace odfiltrována firewallem, což zvyšuje efektivitu IPS systému. Jistým kompromisem je v tomto případě počet potřebných senzorů ke zprostředkování stejné úrovně zabezpečení, jako je tomu v předchozím

případě. I zde platí obecné pravidlo, čím více senzorů, tím vyšší cena. Ta roste v obou scénářích o to radikálněji, jedná–li se o síť s vysokou dostupností, kdy je nutné užití redundantních spojů. Umístění ochranných prvků také na tyto spoje zpravidla bývá individuální pro každé řešení.



Obr. 3.2: IPS umístěný za firewallem.

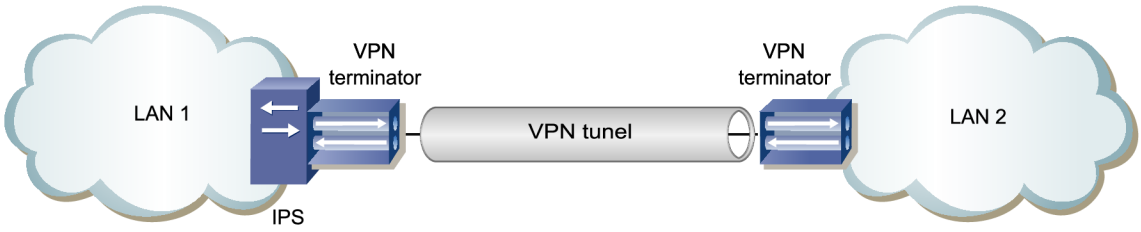
Mezi možné varianty připojení vnitřní sítě k vnější patří také umístění ochranného prvku mezi dva firewally, jak zachycuje obrázek 3.3. První provádí hrubé odfiltrování nežádoucí komunikace a případných útoků na bezpečnostní systém. Propuštěná spojení jsou dále kontrolována IPS prvkem. Pokud je komunikace vyhodnocena jako nezávadná, je přeposílána druhému firewallu a dále pokračuje do vnitřní či vnější sítě.



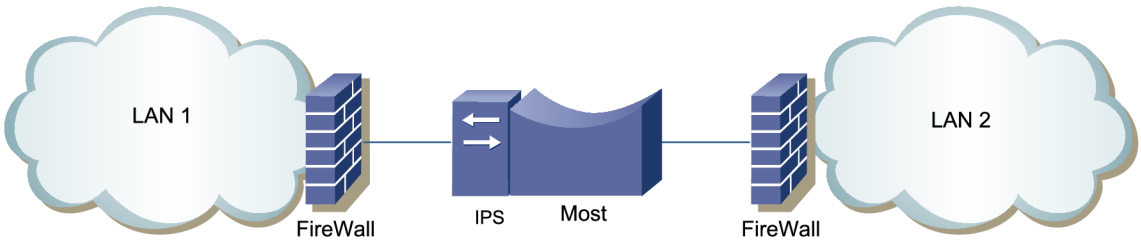
Obr. 3.3: IPS umístěný mezi dvěma firewally.

Předcházející příklady zachycovaly charakteristickou variantu, jakou je hranice Internetu, která však není jedinou existující. Mnoho organizací totiž používá extranet

pro spojení se svými pobočkami či obchodními partnery. Realizováno bývá pomocí VPN tunelu, jako na obrázku 3.4 nebo propojením místních sítí pomocí mostu ukončeného například na firewallech. Tento případ je zachycen na obrázku 3.5. Důvodem umístění senzorů za takovýmto VPN koncentrátorem či firewallem je především ochrana v rámci jedné sítě od ostatních. V případě VPN, musí být komunikace kontrolována před odesláním, tedy v místě, kdy není šifrována.



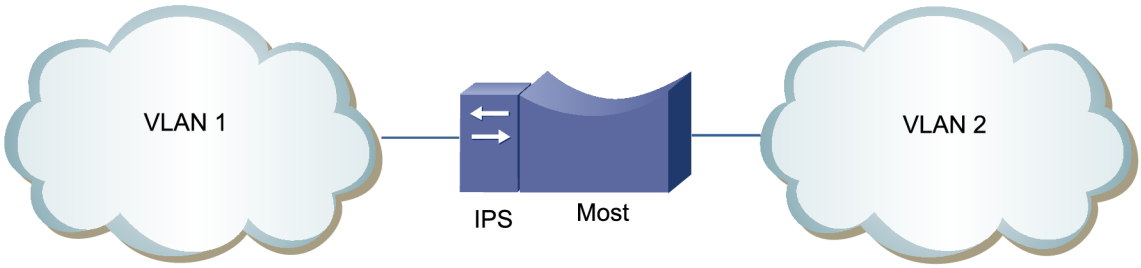
Obr. 3.4: Propojení dvou LAN pomocí VPN tunelu.



Obr. 3.5: Propojení dvou LAN pomocí mostu.

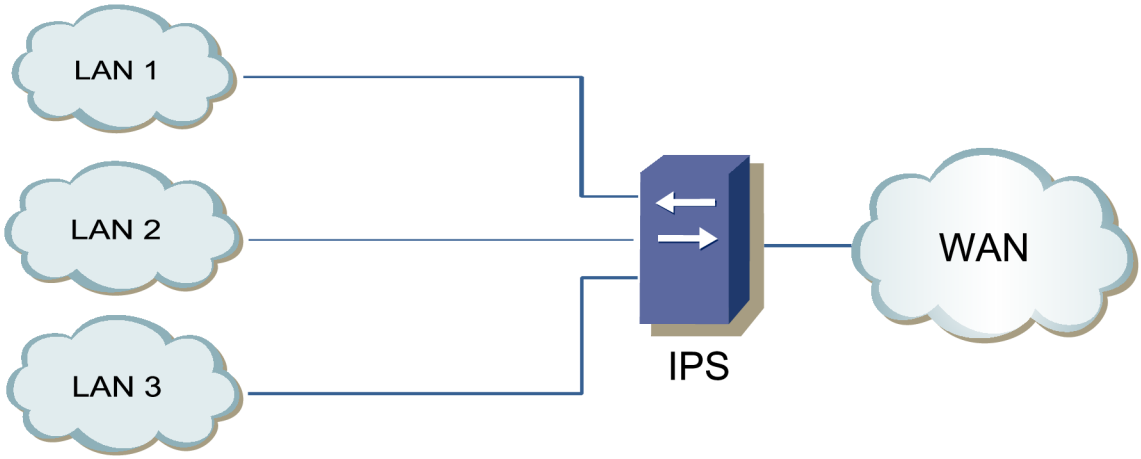
V síti se však nezbytně nemusí vyskytovat taková úzká místa, vhodná pro umístění IPS senzoru. V případě absence routerů, firewallů či mostů je tedy výběr vhodného umístění senzoru více problematické. Existují však případy, kdy je možné škrtící body uměle vytvořit. Příkladem může být propojení dvou VLANů mostem. Tato situace je naznačena na obrázku 3.6. Tím lze uměle vytvořit úzké místo, kde lze IPS senzor umístit. Takové řešení není sice úplně nejkorektnější, avšak umožňuje však oddělení a vzájemnou ochranu sítí.

Dalším příkladem může být umístění IPS senzorů v rozlehlých oblastech. V síti WAN je sice možné užití IPS senzorů, ale je nutné je umístit mezi LAN a sítěmi WAN. Při tomto rozložení je možné umístění jednoho IPS senzoru do každé vzdálené oblasti a jednoho nebo více senzorů do středních nebo velkých sítí. Realizací takového



Obr. 3.6: Umělé vytvoření úzkého místa pomocí mostu.

návrhu však podstatně roste celková cena. Řešením může být umístění jednoho centralizovaného senzoru, jak naznačuje obrázek 3.7. Toto řešení s sebou však přináší také omezené v podobě odesíláním dat pouze skrze jednu síť.[2]



Obr. 3.7: Připojení několika LAN k sítí WAN.

3.5 Porovnání HIPS a NIPS

- HIPS je schopen kontrolovat také šifrovanou komunikaci. Respektive má přístup v místě, kdy není šifrována.
- NIPS nezatěžuje výpočetní prostředky hostitelského systému, protože používá své vlastní.
- NIPS je vlastně bodem možného selhání sítě, což může být považováno za nevýhodu. Tento fakt ovšem platí také pro ostatní síťová zařízení, jako například routery, switche nebo firewally.
- NIPS disponuje lepší detekční schopností při událostech, které probíhají rozptýleně v rámci celé sítě na nízkých vrstvách, jako například skenování stanic. Kdy HIPS disponuje pouze svými lokálními informacemi a komunikace s centrálním uzlem by byla časově náročná a neefektivní.[8]

4 SNORT

Snort je volně šířitelný software při dodržení licence GNU GPL v.2. Svou působností spadá do kategorie IPS i IDS software. V současné době je jeho nasazení možné na platformách Linux a Windows.[22]

Umožňuje analýzu protokolů, prohledávání kontextu a je obvykle používán k aktivní blokaci nebo pasivní detekci různorodých typů útoků, jako například přetečení bufferu, skenování portů, napadení webové služby, SMB protokolu, zjišťování typu operačního systému a mnoho dalších. Snort může být použit ve spojení s dalšími programy, jako například Squil, OSSIM nebo BASE, sloužících ke grafické prezentaci dat o útoku.[21]

4.1 Režimy

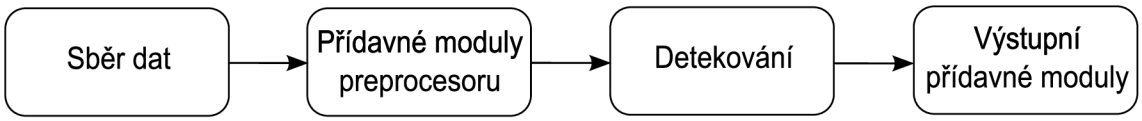
Snort je možné provozovat ve čtyřech režimech:

- Zachytávání paketů – provádí zachytávání probíhající síťové komunikace.
- Zaznamenávání paketů – jedná se vlastně o rozšířené zachytávání, kdy data obsažená v paketech jsou zaznamenávána do souboru na médium, jakým bývá nejčastěji pevný disk.
- Mód NIDS – v tomto režimu provádí zachytávání síťové komunikace a na základě definovaných pravidel provádí její analýzu. Případně další příslušné akce, jako například vyhlášení výstrahy.
- Mód NIPS – je modifikací módu NIDS. Zásadním rozdílem je schopnost vyzvedávat pakety z NetFilter[1] nebo IPFW[2], které porovná s pravidly. Při vyhodnocení závadné komunikace dochází k jejímu zahazování.

4.2 Komponenty

Vnitřní strukturu programu Snort lze logicky rozdělit do řetězce čtyř částí, které společně úzce spolupracují. Umístění a vzájemnou komunikaci jednotlivých komponent zachycuje obrázek 4.1.

[1] Modul Linuxového jádra, umožňující zachytávání a manipulaci se síťovými pakety.
[2] IPFW je obdobou NetFilter. Nasazen a sponzorován je projektem FreeBSD.



Obr. 4.1: Schéma vnitřního uspořádání systému Snort.

4.2.1 Komponenta pro sběr dat

Tato komponenta slouží primárně k zachycení sledované komunikace a jejímu předání následující komponentě ke zpracování. Proces sběru dat bývá realizován s pomocí některých knihoven, jako například libcap nebo Libnet. Lze pomocí nich přijímat datagramy z linkové vrstvy, obsahující datové jednoty následujících vrstev. Mají tedy přístup ke komunikaci v nejnižším místě sedmivrstvého síťového modelu, pokud pomineme fyzickou vrstvu. Ta však spadá pro systémy detekce a prevence mimo oblast zájmu.

4.2.2 Přídavné moduly preprocesoru

Mají za úkol testování a kontrolu příchozích dat, zachycených komponentou pro sběr dat. Určují jak dále naložit s každou jednotlivou částí zachycené komunikace. Zda má být předána k analýze, modifikována, zahozena nebo zda má být vydána výstraha. Výhodou při nasazení této komponenty je stanovování struktury dat, určených k dalšímu zpracování. Provádějí například změny formátu URI a URL adres na standardizovaný tvar, detekci skenování portů nebo dekódování paktů rozličných protokolů.

4.2.3 Detekční komponenta

Komponenta provádí dekódování příchozích paketů a jejich umísťování do struktury podle čísla vrstvy komunikačního modelu. Toto zařazení usnadňuje systematické porovnání dat s definovanými pravidly. Příkladem může být situace, kdy je na základě jednoho pravidla paket, nebo jeho část, porovnáván, zda obsahuje sledovanou hodnotu nebo sekvenci znaků. Toto procedura se opakuje pro všechna přijatá data a definovaná pravidla.

4.2.4 Výstupní přídavné moduly

Pomocí komponenty zásuvných modulů dochází k zobrazení nebo jinému způsobu prezentace výsledků analýzy.[20]

4.3 Pravidla

Velice důležitou částí IDS/IPS Snort jsou pravidla. Jejich špatném nasazení či úplná absence může způsobit nefunkčnost systému. Vyznačují se především jednoduchou syntaxí a flexibilitou. Každé pravidlo se skládá z hlavičky a těla, jak naznačuje jeho obecný formát.

```
akce protokol IP_zdroj port_zdroj směr IP_cil port_cil (tělo)
```

4.3.1 Hlavička

Obsahuje sedm povinných údajů, bez kterých by pravidlo nemohlo být použito. Budou popsány jednotlivé části včetně hodnoty, kterých mohou nabývat.

Akce

Definuje, jak má být s paketem naloženo. Následující možnosti jsou platné pro IDS Snort.

- alert – vygeneruje výstrahu o události
- pass – ignoruje pravidlo
- log – pouze zaznamená do souboru
- dynamic – pravidlo není použito, dokud není jiným aktivováno
- activate – vygeneruje výstrahu a aktivuje dynamické pravidlo

Pokud je Snort provozován v režimu inline, je nabídka akcí rozšířena o následující tři.

- drop – provede zahození paketu a vydání výstrahy
- sdrop – provede pouze zahození paketu
- reject – jako drop, ale odešle druhé straně upozornění o zahození

Protokol

V současné době jsou podporovány jsou tři typy protokolů. Dva z rodiny IP, pracující na transportní vrstvě. Třetí je provozován na síťové vrstvě modelu ISO OSI, sloužící pro monitorování stavu sítě a zasílání potřebných informací.

- TCP
- UDP
- ICMP

IP adresy

Zdrojová (`IP_zdroj`) a cílová (`IP_cil`) adresa jsou zadávány ve standardním formátu, definovaným pomocí CIDR.

Porty

Zadávané hodnoty zdrojového (`zdroj_port`) a cílového (`cil_port`) portu musí být v obecně platném rozsahu `0 - 65535`.

Rozsahy IP adres a portů

Výčet všech IP adres či portů je možné zadat pomocí operátoru `any`. Definování potřebného rozsahu hodnot je možné při dodržení následující syntaxe.

- `any` – zastupuje celý rozsah adres či portů
- `![x]` – vyloučení `x` z rozsahu `any`
- `[x,y]` – výčet zadaných hodnot
- `[x:y]` – definování rozsahu hodnot: `x` až `y`
- `[:y]` – jednostranná definice rozsahu: 0 až `y`
- `[x:]` – jednostranná definice rozsahu: `x` až 1024
- `[x,y:z]` – kombinace operací

Směr

Pomocí operátoru je možné definovat, zda má být pravidlo aplikováno na konkrétní směr komunikace, či má mít obousměrnou platnost.

- `->` – platnost pro jeden směr
- `<>` – obousměrná plantnost

4.3.2 Tělo pravidla

Vlastní tělo pravidla může obsahovat poměrně velké množství rozličných parametrů. Základní syntaxí pro jejich tvorbu je vzájemné oddělení středníkem. Argumenty jednotlivých parametrů jsou poté oddělovány dvojtečkou. Rozčlenění jednotlivých parametrů je možné přibližně do čtyř skupin.[23]

- obecné
- detekování obsahu paketu
- detekování hlavičky paketu
- následně prováděné operace

Obecné

Většina obecných parametrů bývá užita v každém z pravidel. Některé jsou přímo vyžadovány, jako například `sid`.

- msg – text výstrahy
- classtype – zařazování do tříd událostí dle `classification.config`
- sid – jedinečný identifikátor pravidla
- gid – identifikace komponenty, která výstrahu vygenerovala
- rev – revize pravidla
- refernce – odkaz na externí zdroj, popisující útok
- priority – nastaví priority pravidlu

Detekce obsahu hlavičky

Následující parametry umožňují detekovat pole příznaků, nesené hlavičkou paketu.

- ttl – doba životnosti paketu
- flags – nastavené příznaky v poli Flags
- tos – hodnoty nastavené v poli TOS
- ack – sekvenční čísla potvrzovacích paketů
- itype – specifický typ ICMP zprávy
- window – nastavenou velikost okna přijatého TCP paketu

Detekce obsahu nesených dat

Pomocí těchto parametrů lze kontrolovat, zdali paket přenáší hledanou strukturu dat.

- content – hledání zadaného řetězce
- nocase – nerozlišuje malé a velké znaky
- rawbytes – deaktivuje dekódování a vyhodnocuje surový obsah paketu
- dsize – testuje velikost přenášeného obsahu
- offset – posunutí začátku hledání
- depth – definuje hloubku hledání
- uricontent – prohledává URI požadavek
- urilen – definuje délku URI požadavku

Následné operace

Uvnitř samotného pravidla je možné definovat také dodatečné akce, jako například zaznamenávání konkrétní události do samostatného souboru, nahrazení nalezeného řetězce nebo aktivace dynamického pravidla.

- logto – zaznamenání události do samostatného souboru
- session – umožňuje získat informace o daném TCP spojení
- replace – nahrazuje hledanou hodnotu jinou (pouze mód inline)
- resp – v případě výstrahy ukončuje TCP sezení
- react – ukončuje spojení a vystaví výstrahu
- activates – aktivace dynamického pravidla
- activated_by – určení aktivačního zdroje (v dynamických pravidlech)
- count – počet propuštěných paketů před aktivací dynamického pravidla

Příkladem může být následující pravidlo, které definuje zahazování paketů TCP protokolu, přicházejících se zdrojovou adresou 192.168.30.5, pokud je hodnota životnosti v hlavičce nižší, než dva. Vygenerovaná výstraha bude obsahovat zprávu o události `Droped TCP Packet low TTL` a identifikátor pravidla `sid`.

```
drop tcp 192.168.30.5/24 any -> any any (msg:"Dropped TCP \
Packet with low TTL"; ttl: < 2; classtype:network-scan; \
sid:1000006; rev:1;)
```

4.4 Formát výstrah

Výstrahy, generované softwarem Snort, mají textovou podobu. Není tedy nutné použití dalšího nástroje, který by umožnil srozumitelnou reprezentaci obsahu. Příkladem může být výstraha vygenerovaná na základě příchozího TCP paketu s krátkou dobou života.

```
[**] [1:1000006:1] Dropped TCP Packet with low TTL [**]
[Classification: Detection of a Network Scan] [Priority: 3]
05/05-23:57:46.358699 192.168.30.5:60464 -> 192.168.30.100:9000
TCP TTL:1 TOS:0x0 ID:20714 IpLen:20 DgmLen:44
******S* Seq: 0xFF4ED176  Ack: 0x0  Win: 0x800  TcpLen: 24
```

Význam jednotlivých položek bude vysvětlen pro každý řádek zvlášť.

První

- `1:` – identifikátor generující komponenty (GID)
- `1000006` – identifikátor pravidla
- `:1` – revizní číslo pravidla
- `Droped TCP Packet with low TTL` – krátký popis události

Druhý

- zařazení do třídy podle typu detekované události
- výpis konkrétní priority dané události

Třetí

- `05/05-23:57:46.358699` – časové razítko
- `192.168.30.5` – zdrojová adresa
- `60464` – zdrojový port
- `-->` – směr události
- `192.168.30.100` – cílová adresa
- `9000` – cílový port

Čtvrtý

- `TCP` – typ protokolu
- `TTL:1` – doba života paketu
- TOS:0x0 – typ služby [3]
- `ID:20714` – identifikátor paketu
- `IpLen:20` – velikost IP hlavičky
- `DgmLen:44` – celková velikost paketu

Pátý

- `******S*` – nastavené příznaky v IP hlavičce (SYN)
- `Seq: 0xFF4ED176` – pořadové číslo paketu
- `Ack: 0x0` – pořadové číslo potvrzení
- `Win: 0x800` – velikost okna
- `TcpLen: 24` – velikost TCP části

[3] Podle typu služby jsou pakety zařazovány do front různých priorit při použití QoS.

4.5 Nastavení Libpcap

Standardně jsou pakety kopírovány z paměti jádra od uživatelské paměti, odkud jsou zpracovávány systémem Snort. Instalací knihovny LibPCAP dojde k implementaci sdíleného „kruhového bufferu“. Při vyžití takto vytvořeného paměťového místa jsou pakety umisťovány přímo zde, odkud je systému Snort umožněno přímé zpracování. Jedná se tedy o nejrychlejší nabízený způsob předávání dat.

Maximální počet těchto sdílených kruhových bufferů je 32768. Pokud je použita technologie Ethernet s velikostí přenášených rámců 1530 bytů, je potřeba 52 Megabytů paměti. Požadovaný počet se nastavuje pomocí proměnné PCAP_FRAMES, do které uložíme cílový počet. Dalším možností je nechat knihovnu určit nejvyšší možnou hodnotu.[23]

```
PCAP_FRAMES = max
```

4.6 Inline mód

Základem provozování systému Snort inline je firewall implementovaný v OS Linux, který obsahuje řetězce INPUT, OUTPUT a FORWARD. Každý z nich může obsahovat pravidla včetně definované akce.

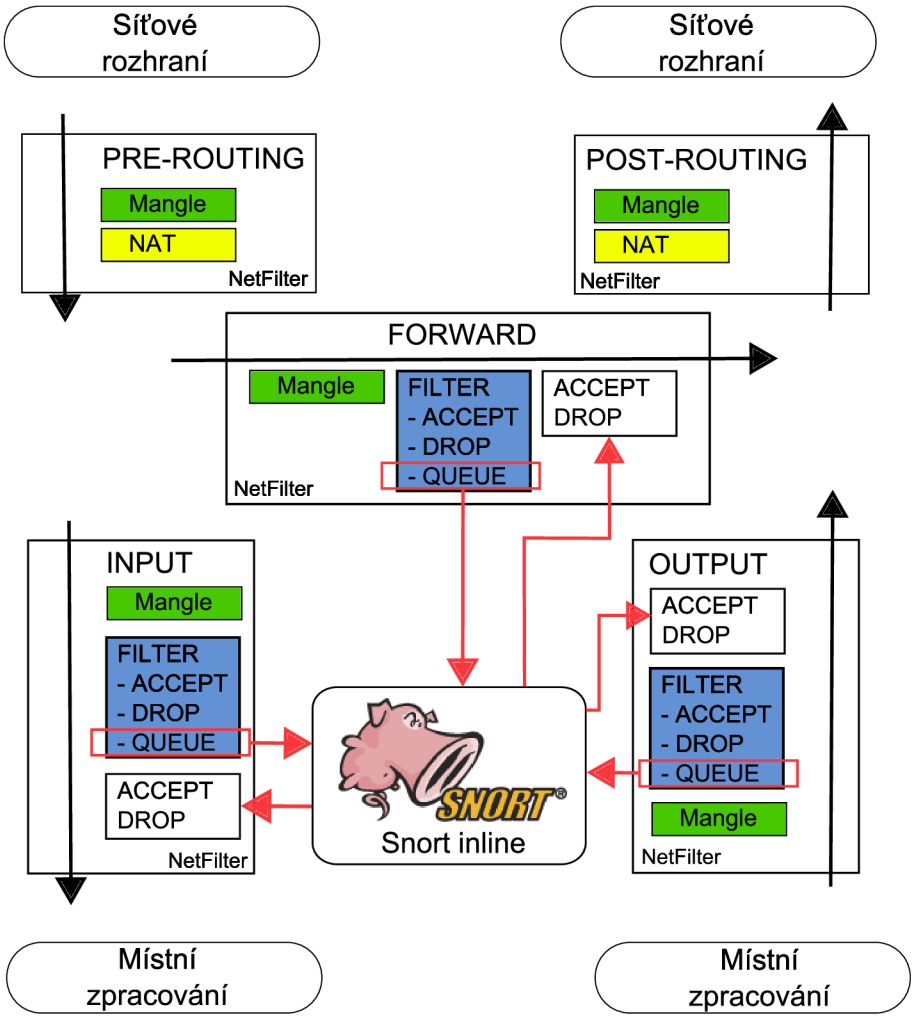
- ACCEPT – povolení
- DROP – zahození
- QUEUE – zařazení do fronty

Názorným příkladem může být zařazení příchozí komunikace do fronty, pokud je užitým protokolem TCP a cílový port HTTP. Pomocí nástroje iptables pravidlo přidáme.

```
iptables -I INPUT -p tcp --dport 80 -j QUEUE
```

Snort pracující v režimu inline vyzvedává pakety z front a porovnává je s vlastními pravidly, na základě kterých provádí jednu z definovaných akcí, popsaných v kapitole 4.3.1. Jednoduchým příkladem mohou být `alert` a `drop`. Při akci `alert` dochází k navrácení paktu zpět do řetězce a vydání varování. V případě `drop` jsou pakety rovnou zahozeny. Vnitřní uspořádání NetFilteru[4] a jejich zpracování softwarem Snort zachycuje obrázek 4.2.

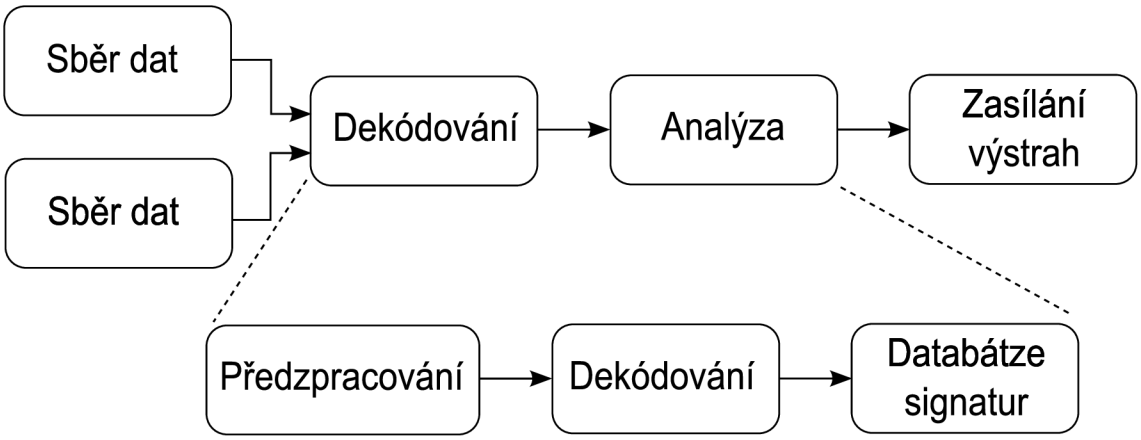
[4]Modul Linuxového jádra, umožňující zachytávání a manipulaci se síťovými pakety.



Obr. 4.2: Přebírání paketů z tabulek NetFilteru.

5 OSSEC

Ossec je multiplatformní, škálovatelný, volně šiřitelný software při dodržení licence GNU GPL v.3. Umožňuje vytváření a následnou správu kontrolních součtů souborů, analýzu výpisů z programů či vytváření záznamů o přihlašovacích procedurách. Je možné ho provozovat na širokém spektru platforem, jako například Linux, BSD, Solaris, Windows, AIX, MAC nebo Vmware ESX.[15] Jeho vnitřní schéma zachycuje obrázek 5.1



Obr. 5.1: Schéma vnitřního uspořádání systému OSSEC.

OSSEC svým zaměřením velkou měrou spadá do kategorie bezpečnostních systémů, umístěných na hostiteli. Detekci průniku vyhodnocuje na základě prováděných kontrol.

- analýza systémových výpisů
- kontrola integrity souborů
- monitorování registrů
- detekce rootkitů

V případě nalezené hrozby nebo nestandardního chování je vydáno varování. Je také schopen zamezit dalšímu trvání hrozby vykonáním příslušné akce. Ta však musí být pro daný typ události definována v sekci `active-response`.[1]

5.1 Konfigurace

Nastavení parametrů, definujících chování detekčního systému OSSEC, je možné pomocí konfiguračního souboru `ossec.conf`. Údaje jsou zde uloženy v textové podobě ve formátu XML. Obsahovat může následující sekce.

- global – nastavení platná v rámci celého systému
- syscheck – kontrola integrity
- rootcheck – detekování rootkitů
- localfile – monitorování lokálních souborů
- active-response – nastavení aktivní odezvy
- client – nastavení agentů
- rule – zadání cesty k souboru s pravidly
- remote – konfigurace připojení ke vzdálené straně
- alerts –nastavení formátu výstrah
- database_output – nastavení parametrů pro připojení k databázi

Zmíněné položku nemají obecnou platnost, tedy některé z nich je možné uplatnit pouze na OSSEC serverech či agentech.[5]

5.2 Správa agentů

Aby bylo umožněno připojení agentů k OSSEC serveru, musí být splněno několik podmínek. Každý agent musí mít na serveru definovaný záznam, obsahující název, identifikační číslo a klíč. Správu jednotlivých agentů je možné provádet pomocí manažera umístěného standardně ve složce `/var/ossec/bin/manage_agents`.

Každému agentovi musí být zadána IP adresa OSSEC serveru a klíč. Ten je generován při vytváření záznamu na serveru, ze kterého je následně přenesen pomocí zabezpečeného spojení k agentovi. Klíč umožňuje jednoznačnou identifikaci agenta.

Vlastní komunikace mezi agenty a serverem probíhá pomocí nespojově orientovaného protokolu UDP na portu 1514.

5.3 Formát výstrah

Výstrahy generované systémem OSSEC jsou ukládány v textové podobě. Nacházejí se ve složce `/var/ossec/logs/alerts`. Následně jsou odesílány elektronickou poštou

na adresu zadanou při instalaci, která je popsána je v kapitole 8.2.1. Příkladem může být následující výstraha, informující o připojení nového OSSEC agenta.

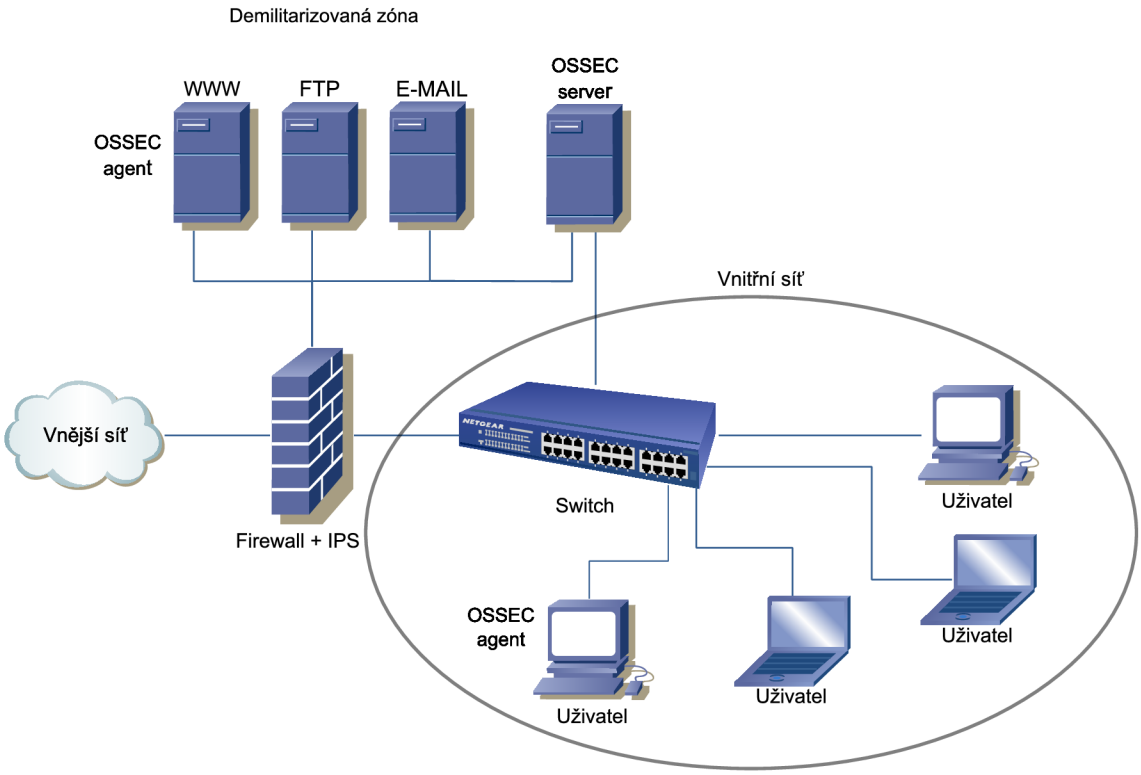
```
OSSEC HIDS Notification.
2010 Mar 30 18:22:44

Received From: (SVR020) 192.168.20.20->ossec
Rule: 501 fired (level 3) -> "New ossec agent connected."
Portion of the log(s):

ossec: Agent started: 'SVR020->192.168.20.20'.
--END OF NOTIFICATION
```

Zaslaná výstraha obsahuje položky, jejichž význam je zde popsán.

- `2010 Mar 30 18:22:44` – časové razítko
- `SVR020` – identifikátor OSSEC agenta, který varování zaslal
- `Rule: 501 fired (level 3)` – číslo pravidla a stupeň závažnosti
- „`New ossec agent connected.`" – krátký popis události
- „`ossec: Agent started: ...`" – popis události

6 REALIZOVANÉ ŘEŠENÍ

Cílem praktické části byl návrh menší firemní sítě včetně implementace systémů, sloužících k detekci změn, nestandardního chováni a ochranně před útoky. Rozmístění a propojení jednotlivých prvků sítě je zachyceno na obrázku 6.1.



Obr. 6.1: Schéma realizované sítě.

V rámci demilitarizované zóny jsou instalovány tři servery, postavené na platformě Linux. Jako vhodná varianta se nabízí distribuce CentOS 5.4. Servery poskytují služby uživatelům přistupujícím jak z místní, tak i venkovní sítě. Mezi provozované služby patří přístup na firemní webové stránky, sdílení souborů mezi definovanými uživateli nebo možnost komunikace pomocí elektronické pošty. K vlastním prvkům DMZ není možné přistupovat přímo, protože jsou skryté. Uživateli jsou k dispozici pod IP adresou vnitřního či vnějšího rozhraní prvku IPS. Výjimkou je `E-MAIL` server, který je přístupný pouze z vnitřní sítě.

Významným prvkem je IPS, sloužící jako centrální propojovací uzel mezi demilitarizovanou zónou, vnitřní a vnější sítí. Vykonává zde funkci firewallu a routeru, jejichž běh bude zajišťovat operační systém CentOS. Toto centrální umístění je zároveň výhodné pro nasazení systému prevence průniku, realizované pomocí open-source softwaru Snort.

Součástí navrhovaného řešení je také detekční systém OSSEC, sloužící ke sledování nežádoucích aktivit jednotlivých prvků sítě. Varovné zprávy jsou zasílány agenty, provozovanými na koncových stanicích i serverech. Důležitým článkem je server, který zprávy o událostech přijímá. Pro vlastní běh programu bylo využito shodného OS jako v případě prvku IPS.

Při volbě programového vybavení jednotlivých prvků sítě byl brán zřetel na několik aspektů. Mezi hlavními byla jejich cena, efektivita nebo náročnost realizace řešení. Celá síť, zachycená na obrázku 6.1, byla z důvodu nedostatku potřebného fyzického hardware realizováno ve virtuálním prostředí.

Následující kapitoly jsou věnovány především popisu jednotlivých kroků, vedoucích ke správné konfiguraci jednotlivých prvků sítě. Instalaci a nastavení bezpečnostních prvků, jejichž správná funkce bude ověřena provedením několika útoků, směřovaných především na dostupnost poskytovaných služeb.

7 NASTAVENÍ PRVKŮ SÍTĚ

Tato kapitola je věnována především nastavení IP adres prvků a povolení potřebných služeb na firewallu. Pro menší firemní síť, zachycenou na obrázku 6.1, bylo nutné definovat rozsahy adres pro jednotlivé části. S ohledem na přehlednost byly zvoleny následující rozsahy.

- 192.168.10.0/24 – Vnitřní síť
- 192.168.20.0/24 – Demilitarizovaná zóna
- 192.168.30.0/24 – Externí síť

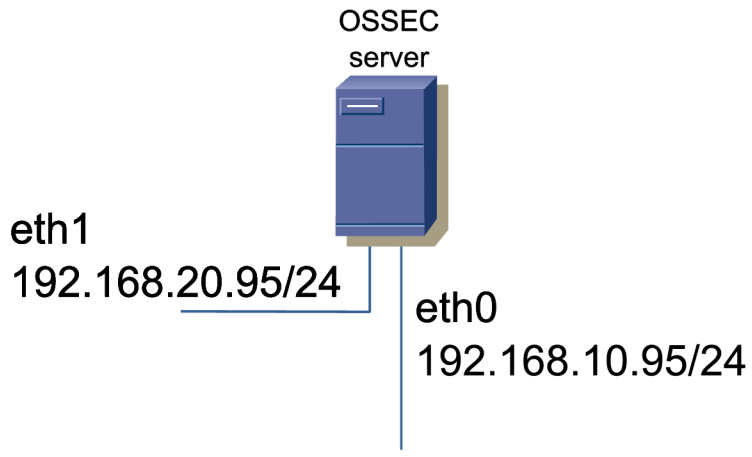
Jednou z možností při povolení síťových portů v systému CentOS je použití systémového konfiguračního nástroje.

```
system-config-securitylevel-tui
```

V průběhu instalací jednotlivých služeb pomocí něho bude nutné povolit vypsané porty, na kterých naslouchají služby. Pokud by nastavení neproběhlo, síťový firewall by komunikaci blokoval. Funkcionalita celé sítě by poté byla silně omezena.

7.1 OSSEC server

Server disponuje dvěma síťovými rozhraními, která jsou připojená do demilitarizované zóny a místní sítě. Obě slouží pouze pro příjem výstrahy, zasílaných OSSEC agenty. Nastavení IP adres síťových rozhraní zachycuje obrázek 7.1.



Obr. 7.1: Nastavení rozhraní OSSEC serveru.

Pro příjem zpráv je nutné povolit ve firewallu příchozí komunikaci na portu 1514 pro protokol UDP.

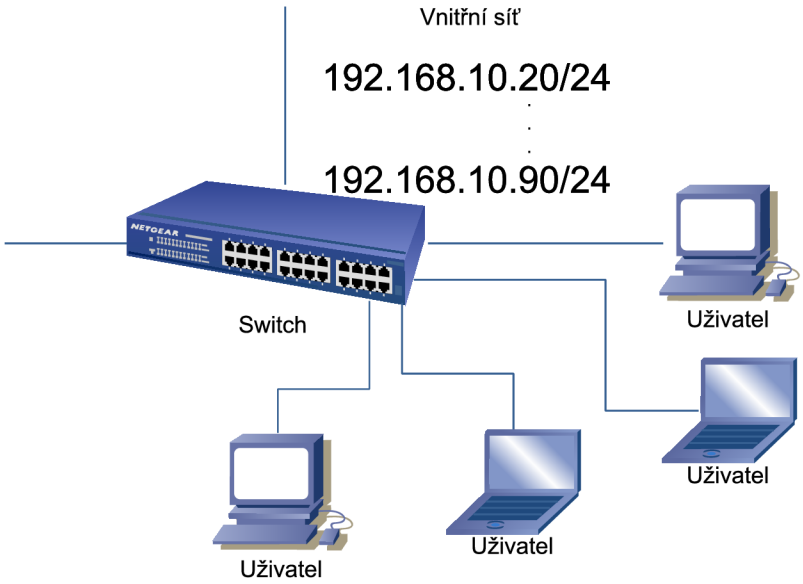
7.2 WWW, SFTP a E-mail server

V rámci úspor hardwarových prostředků jsou všechny tři servery, poskytující služby, provozovány na jednom počítači. Disponuje jedním síťovým rozhraním a staticky nastavenou IP adresou `192.168.20.20/24`. Ve firewallu je třeba povolit porty, na kterých naslouchají poskytované služby.

- 25 – SMTP
- 22 – SSH
- 80 – HTTP
- 143 – IMAP
- 443 – HTTPS
- 2222 – SFTP

7.3 Místní síť

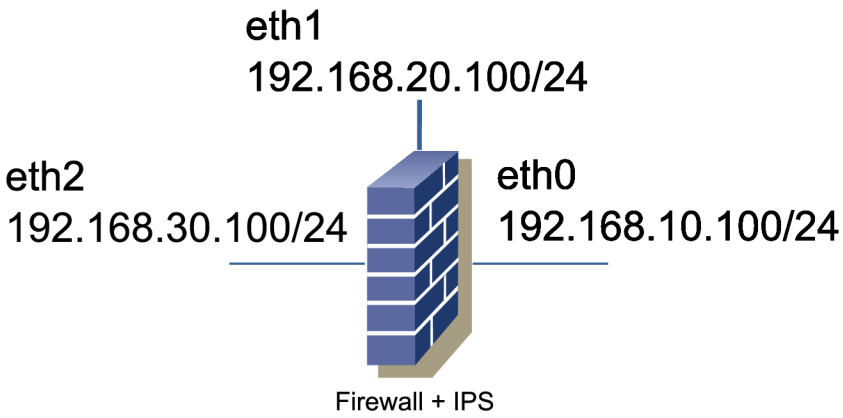
Jednotlivým uživatelům místní sítě jsou IP adresy přidělovány automaticky pomocí DHCP serveru, umístěném na prvku IPS. Rozsah přidělovaných adres je zachycen na obrázku 7.2.



Obr. 7.2: Rozsah přidělovaných adres v místní síti.

7.4 Firewall s IPS

Prvek IPS je díky svému centrálnímu umístění jednou z nejvýznamnějších částí celé sítě. Vykonává zde funkci routeru, firewallu a provádí kontrolu příchozí i odchozí komunikace. Disponuje třemi rozhraními, jejichž IP adresy jsou definovány staticky. Konkrétní nastavení zachycuje obrázek 7.3.



Obr. 7.3: Nastavení síťových rozhraní prvku IPS.

Firewall

Prvek IPS plní také funkci firewallu, slouží k oddělení vnitřní sítě od vnější a blokování nechtěné komunikace. Pro přístup ke službám, k implementovaným v demilitarizované zóně, je třeba také na prvku IPS povolit následující porty.

- 25 – SMTP
- 22 – SFTP
- 80 – HTTP
- 143 – IMAP
- 443 – HTTPS

Možnost vzdálené konfigurace a správy samotného prvku umožníme povolením portu 2222, na který je přenesena služba SSH.

DHCP server

IPS prvek sítě vystupuje také v roli serveru pro přidělování IP adres stanicím umístěným v místní síti. Na systému CentOS provedeme jeho instalaci pomocí správce balíčků.

```
yum install dhcpd
```

Dále přidáme do jeho konfiguračního souboru `dhcpd.conf`, umístěným ve složce `/etc`, následující nastavení.

```
subnet 192.168.10.0 netmask 255.255.255.0 {
   option routers                  192.168.10.100;
   option subnet-mask              255.255.255.0;
   option domain-name-servers      192.168.30.1;
   range 192.168.10.20 192.168.10.90;
}
```

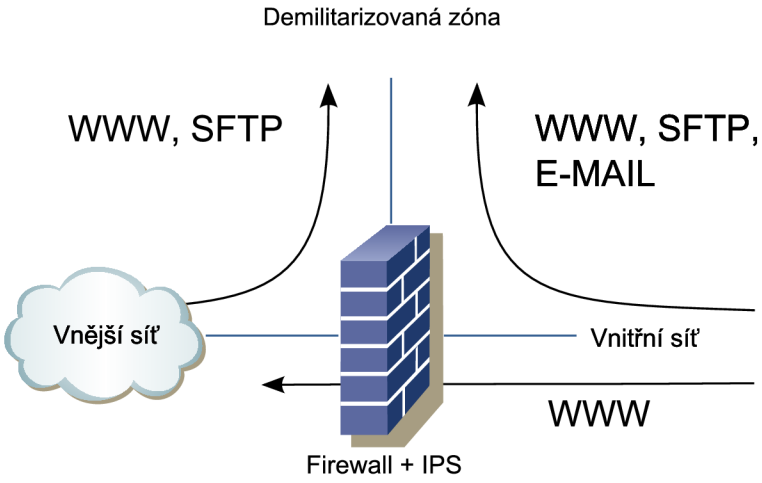
Změna portu SSH

Na prvku IPS je třeba změnit nastavení služba SSH. Standardní port 22 bude použit pro přístup uživatelů k přenosu souborů. Provedeme editaci souboru `sshd_config`, který je umístěn ve složce `/etc/ssh`, kde provedeme změnu použitého portu na 2222 a službu restartujeme.

```
/etc/init.d/sshd restart
```

Router

Služby umístěné v demilitarizované zóně jsou poskytovány skrytě. Uživatelům jsou k dispozici odesláním požadavku na vnější či vnitřní rozhraní prvku IPS. Výjimku je elektronická pošta, která je dostupná pouze místně. Dále je umožněn přístup z lokální sítě na webové stránky, umístěné v rámci Internetu. Celkové schéma dostupných služeb je zachyceno na obrázku 7.4.



Obr. 7.4: Schéma dostupnosti poskytovaných služeb.

Prvním krokem při zpřístupňování služeb demilitarizované zóny je na prvku IPS nutné povolit předávání komunikace mezi jednotlivými rozhraními. Jednorázové nastavení lze provézt zápisem znaku jedna do souboru `ip_forward`.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Trvalé povolení je možné editací souboru `/etc/sysctl.conf` a nastavení proměnné `forwarding` na hodnotu jedna.

```
net.ipv4.conf.default.forwarding = 1
```

Opětovnou inicializaci nastavení je nutné provádět při každém startu systému nebo změně operačního módu[1]. Zajištění této funkce realizujeme následujícím příkazem.

```
echo sysctl -p >> /etc/rc.d/rc.local
```

Další částí je nastavení jednotlivých tabulek NetFilteru, jehož schéma zachycuje obrázek 4.2. Nejprve provedeme přesměrování požadavků, přicházejících na vnější rozhraní prvku, pro službu WWW a SFTP v tabulce PRE-ROUTING.

```
iptables -t nat -A PREROUTING -i eth2 -d 192.168.30.100 \
-p tcp --dport 80 -j DNAT --to-destination 192.168.20.20
iptables -t nat -A PREROUTING -i eth2 -d 192.168.30.100 \
-p tcp --dport 22 -j DNAT --to-destination $192.168.20.20:2222
```

Komunikace směřující k odlišnému rozhraní musí být povolena v tabulce FORWARD. Akci prováděnou NetFilterem s pakety, které vyhovují pravidlu nastavíme QUEUE. Tedy zařazování do fronty, odkud jsou programem Snort vyzvednuty a kontrolovány.

```
iptables -A FORWARD -i eth2 -m state --state NEW,RELATED, \
ESTABLISHED -d 192.168.20.20 -p tcp -j QUEUE
```

Poslední částí je povolení navázané komunikace v tabulce FORWARD, která směřuje od serverů zpět k uživateli.

```
iptables -A FORWARD -i eth1 -o eth2 -m state --state \
RELATED, ESTABLISHED -p tcp -j QUEUE
```

Umožnění přístupu z vnějšího rozhraní ke službám WWW a SFTP je pouze krátkým výňatkem z realizovaného směrování. Kompletní nastavení je uloženo v souboru `routing.bash` na přiloženém médiu.

[1] Runlevel

8 INSTALACE

V této kapitole jsou postupně popsány jednotlivé kroky, vedoucí k požadované funkcionalitě celé sítě. V první části bude provedena instalace a nastavení bezpečnostních prvků. Další část je věnovaná postupům, vedoucím k požadované úrovni implementovaných služeb.

8.1 Snort inline

Instalace

Open source systém prevence průniku Snort bude provozován na operačním systému Linux, distribuci CentOS 5.4 i386. Před započetím vlastní instalace Snortu musí operační systém obsahovat nástroje potřebné pro instalaci. Jejich instalace do systému CentOS je možná pomocí správce balíčků, zadáním následujícího příkazu.

```
yum install gcc gcc-c++ autoconf make libtool flex bison\
byacc pcre pcre-devel
```

Po úspěšném doplnění potřebných nástrojů, potřebných k dalším instalacím, je nutné dodat knihovny, nezbytné pro spuštění a provozování IPS Snort.

8.1.1 Libpcap

Knihovna sloužící k zachytávání paketů ze sítě. Při jejím využití musí všechna přijímána či odesílána data projít tímto mechanismem. Poskytuje také rozhraní aplikacím na vyšších vrstvách, provádějících vlastní zpracování.[24] Stažení rozbalení provedeme do složky `/usr/src` pomocí následujících příkazů.

```
wget www.tcpdump.org/release/libpcap-1.1.1.tar.gz
tar xvzf libpcap-1.1.1.tar.gz
cd libpcap-1.1.1
```

Dále provedeme kroky slouží k vlastní kompilaci a instalaci do systému.

```
./configure --prefix=/usr
make
make install
```

8.1.2 Libdnet

Libdnet poskytuje zjednodušené rozhraní pro práci se síťovými pakety. Skládá se z několika programů, které umožňují provádění rozličných operací.[9]

- manipulace síťových adres
- manipulaci a vyhledávání v arp, cache a routovacích tabulkách
- manipulaci a vyhledávání síťových rozhraní
- tunelování IP protokolu
- přeposílání Ethernet rámců a surových IP paketů

Knihovnu je možné získat z domovských stránek [9] projektu. Stažení a rozbalení provedeme do složky `/usr/src` pomocí následujících příkazů.

```
cd /usr/src
wget prdownloads.sourceforge.net/libdnet/libdnet-1.11.tar.gz
tar xzvf libdnet-1.11.tar.gz
cd libdnet-1.11
```

Následují kroky sloužící k vlastní kompilaci a instalaci do systému.

```
./configure --prefix=/usr
make
make install
```

8.1.3 Libnet

Libnet je softwarovým nástrojem, umožňující sestavování sítových paketů. Systém Snort vyžaduje verzi `1.0.x`. Ta ovšem není na domovských stránkách k dispozici. Na stránkách `www.filewatcher.com` byl zvolen jeden z možných odkazů. Stažení a rozbalení souborů provedeme do složky `/usr/src` následujícími příkazy.[10]

```
cd /usr/src
wget ftp://ftp.eenet.ee/pub/gentoo/distfiles/\
libnet-1.0.2a.tar.gz
tar xvzf libnet-1.0.2a.tar.gz
cd libnet-1.0.2a
```

Provedeme kompilaci a následnou instalaci do systému.

```
./configure --prefix=/usr
make
make install
```

8.1.4 Pravidla

Nyní operační systém obsahuje všechny potřebné komponenty pro instalaci a provozování IPS Snort. Nejprve provedeme stažení zdrojových souborů z domovských stránek [22] projektu do složky `/usr/src`. Následné soubory z archivu extrahujeme.

```
cd /usr/src wget http://dl.snort.org/snort-current/\
snort-2.8.6.tar.gz
tar xzvf snort-2.8.6.tar.gz
cd snort-2.8.6
```

Před započetím vlastní kompilace a instalace je vhodné doplnit Snort o pravidla, podle kterých by mohl vyhodnocoval, zda je přijatá komunikace závadná. Ve stažených zdrojových kódech totiž žádná obsažena nejsou. Provedeme tedy instalaci standardních pravidel, dostupných na domovských stránkách projektu [22]. Podmínkou je bezplatná registrace. Po úspěšném přihlášení provedeme stažení a rozbalení pravidel z archivu.

```
wget dl.snort.org/reg-rules/snortrules-snapshot-CURRENT.tar.gz
tar xvzf ../snortrules*
```

Následuje konfigurace zdrojových kódů, při které provedeme změnu ze standardně nastaveného detekčního režimu na prevenční. Tím je umožněno ověřit obsah paketů ještě před jejich odesláním. Konfiguraci, kompilaci a následnou instalace do systému provedeme.

```
./configure --enable-inline
make
make install
```

8.1.5 Nastavení

Chování systému Snort je možné přizpůsobit editací hlavního konfiguračního souboru `snort.conf`, umístěného ve složce `etc`. Standardní nastavení však ponecháme beze změn. Výjimkou jsou proměnné, definující rozsah adres vnitřní sítě[1] a umístění pravidel. Nastavení upravíme, aby korespondovalo s následujícím vzorem.

```
var HOME_NET 192.168.20.0/24
var EXTERNAL_NET !$HOME_NET
var RULE_PATH /etc/snort_inline/rules
```

[1]Do proměnné HOME_NET byl uložen pouze rozsah IP adres demilitarizované zóny, aby bylo možné detekovat také útoky, generované uživateli místní sítě.

Nyní vytvoříme potřebné cílové složky pro uložená nastavení a výpisy. Upravený konfigurační soubor a pravidla do těchto složek nakopírujeme.

```
mkdir /etc/snort_inline /var/log/snort_inline
cp etc/* /etc/snort_inline/
cp rules/ /etc/snort_inline/ -R
```

8.1.6 Spuštění

Spuštění IPS Snort je závislé na modulu jádra `IP_QUEUE`, který umožňuje zařazování paketů do fronty, odkud jsou následně vyzvednuty a předány k dalšímu zpracování. Provedeme načtení modulu a kontrolu, zda je operace proběhla úspěšně.

```
modprobe ip_queue
lsmod | grep ip_queue
```

Závěrečným krokem je spuštění programu společně s potřebnými parametry, mezi které patří cesta ke konfiguračním souborům, adresáři pro zaznamenávání výstrah nebo nastavení proměnné PCAP_FRAMES, popsané v kapitole 4.5.

```
PCAP_FRAMES=max snort -Qvdc /etc/snort_inline/snort.conf\
-l /var/log/snort_inline -D/
```

Významy použitých parametrů:

- -Q – přejímá pakety z front na všech rozhraních
- -c – cesta ke konfiguračnímu souboru
- -v – povolí výpis do konzole
- -d – zobrazí data aplikační vrstvy
- -l – cesta k adresáři pro zaznamenávání výstrah
- -D – spustí Snort jako službu na pozadí

8.2 OSSEC

OSSEC je open source systém, který je možno provozovat na rozličných platformách. V rámci navržené testovací sítě bude popsán postup instalace pro systémy Linux a Windows. Na platformě Windows je možné provozovat tento detekční systém pouze v roli agenta. Aby bylo umožněno fungování celého systému, je nezbytně nutná také role OSSEC serveru, který je vázán na platformu Linux, Solaris, *BSD, MAC nebo AIX.

8.2.1 Server

Před instalací OSSEC serveru musí operační systém obsahovat nástroje, potřebné pro kompilaci a konfiguraci instalačních skriptů. Jednou z možných cest v systému CentOS je instalace pomocí správce balíčků `yum`. Přidání potřebných nástrojů provedeme následujícím příkazem.

```
yum install gcc gcc-c++ autoconf make
```

Pokud je operace úspěšně dokončena, můžeme přistoupit k dalšímu kroku, kterým je stažení archivu se zdrojovými soubory do složky `/usr/src`.

```
cd /usr/src
wget www.ossec.net/files/ossec-hids-2.4.1.tar.gz
```

Rozbalení zdrojových souborů a následné spuštění instalačního skriptu provedeme následujícími příkazy.

```
tar zxfv ossec-hids-2.4.1.tar.gz
cd ossec-hids*
./install.sh
```

Vlastní instalace probíhá v příkazové řádce formou postupného zadávání požadovaných parametrů. V prvních krocích dochází k volbě jazyku a cílové složky pro instalaci souborů. Český jazyk k dispozici není, ponecháme tedy standardní volbu angličtiny `[en]`. Ze staženého balíčku je možné instalovat následující dři druhy instancí.

- server
- agent
- lokální

Zadáme možnost `[server]`. Cílovou složku pro instalaci ponecháme standardní `/var/ossec`. Dále je nabízena možnost zasílání výstrah elektronickou poštou. Tuto možnost budeme využívat, proto zadáme souhlas `[y]`. Následuje zadání adresy e-mailové schránky a IP adresy SMTP serveru. Položky vyplníme následujícími údaji.

```
admin@test.net
192.168.20.20
```

Dále je možno zvolit, zdali bude instalován také program pro kontrolu integrity souborů a nástroj sloužící k detekci škodlivého software v počítači. V obou případech volíme kladně `[y]`. OSSEC server bude provozován čistě jako HIDS bez jakékoliv aktivní odezvy. Na dotaz odpovíme záporně stiskem klávesy `[n]` a následným potvrzením. V závěru instalace jsou zobrazeny soubory, standardně určené ke kontrole.

```
/var/log/messages
/var/log/secure
/var/log/maillog
```

Spuštění korektně nainstalováného OSSEC serveru provedeme následujícím příkazem.

```
/etc/init.d/ossec start
```

OSSEC server nyní pracuje správně. Neobsahuje však potřebné záznamy o agentech, které umožňují jejich přihlášení. Správu agentů má na starost jejich manažer. Jeho spuštění provedeme následujícím příkazem.

```
/var/ossec/bin/manage_agents
```

Zadáním volby `[A]` vytvoříme nový záznam o agentovi s identifikačním číslem 020, který představuje jednu ze stanic místní sítě.

```
****************************************
* OSSEC HIDS v2.4.1 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
```

```
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q:   A

- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
   * A name for the new agent: Host020
   * The IP Address of the new agent: 192.168.10.20
   * An ID for the new agent[001]: 020

Agent information:
   ID:020
   Name:Host020
   IP Address:192.168.10.20
Confirm adding it?(y/n): y
Agent added.
```

Postup pro přidání agenta s identifikačním číslem 120, běžícího na webovém serveru je obdobný. V průběhu zadávání vyplníme následující údaje.

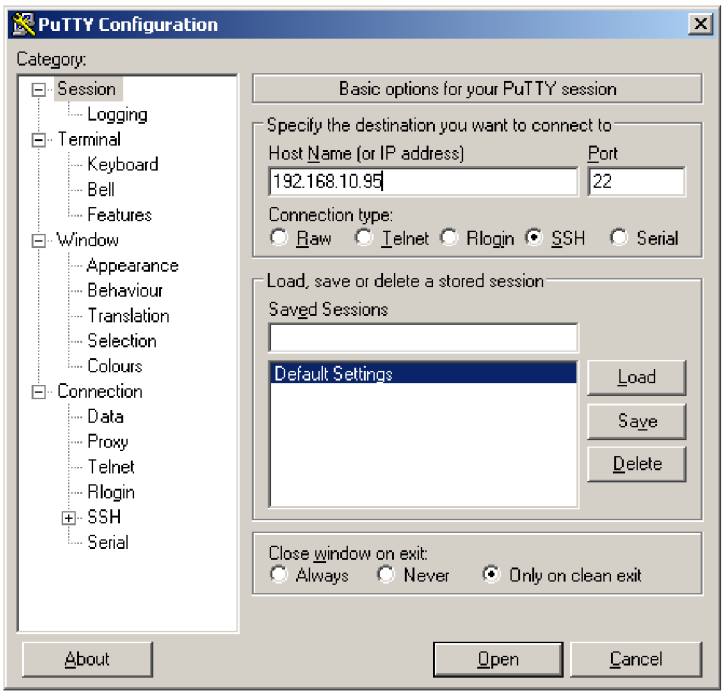
```
   * A name for the new agent: SVR020
   * The IP Address of the new agent: 192.168.20.20
   * An ID for the new agent[001]: 120

Agent information:
   ID:120
   Name:SVR020
   IP Address:192.168.20.20
Confirm adding it?(y/n): y
Agent added.
```

8.2.2 Agent pro Windows

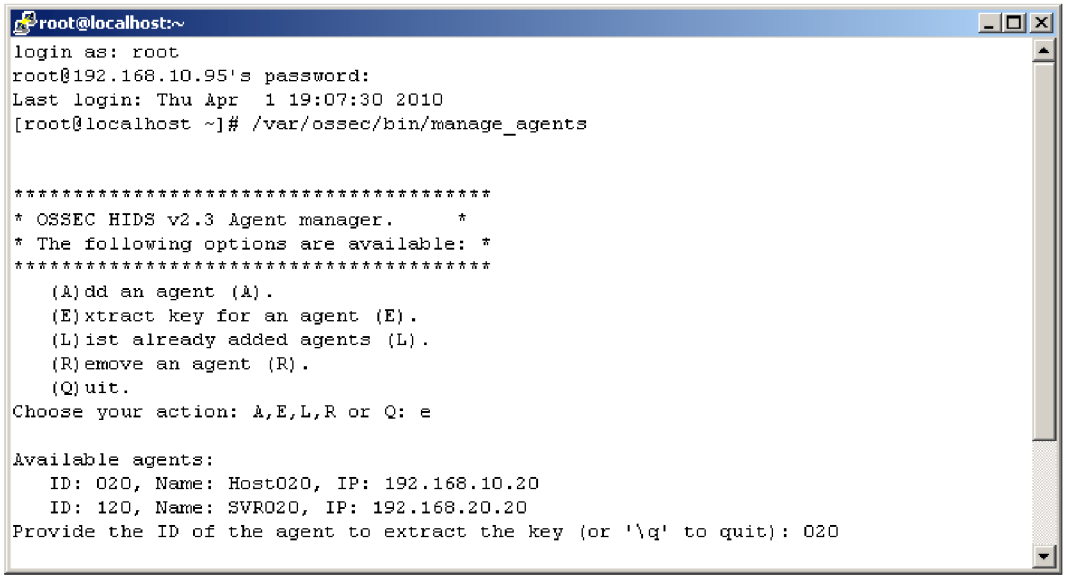
Prvním krokem instalace OSSEC agenta na platformě Windows je stažení instalátoru z domovských stránek [15] projektu. Následně spustíme instalaci OSSEC agenta pro platformu Windows. Přednastavené hodnoty programu ponecháme beze změny a instalaci dokončíme. Spuštěného agenta zachycuje obrázek 8.1

Obr. 8.1: Spuštěný OSSEC agent pro Windows.

Spojení OSSEC agenta se serverem je vázáno zadáním internetové adresy a autentizačního klíče. Jedním z možných postupů pro jeho přenos je využití SSH spojení, realizované klientem pro Windows, jakým může být například program Putty. Postup přihlášení na OSSEC server jako uživatel root zachycují obrázky 8.2 a 8.3



Obr. 8.2: Spuštšní SSH klineta Putty.



Obr. 8.3: Přihlášení na OSSEC server.

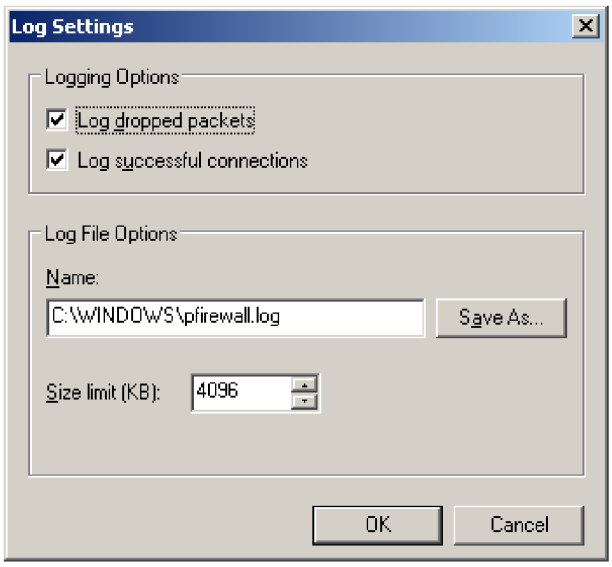
Na OSSEC serveru má každý jednotlivý agent uloženy záznamy, obsahující identifikační číslo, jméno a autentifikační klíč, který je třeba extrahovat. Operaci je možné provézt pomocí správce agentů.

```
/var/ossec/bin/manager_agents
```

V interaktivním menu, zachyceném na obrázku 8.3, zvolíme možnost `[E]` pro extrahování klíče. Po zadání identifikátoru dojde k výpisu klíče na obrazovku, odkud ho zkopírujeme do spuštěného OSSEC agenta. Stiskem tlačítka `save` provedeme uložení zadaných údajů.

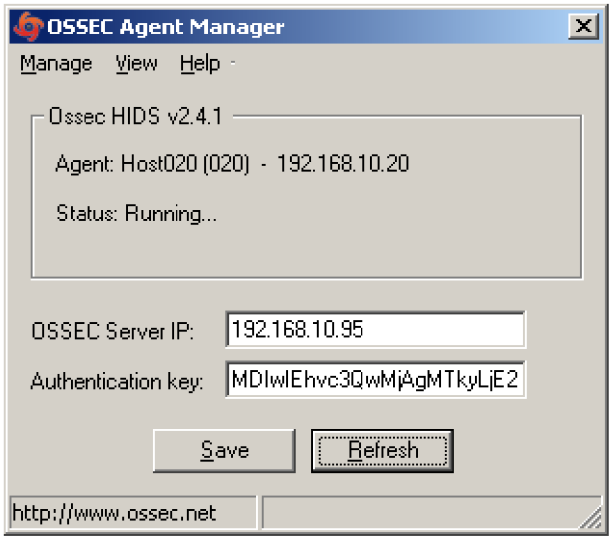
Nyní nastavíme standardní firewall ve Windows, aby do souboru zaznamenával zahozené pakety, jak zachycuje obrázek 8.4. Dále upravíme konfigurační soubor OSSEC agenta `ossec.conf` tak, aby tyto záznamy procházel a zjišťoval kompromitující události. Nastavení provedeme přidáním následujícího kódu na konec souboru.

```
<ossec_config>
  <localfile>
     <location>C:\Windows\pfirewall.log</location>
     <log_format>syslog</log_format>
  </localfile>
</ossec_config>
```



Obr. 8.4: Zaznamenávání zahozených paktů do souboru.

Příkazem `Manage/Start` OSSEC provedeme spuštění agenta. Nakonfigurovaný a plně funkční je zachycen na obrázku 8.5.



Obr. 8.5: Nastavený OSSEC agent pro Windows.

8.2.3 Agent pro Linux

Instalace OSSEC agenta na platformě Linux má obdobný průběh, jako instalace serveru. Do systému je potřeba doplnit nástroje pro kompilaci.

```
yum install gcc gcc-c++ autoconf make
```

Provedeme stažení instalačních souborů do adresáře `/usr/src`.

```
cd /usr/src
wget www.ossec.net/files/ossec-hids-2.4.1.tar.gz
```

Rozbalíme zdrojové soubory z archivu a spustíme instalátor.

```
tar zxfv ossec-hids-2.4.1.tar.gz
cd ossec-hids-2.4.1
./install.sh
```

Vlastní instalace probíhá v příkazové řádce formou dotazování na jednotlivé parametry. V prvních krocích dochází k volbě jazyku a cílové složky pro instalaci souborů. Český jazyk k dispozici není, ponecháme tedy standardní volbu anglického jazyka `[en]`. Ze zdrojových kódů je možné instalovat tři typy instancí. Volíme instanci `[agent]`. Cílovou složku pro instalaci ponecháme `[/var/ossec]`. Dále budeme vyzváni k zadání IP adresy OSSEC serveru.

```
192.168.20.95
```

Kontrolovány budou standardní výpisy systému, rozšířené o záznamy generované webovým serverem. Přidání dalších položek ke kontrole je možné editací konfiguračního souboru `ossec.conf`, který je umístěn v adresáři `/var/ossec/etc/`.

```
/var/log/messages
/var/log/secure
/var/log/maillog
/var/log/httpd/error_log
/var/log/httpd/access_log
/var/log/proftpd/auth.log
```

Nezbytnou součástí spuštění agenta je importování autentifikačního klíče z OSSEC serveru. Prvními kroky jsou přihlášení se server pomocí SSH jako root a spuštění manažera, který má na starosti účty jednotlivých agentů.

```
ssh root@192.168.20.95
/var/ossec/bin/manage_agents
```

Zadáme možnost [E] pro extrahování klíče, následovanou zadáním ID agenta 120. Provedeme kopii zobrazeného klíče a odhlášení z SSH.

```
****************************************
* OSSEC HIDS v2.4.1 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
```

Na místní stanici spustíme manažera agentů.

```
/var/ossec/bin/manage_agents


****************************************
* OSSEC HIDS v2.4.1 Agent manager.     *
* The following options are available: *
****************************************
   (I)mport key from the server (I).
   (Q)uit.
```

Potvrdíme jedinou možnost [I] pro importování klíče. Dříve zkopírovaný klíč vložíme. Zobrazí se informace o nově přidávaném agentovi.

```
Agent information:
   ID:120
   Name:SVR020
   IP Address:192.168.20.20

Confirm adding it?(y/n): y
Added.
```

Závěrem instalace provedeme spuštění OSSEC agenta.

```
/etc/init.d/ossec start
```

8.3 Servery

V rámci navržené sítě bude provozováno několik serverů, poskytujících služby uživatelům přistupujícím z vnitřní i vnější sítě. Výjimkou jsou služby elektronické pošty, které jsou dostupné pouze v rámci organizace. Popsány budou jednotlivé kroky, vedoucí k úspěšné instalaci a nastavení WWW, SFTP a MAIL serveru. Všechna řešení budou založena na platformě Linux.

8.3.1 WWW server

Pro interní i externí uživatele, přistupující k navržené síti bude provedena implementace programových prostředků, které umožní přístup na firemní stránky. Jako nejvhodnější kandidát byl zvolen open source software Apache s podporou šifrovaného spojení. Instalaci potřebných komponent i serveru provedeme pomocí systémového správce balíčků.

```
yum install mod_ssl openssl http
```

K realizaci šifrovaného spojení mezi klientem a serverem použijeme vlastní certifikát podepsaný soukromým klíčem. Nejprve provedeme vygenerování klíče délky 1024 bitů.

```
openssl genrsa -out ca.key 1024
```

Dalším krokem je vytvoření firemního certifikátu pro webový server. Při jeho generování budeme dotazováni na údaje o firmě, jakými jsou například název organizace nebo e-mailová adresa. Následuje příkaz pro vytvoření certifikátu společně s vyplněnými údaji.

```
openssl req -new -key ca.key -out ca.csr

Country Name [GB]:CZ
State or Province Name [Berkshire]:Czech
Locality Name [Newbury]:Brno
Organization Name [My Company Ltd]:VUTBR
Organizational Unit Name []:UTKO
Common Name []:WWW Email Address []:admin@test.net
```

Následuje podepsání certifikátu vygenerovaným soukromým klíčem.

```
openssl x509 -req -days 365 -in ca.csr -signkey ca.key \
-out ca.crt
```

Vytvořený soukromý klíč a podepsaný certifikát nakopírujeme do určených složek.

```
cp ca.crt /etc/pki/tls/certs
cp ca.key /etc/pki/tls/private/ca.key
```

Dále provedeme editaci konfiguračního souboru zabezpečeného spojení, kde odkomentujeme následující řádky a upravíme cesty k certifikátu a soukromému klíči.

```
vi +/SSLCertificateFile /etc/httpd/conf.d/ssl.conf
SSLCertificateFile /etc/pki/tls/certs/ca.crt
SSLCertificateKeyFile /etc/pki/tls/private/ca.key
```

Následuje přidání následujících řádků do souboru s nastaveními Apache serveru, který je umístěn v adresáři `/etc/httpd/conf/httpd.conf`.

```
NameVirtualHost *:443
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/ca.crt
    SSLCertificateKeyFile /etc/pki/tls/private/ca.key
    <Directory /var/www/html>
    AllowOverride All
    </Directory>
    DocumentRoot /var/www/html
    ServerName www.test.net
 </VirtualHost>
```

Posledním krokem je spuštění webového serveru.

```
/etc/init.d/httpd start
```

8.3.2 SFTP server

Uživatelům firemní sítě je umožněno mezi sebou sdílet důležité pracovní dokumenty, které mohou obsahovat citlivé údaje. K tomuto úložišti mají možnost přistupovat z vnitřní sítě, ale také z internetu. Nasazení FTP služby není vhodnou volbou, protože přihlašovací údaje i vlastní data jsou přenášena v nešifrované podobě. Výhodnější je implementace Secure FTP, které veškeré informace přenáší šifrovaně.

Před vlastní instalací serveru je nutné doplnění systému o potřebné komponenty, jako například openssl nebo nástroje pro správu verzí cvs.

```
yum install openssl openssl-devel cvs mod_ssl
```

Operační systém obsahuje požadované nástroje a je možné přistoupit ke stažení zdrojových souborů ProFTPD serveru [16] do složky `/usr/src`.

```
cd /usr/src
cvs -z3 -d:pserver:anonymous@proftp.cvs.sourceforge.net:\
/cvsroot/proftp co -P proftpd
cd proftpd
```

Následuje spuštění konfiguračního skriptu s požadovanými parametry, jako například `sysconfdir`, určující umístění souboru s nastavením, povolení nástroje `openssl` nebo direktiva `with-modules`, pomocí které se připojují další moduly.

```
./configure --prefix=/usr --sysconfdir=/etc/proftpd \
--localstatedir=/var/run --enable-openssl \
--with-modules=mod_sftp
```

Vlastní kompilaci a instalaci ProFTPD serveru s podporou zabezpečeného přenosu provedeme.

```
make
make install
```

Server je nyní nainstalován. Následovat bude několik bodů, vedoucích k jeho správnému nastavení. Mezi hlavní požadavku lze zařadit podporu šifrované komunikace, přihlašování pouze uživatelů definovaných v externím souboru. Po přihlášení mají uživatelé možnost pouze stahovat dostupné soubory. Výjimku tvoří adresář `upload`, kam je umožněn zápis.

Prvním krokem konfigurace je vytvoření cílových složek s požadovaným oprávněním.

```
mkdir /usr/ftp
mkdir /usr/ftp/upload
mkdir /var/log/proftpd
chmod 777 /usr/ftp/upload/
```

Byla zvolena autentizace uživatelů, probíhající na základě předem definovaného souboru. Tento způsob umožňuje provedení požadovaných operací bez nutnosti vytváření domovského adresáře přímo v systému. Pomocí skriptu `ftpasswd` vytvoříme soubor obsahující přihlašovací jména, identifikační číslo, standardní domovský adresář a také neplatný příkazový interpret. Ten zamezuje případu, kdy se uživatel snaží zaměnit standardní příkazový pro práci se soubory za systémový. Provedeme přidání nového uživatele `sftp`, kterému nastavíme domovský adresář `/usr/ftp` a identifikátor `5001`.[4]

```
./ftpasswd --passwd --name sftp --home /usr/ftp\
--shell=/dev/null --uid=5001
```

Pokud autentizační soubor obsahuje všechny požadované uživatele, provedeme jeho přesun do adresáře s nastavením.

```
mv ftpd.passwd /etc/proftpd
```

Důležitým krokem při konfiguraci serveru je editace souboru `proftpd.conf`, který je umístěn ve složce `/etc/proftpd`. Obsahuje všechna důležitá nastavení. Jedním z nich je i komunikační port, na kterém ProFTPD démon naslouchá. Protože port 22 je již obsazen služnou SSH, nastavíme server tak, aby naslouchal na portu 2222 a akceptoval uživatele definované v autentizačním souboru.

```
Port         2222
AuthUserFile /etc/proftpd/ftpd.passwd
```

Dále provedeme povolení modulu pro zabezpečení a zadání umístění souboru se záznamy.[19]

```
<IfModule mod_sftp.c>
   SFTPEngine on
   SFTPLog /var/log/proftpd/sftp.log

   SFTPCompression delayed
 </IfModule>
```

Přihlášeným uživatelům nastavíme výchozí domácí adresář `/usr/ftp`, který je uzamčen pouze ke čtení. Pro odesílání souborů na server je zde umístěna složka `upload`, do které je umožňen pouze zápis.

```
<Global>
   DefaultRoot /usr/ftp
   AllowOverwrite yes
   <Limit SITE_CHMOD>
      DenyAll
   </Limit>
  <Directory> /usr/ftp/upload/>
     Umask 012
     AllowOverwrite yes

    <Limit READ RMD DELE>
       DenyAll
    </Limit>

    <Limit STOR CWD MKD>
       AllowAll
    </Limit>
  </Directory>
</Global>
```

Posledním krokem je spuštění serveru.

```
/etc/init.d/proftpd start
```

8.3.3 Mail server

V rámci navržené sítě provede instalaci poštovního serveru, sloužícího k odesílání a příjem pošty. Využit bude především programem OSSEC pro zasílání varovných zpráv.

Ze systému odstraníme standardní nástroj pro odesílání pošty sendmail. Pomocí správce balíčku provedeme instalaci IMAP serveru dovecot a SMTP serveru postfix.

```
yum remove sendmail
yum install dovecot postfix
```

Nyní provedeme editaci hlavního konfiguračního souboru SMTP serveru `main.cf`, umístěného v adresáři `/etc/postfix` tak, aby obsahoval následující nastavení.[17]

```
myhostname = mail.test.net
mydomain = test.net
myorigin = $mydomain
inet_interfaces = all
mydestination = \$myhostname, localhost, $mydomain
mynetworks = 192.168.20.0/24 , 127.0.0.0/8
home_mailbox = posta/
relayhost = $mydomain
relay_domains =
```

V konfiguračním souboru IMAP serveru `dovecot.conf` , umístěném v adresáři `/etc`, upravíme následující dva řádky.

```
protocols = imap
mail_location = maildir:~/posta
```

Nezbytnou součástí je také vytvoření uživatelů, kterým bude umožněno elektronickou poštu používat.

```
useradd admin
mkdir /home/admin/posta
chmod -R 777 /home/admin/posta
passwd admin
```

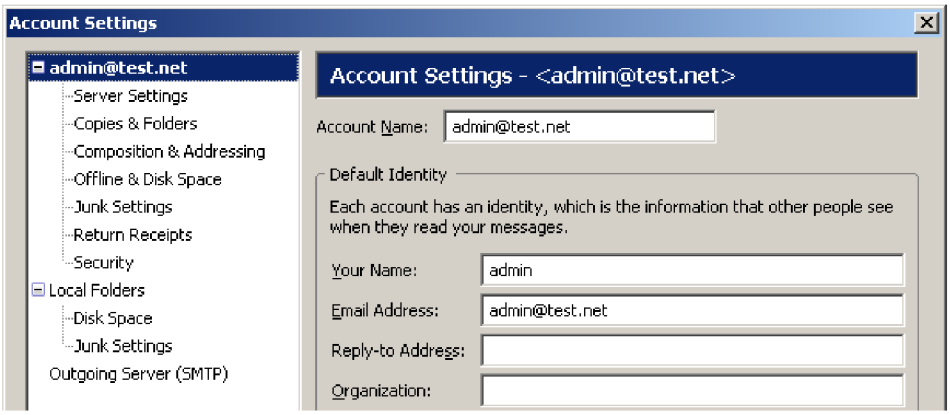
Závěrem provedeme spuštění obou služeb.

```
/etc/init.d/postfix start
/etc/init.d/dovecot start
```

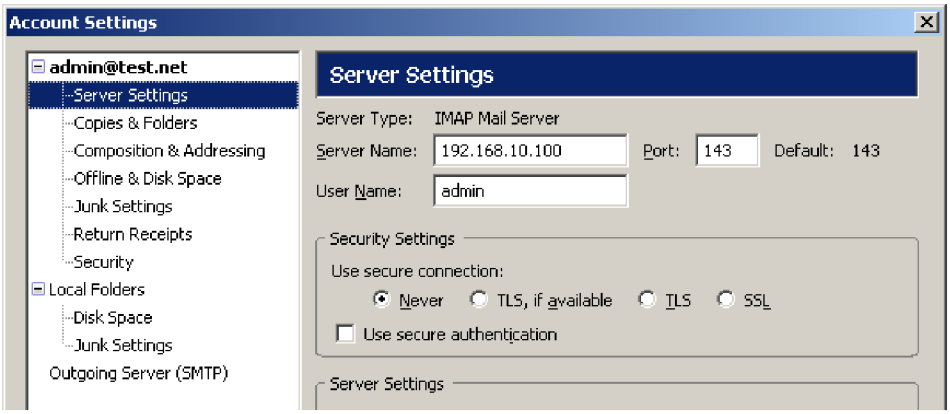
8.4 Nastavení email klienta

Klient pro přijímání elektronické pošty, provozovaný na platformě Windows, umožňuje především snadné zobrazeních varovných zpráv, zaslaných OSSEC serverem do poštovní schránky administrátora. Díky své ceně a bezproblémové funkčnosti byl vybrán neplacený program Thunderbird.

Během nastavováni je třeba zadat e-mailovou adresu 8.6 a IP adresu serveru pro příjem pošty 8.7.

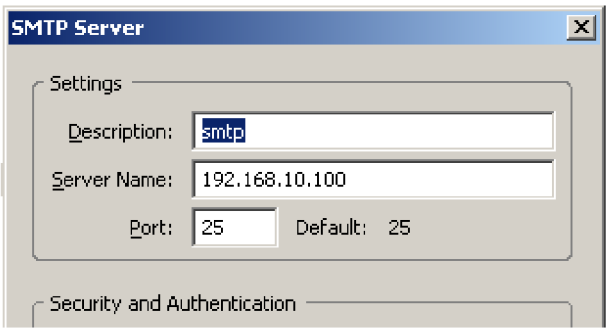


Obr. 8.6: Zadání e-mailové adresy.



Obr. 8.7: Zadání IP adresy a uživatelského jména.

Závěrečným krokem je zadání adresy SMTP serveru,sloužící pro odesílání pošty 8.8.



Obr. 8.8: Zadání IP adresy SMTP serveru.

9 OVĚŘENÍ

Posledním krokem je otestování navržené firemní sítě. Provedeno a popsáno bude několik typů útoků, vedených proti NIPS Snort i HIDS OSSEC. Dále budou popsány také významy jednotlivý výstrah včetně ukázek blokování závadného spojení.

9.1 OSSEC

Ověření funkčnosti IDS OSSEC je rozděleno do dvou částí. Kritériem pro členění byl typ operačního systému provozovaného agenta.

9.1.1 Agent Linux

Změna přístupových práv

OSSEC agent pro Linux, jehož instalace je popsána v kapitole 8.2.3, ve standardním nastavení kontroluje změnu práv pro přístup k souborů v adresáři `/etc/`. Umožníme li například editaci konfiguračního souboru serveru postfix všem uživatelům pomocí následujícího příkazu.

```
chmod 777 /etc/postfix/main.cf
```

Spuštěný agent tuto událost zaznamená a odešle serveru zprávu. OSSEC server následně odešle pomocí elektronické pošty varováni, informující o proběhlé události. Význam jednotlivých položek zprávy je popsán v kapitole 5.3.

```
OSSEC HIDS Notification.
2010 Mar 30 18:34:16

Received From: (SVR020) 192.168.20.20->rootcheck
Rule: 510 fired (level 7) -> "Host-based anomaly \
detection event (rootcheck)."
Portion of the log(s):

File '/etc/postfix/main.cf' is owned by root and has \
written permissions to anyone.

END OF NOTIFICATION
```

Kontrola přihlášení

Agent provádí kontrolu nejen standardních výpisů operačního systémů, ale také výpisy instalovaných programů. Jejich seznam je uveden v kapitole 8.2.3. Provádí tedy také kontrolu neúspěšných pokusů o přihlašování na SFTP server. Opakované zadávání hesla je možno realizovat například pomocí programu medusa [3]. V případě většího množství neúspěšného přihlášení je vydána výstraha.

```
OSSEC HIDS Notification.
2010 May 11 23:51:19

Received From: (SVR020) 192.168.20.20->/var/log/secure
Rule: 11251 fired (level 10) -> "FTP brute force (multiple \
failed logins)."
Portion of the log(s):

May 11 23:51:16 localhost proftpd[20091]: 127.0.0.1 \
(192.168.10.20) - USER sftp (Login failed): Incorrect password
May 11 23:51:16 localhost proftpd[20090]: 127.0.0.1 \
(192.168.10.20) - USER sftp (Login failed): Incorrect password
May 11 23:49:55 localhost proftpd[20088]: 127.0.0.1 \
(192.168.10.20) - USER sftp (Login failed): Incorrect password
May 11 23:49:55 localhost proftpd[20089]: 127.0.0.1 \
(192.168.10.20) - USER sftp (Login failed): Incorrect password

                                    ⋮

May 11 23:49:55 localhost proftpd[20084]: 127.0.0.1 \
(192.168.10.20) - USER sftp (Login failed): Incorrect password

 --END OF NOTIFICATION
```

9.1.2 Agent Windows

Kontrola integrity

Agent pro Windows provádí obdobné kontroly, jako u OS Linux. Jsou však danému operačnímu systému přizpůsobeny. Součástí základní konfigurace je kontrola integrity důležitých souborů. Pokud například provedeme změnu parametrů zavaděče[1] systému, OSSEC na tuto událost upozorní.

[1] Lze realizovat editací souboru boot.ini

```
OSSEC HIDS Notification.
2010 May 14 17:08:11

Received From: (user020) 192.168.10.20->syscheck
Rule: 550 fired (level 7) -> "Integrity checksum changed."
Portion of the log(s):

Integrity checksum changed for: 'C:\boot.ini'
Size changed from '321' to '319'
Old md5sum was: '31b6c529cc4a2b903d12cc99d8d0d08a'
New md5sum is : '505d836146c773eb1ebd5ac3de14a4fa'
Old sha1sum was: '4645b35a7454ab0dd8a9ab6fe6ede1eb78cd8a50'
New sha1sum is : '7943d9b07bf81757daf1a531e27f5ccd9eb7b09f'

--END OF NOTIFICATION
```

Kontrola sítě

Rozšířením kontroly o soubor pfirewall.log, popsané v kapitole 8.2.2, umožňuje agent zasílat varování o zahozených paketech Windows firewallem.

```
OSSEC HIDS Notification.
2010 May 13 20:58:20

Received From: (user020) 192.168.10.20->\Windows\pfirewall.log
Rule: 4151 fired (level 10) -> "Multiple Firewall drop events \
from same source."
Portion of the log(s):

2010-05-13 20:58:33 DROP TCP 192.168.10.100 192.168.10.20 63111 \
25 40 FUP 33165243 0 4096 - - - RECEIVE
2010-05-13 20:58:33 DROP TCP 192.168.10.100 192.168.10.20 63111 \
80 40 FUP 33165243 0 1024 - - - RECEIVE
2010-05-13 20:58:33 DROP TCP 192.168.10.100 192.168.10.20 63111 \
                                    ⋮
2010-05-13 20:58:32 DROP TCP 192.168.10.100 192.168.10.20 63111 \
113 40 FUP 33165243 0 2048 - - - RECEIVE

 --END OF NOTIFICATION
```

9.2 Snort

Závěrečnou částí je ověření nasazeného systému prevence průniku Snort. Jeho schopnost detekovat nechtěné aktivity je velku měrou závislá na použitých pravidlech. Lze je rozčlenit do několika skupin. Hlavními kritérii jsou typy kontrolovaných protokolů a útoků.

- Typy protokolů

  HTTP, FTP, IMAP, POP3, TELNET, TFTP, SQL, ICMP

- Druhy útoků

  DDoS
  Zjišťování použitých protokolů
  Zjišťování přístupných služeb
  Zneužití zranitelnosti software

Jednotlivá pravidla mají sice obecnou platnost, avšak v některých případech je nutné jejich uzpůsobení pro konkrétní implementaci. Oficiální pravidla, jichž instalace je popsána v kapitole 8.1.4, nemusí obsahovat všechny vyžadované kontroly. Bude tedy nutné vytvoření vlastních pravidel.

Služby provozované v rámci navržené sítě, lze pomocí oficiálních pravidel ochránit především před útoky směřovanými na služby webového serveru. Záměrem útočníka přitom často bývá vzdálené spuštění škodlivého kódu a následné převzetí kontroly nad počítačem.

9.2.1 WWW útoky

Útočníky je často vyhledáván způsob, jak na vzdáleném počítači způsobit přetečení bufferu a následně spustit vlastní škodlivý kód. Zapotřebí je však dostatečná znalost assembleru, architektury procesorů a vyšších programovacích jazyků, především jazyka C.

Pomocí velkého množství NOOP instrukcí procesoru je za jistých okolností možné zaplnit přidělené paměťové místo. Pomocí nového ukazatele je přidělen další prostor, který obsahuje útočníkem vytvořený kód. Jeho spuštěním se otevírá cesta, jak nad vzdáleným systém převzít kontrolu.

Výhodné je těmto útokům zabránit již v první fázi, tedy při snaze zaplnit paměť znaky pomocí NOOP instrukcí. Možným způsobem obrany je detekce velkého počtu znaků v URL požadavku, které toto přetečení mohou způsobit.[25]

Příkladem takové aktivity je zaslání následující žádosti webovému serveru.

```
http://192.168.30.100/cgi-bin/helloworld?type=AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA
```

IPS Snortem vygeneruje výstrahu o události a závadné pakety zahodí. Významy jednotlivých částí vydané výstrahy jsou popsány v kapitole 4.4.

```
[**] [1:1394:10] DROPPED SHELLCODE x86 inc ecx NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
05/05-20:23:56.778829 192.168.30.5:50800 -> 192.168.20.20:80
TCP TTL:63 TOS:0x0 ID:25021 IpLen:20 DgmLen:836 DF
***AP*** Seq: 0x20BDF0D2  Ack: 0x884C526  Win: 0xB7  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2396522 7741058
```

Dalším možným způsobem útoku je zadání požadavku, který není normalizovaný. Jeho snahou je dostat se mimo paměťové místo, přidělené pro URI a spuštění kódu pomocí příkazového interpretu.

```
http://192.168.30.100/scripts/..%c0%af../winnt/system32\
/cmd.exe?/c+ver
```

Systém na základě pravidla vyhodnotí paket jako závadný. Provede jeho odstranění a vygeneruje zprávu o události.

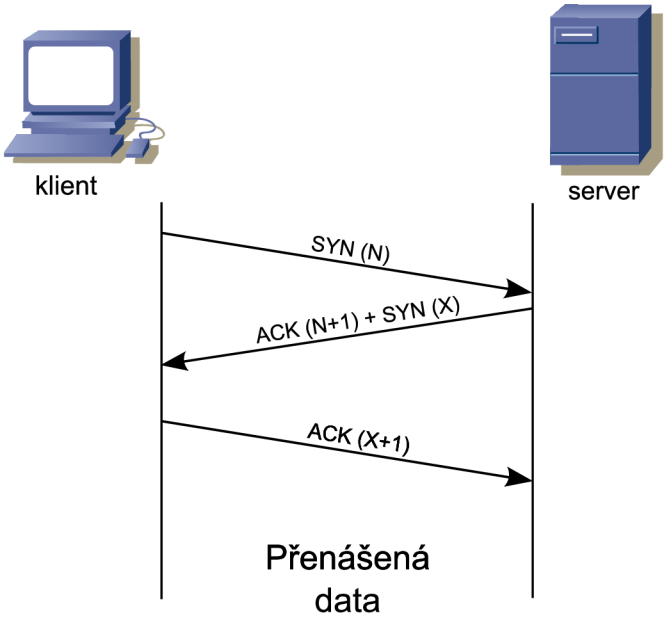
```
[**] [1:1002:10] DROPPED WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
05/05-20:25:22.422722 192.168.30.5:50802 -> 192.168.20.20:80
TCP TTL:63 TOS:0x0 ID:3161 IpLen:20 DgmLen:468 DF
***AP*** Seq: 0x70CAF991  Ack: 0xD27F727  Win: 0xB7  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2422215 7791491
```

9.2.2 Skenování

Skenování je postup, kterým se snažíme zjistit informace o poskytovaných službách nebo podporovaných protokolech. Většinou bývá prvním krokem při hledání slabin vzdáleného zařízení. Skenování různého zaměření provádějí nejen útočníci, ale také správci, kteří se snaží udržet obranyschopnost systému na co nejvyšší úrovni.

SYN skenování

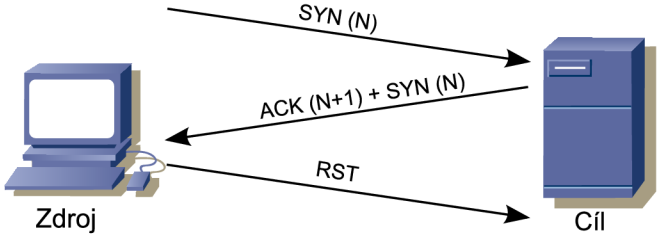
Navazování nového spojení probíhá podle diagramu třícestného sestavování TCP spojení[2], zachyceného na obrázku 9.1



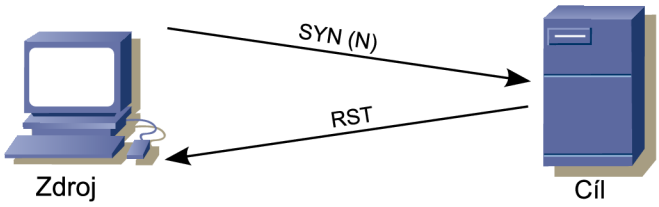
Obr. 9.1: Sestavení nového TCP spojení.

Skenování s příznakem SYN má shodné počáteční rysy, jako navazování normálního spojení. Útočník odešle žádost s příznakem SYN, kterým se snaží inicializovat spojení. Pokud Server na daném portu naslouchá, odpoví zprávou s nastavenými příznaky SYN+ACK. V tomto okamžiku však útočník provede ukončení spojení. Tento sled událostí je zachycen na obrázku 9.2. Pokud je však daný port nepřístupný, server odpovídá ukončením spojení pomocí paketu s nastaveným příznakem RST, zachycený na obrázku 9.3. Uvedený způsob je vhodný pro zjišťování běhu vzdálených služeb, protože nebývá často systémy zaznamenáván.

[2]Tree-Way Handshake



Obr. 9.2: Skenování otevřeného portu.



Obr. 9.3: Skenování uzavřeného portu.

Oficiální pravidla, jejichž instalace je popsána v kapitole 8.1.4, detekování tohoto typu skenování neumožňují. Přidáme proto následující pravidlo.

```
drop tcp $EXTERNAL_NET any -> any any \
(msg:"SCAN SYN"; fragbits: !D; dsize: 0; flow:stateless; \
flags:S,12;detection_filter: track by_src ,count 10, \
seconds 60; classtype:network-scan; sid:1000002; rev:2;)
```

Vlastní skenování provedeme pomocí programu NMAP [12], kterému zadáme IP adresu cíle, rozsah testovaných portů a nastavíme příznak SYN, nachází se v hlavičce TCP datagramu.

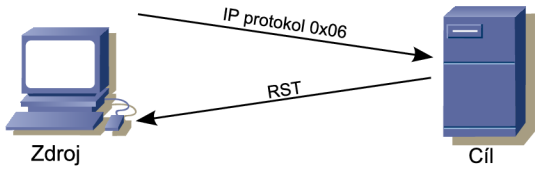
```
nmap -sS 192.168.30.100 -p 1-3333
```

Systém Snort na komunikaci zareaguje vydáním výstrahy a následným blokováním všech žádostí, přicházejících ze zdrojové IP adresy.

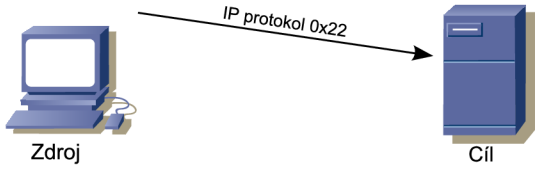
```
[**] [1:1000002:2] SCAN SYN [**]
[Classification: Detection of a Network Scan] [Priority: 3]
05/12-23:18:46.901831 192.168.30.5:42401 -> 192.168.30.100:1979
TCP TTL:59 TOS:0x0 ID:2028 IpLen:20 DgmLen:44
******S* Seq: 0x5DC53C2  Ack: 0x0  Win: 0x1000  TcpLen: 24
TCP Options (1) => MSS: 1460
```

Skenování protokolů

Pomocí tohoto skenování je možné zjistit, jaké protokoly jsou na stanici nasazeny. Pokud je cílovým zařízením router, lze detekovat například směrovací protokoly, jako například EGP nebo IGP. Pokud je protokol provozován, je nazpět zaslán paket s příznakem RST. V opačném případě není zaslána žádná odpověď. Tuto situaci zachycují obrázky 9.5 a 9.4.



Obr. 9.4: Existence testovaného protokolu.



Obr. 9.5: Absence testovaného protokolu.

Pro odhalení útoku je nutné nasazení vlastního pravidla.

```
drop ip $EXTERNAL_NET any -> any any \
(msg:"Protocol Scan";  dsize: 0; flow:stateless; \
ip_proto: !icmp; ip_proto: !udp; \
detection_filter: track by_src ,count 1, seconds 360; \
classtype:network-scan; sid:1000003; rev:2;)
```

Skenování použitých protokolů lze realizovat pomocí programu NMAP [12].

```
nmap -s0 192.168.30.100
```

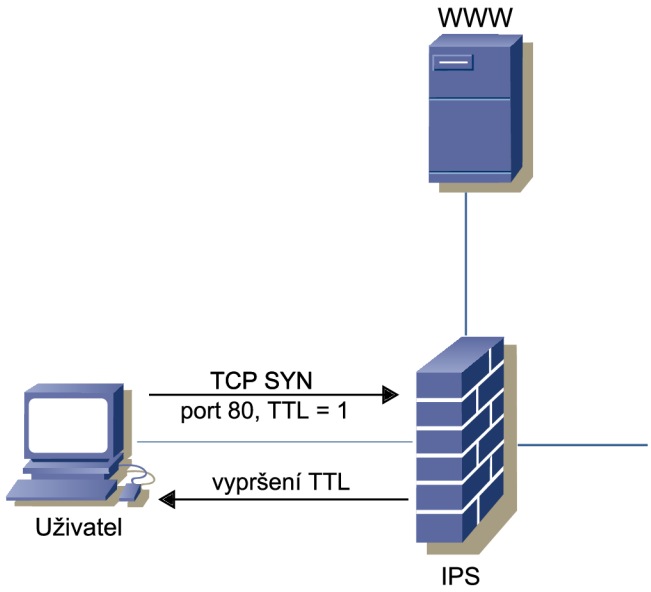
Systém Snort na základě definovaného pravidla pakety zahazuje a vydává následující výstrahu.

```
[**] [1:1000003:2] Protocol Scan [**]
[Classification: Detection of a Network Scan] [Priority: 3]
05/13-01:09:53.345390 192.168.30.5 -> 192.168.30.100
PUP TTL:48 TOS:0x0 ID:59323 IpLen:20 DgmLen:20
```

TTL skenování

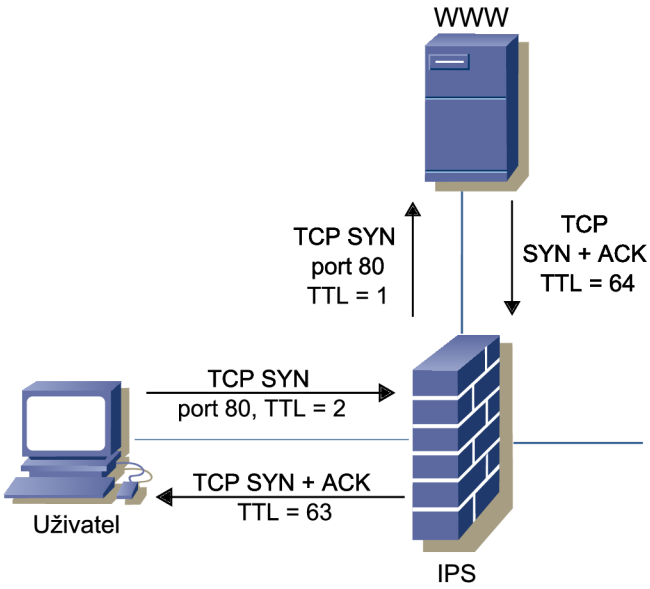
Každý IP paket, putující sítí ve své hlavičce obsahuje hodnotu TTL, určující dobu jeho života. Ta je při průchodu každým prvkem[3] sítě dekrementována. Pomocí postupného snižování hodnoty TTL je zamezeno situace, kdy v síti dochází k nekonečným smyčkám zbloudilých paketů. Příchozí TCP zprávy s krátkou dobou životnosti mohou signalizovat problémy na přenosové cestě mezi zdrojem a cílem. Aspektů, způsobujících komplikace lze nalézt několik. Příkladem mohou být zahlcení kapacity linky, chybu ve směrovacích tabulkách nebo záměrné odeslání paktu s krátkou dobou životnosti.

Za určitých okolností je pomocí TCP zpráv s nízkou hodnotou TTL možné zjišťovat vzdálenost od hraničního směrovače k prvkům demilitarizované zóny. Základním principem je postupné zvyšování hodnoty TTL. Počáteční dva kroky zachycují obrázky 9.6 a 9.7. Poslední částí odeslání TCP paket s příznakem RST, kterým útočník spojení ukončí.



Obr. 9.6: První fáze zjišťování vzdálenosti služby.

[3] Pracující na třetí nebo vyšší vrstvě modelu ISO OSI.



Obr. 9.7: Druhá fáze zjišťování vzdálenosti služby.

Vytvoříme proto pravidlo, kterým definujeme zahazování paketů s krátkou dobou životnosti.

```
drop tcp $EXTERNAL_NET any -> any any\
(msg:"TCP Packet with low TTL"; ttl: < 5;\
detection_filter: track by_src ,count 1, seconds 60;\
classtype:network-scan; sid:1000006; rev:1;)
```

Odeslání paketu s krátkou dobou životnosti je možné pomocí programu NMAP [12].

```
nmap --ttl 1 192.168.30.100
```

IPS Snort příchozí zprávu vyhodnotí. Provede zahození paketu a vygeneruje výstrahu.

```
[**] [1:1000006:1] Droped TCP Packet with low TTL [**]
[Classification: Detection of a Network Scan] [Priority: 3]
05/05-23:57:46.358699 192.168.30.5:60464 -> 192.168.30.100:9000
TCP TTL:1 TOS:0x0 ID:20714 IpLen:20 DgmLen:44
******S* Seq: 0xFF4ED176  Ack: 0x0  Win: 0x800  TcpLen: 24
```

9.2.3 ICMP zahlcení

Aby bylo možné zjistit dostupnost serverů, poskytujících služby, bývá zpravidla povolen ICMP protokol. Často však bývá zneužíván při DoS či DDoS útocích. Útočník vytvoří záplavu ICMP paketů o nadstandardní velikosti, pomocí které zahltí kapacitu připojení i hardwarové možnosti vzdáleného stroje.

Způsoby obrany před tímto lze rozdělit do několika kroků. Mezi první patří omezení počtu přijímaných ICMP zpráv za určitý čas přímo ve firewallu. Dalším krokem může být kontrola nestandardní velikosti těchto zpráv pomocí systémů, jako například Snort.

Vytvoříme vlastní pravidlo, které bude zahazovat ICMP zprávy větší než 1508 bytů.

```
drop icmp $EXTERNAL_NET  any -> any any \
(msg:"Large ICMP packet size"; dsize: >1508; \
classtype: icmp-event; detection_filter: track by_src, \
count 1, seconds 60;sid:1000004; rev:2;)
```

Pomocí programu hping [6] je možné vygenerovat ICMP zprávu požadované velikosti.

```
hping 192.168.30.100 -d 1600
```

IPS Snort zaznamená příchozí zprávu, jejíž velikost je větší, než je definováno pravidlem. Paket proto zahodí a vygeneruje výstrahu.

```
[**] [1:1000004:2] Large ICMP packet size [**]
[Classification: Generic ICMP event] [Priority: 3]
05/05-21:55:27.787833 192.168.30.5:2783 -> 192.168.30.100:0
ICMP TTL:64 TOS:0x0 ID:217 IpLen:20 DgmLen:1628
Type:8  Code:0  ID:18493   Seq:7168  ECHO
```

9.2.4 SSH útok

SSH je komunikační protokol, umožňující zabezpečené spojení mezi dvěma koncovými zařízeními. Zprostředkovává přístup k příkazovému interpretu druhé strany a přenos souborů. Je nástupcem nezabezpečeného protokolu telnet. Pomocí SSH je tedy možné vzdáleně spravovat různá zařízení, jako například servery, routery nebo uživatelské stanice. Zároveň však vzniká riziko zneužití. Proto je minimálně vhodné vytvářet záznamy o každém přístupu.

Jednou z možných cest, vedoucích k získání přístupu ke vzdálenému zařízení pomocí SSH, je útok hrubou silou. Při jeho provádění jsou testovány všechny možné kombinace hesel ze zvoleného rozsahu. Jednoznačnou výhodou je jistota nalezení hledaného výsledku. Je však časově nejnáročnější a tedy značně neefektivní metodou. Pokud je hledané heslo dostatečně „slabé“ a vyskytuje ze mezi běžnými slovy, lze použít takzvaný slovníkový útok. Testování správnosti hesla probíhá ve třech fázích.

- Připojení ke vzdálené straně – obrázek 9.1
- Zadání hesla – tři pokusy
- Odpojení vzdálenou stranou

Častému připojování a odpojování lze snadno zabránit přidáním pravidlo.

```
drop tcp any 2222 -> $EXTERNAL_NET any \
(msg:"DROPPED Potential SSH BruteForce Scan!"; \
flow:established,to_client; content:"SSH-"; content:"OpenSSH"; \
detection_filter:track by_dst, count 3, seconds 60; \
classtype:attempted-user; sid:10000012; rev:5;)
```

Testování provedeme pomocí programu medusa [3]. Cílem bude hraniční prvek IPS, který používá SSH na portu 2222.

```
medusa -h 192.168.30.100 -n 2222 -u root -P passwords -M ssh
```

Pravidlo Snortu povoluje tři neplatná připojení. Tedy zadání maximálně dvanácti neplatných hesel, než komunikaci pro konkrétní zdroj zablokuje a vystaví výstrahu.

```
[**] [1:10000012:5] DROPPED Potential SSH BruteForce Scan! [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
05/13-15:37:12.680791 192.168.30.100:2222 -> 192.168.30.5:35041
TCP TTL:64 TOS:0x0 ID:56078 IpLen:20 DgmLen:72 DF
***AP*** Seq: 0x9F4E3DB0  Ack: 0x1F612689  Win: 0x5B  TcpLen: 32
TCP Options (3) => NOP NOP TS: 58779332 25404317
```

9.2.5 Omezení přístupu

Na směrovacím prvku IPS, zachyceném na obrázku 6.1, je uživatelům místní sítě umožněn přistup na webové stránky, umístěné v Internetu. Často se však můžeme setkat se situací, kdy je třeba na některé z nich přístup omezit či úplně zakázat. Jednoduchou a lehce spravovatelnou možností se nabízí přidání pravidla do systému Snort.

Příkladem mohou být stránky sociální sítě `www.facebook.com`. Zakázání přístupu provedeme přidáním vlastního pravidla, které kontroluje TCP pakety, přicházející z vnější sítě, která je definovaná v kapitole 8.1.5. Zjišťuje, zda paket obsahuje jmennou adresu stránek sociální sítě. V případě shody je zahozen.

```
drop tcp $EXTERNAL_NET  any -> any any \
(msg:"DROPPED Connecting to www.facebook.com";\
content:"www.facebook.com"; nocase;\
classtype: web-application-activity;sid:1000007; rev:1;)
```

Snort vytvoří výstrahu a provede definovanou akci.

```
[**] [1:1000007:1] DROPPED Connecting to www.facebook.com [**]
[Classification: access to  web application] [Priority: 2]
05/12-17:03:16.600724 192.168.10.20:1101 -> 69.63.189.39:80
TCP TTL:127 TOS:0x0 ID:365 IpLen:20 DgmLen:546 DF
***AP*** Seq: 0xF6255B0C  Ack: 0x41376A9B  Win: 0xFFFF  TcpLen: 20
```

10 ZÁVĚR

Většina dnes nasazovaných bezpečnostních strategií obsahuje pouze základní ochranné prvky, jako jsou například firewally či antiviry. Jejich nasazení je účinné proti konkrétním typům síťových útoků či vybrané skupině škodlivého software. Nedokáží však pokrýt celé spektrum hrozeb, se kterým je možné se setkat. Komplexnější ochranu je možné získat nasazením dalších prvků ochrany.

Mezi takové nástroje patří i systémy prevence a detekce průniku. S výhodou jsou nasazovány také jejich kombinace. Realizovány mohou být jako samostatný hardware nebo nasazeny v podobě software. Primárním účelem je sledování aktivit v rámci systému nebo analýza síťové komunikace. Jednotlivé události systémy vyhodnocují na základě předem definovaných pravidel. Jejich součástí je také příslušná akce, která má být vykonána v případě nalezení nežádoucího obsahu či události. Mezi reakce systému lze zařadit zaslání výstrahy správci systému, jak tomu bývá u systémů detekce. Pokud je však vyžadováno neprodlené blokování, je nutné nasazení systému IPS.

V rámci navržené sítě byly nasazeny prvky detekce i prevence průniku. Užití obou typů nástrojů přináší vyšší míru poskytované bezpečnosti a komplexnosti. Výhod detekčního systému OSSEC je využito při nasazení na koncových stanicích a serverech, kde provádí monitorování širokého spektra událostí. Příkladem může být kontrola integrity souborů, neplatných pokusů o připojení ke vzdálené službě nebo nestandardních aktivit programů. Pomocí systému prevence průniku Snort, umístěném na hraniční směrovači, je možné detekovat závadnou komunikaci, směřující především směrem do vnitřní sítě. Ta by zde mohla kompromitovat uživatelské stanice či servery.

Nasazené bezpečnostní prvky byly otestovány vybranými typy útoků, zaměřených především na získání informací o vnitřní síti, provozovaných protokolech nebo kompromitaci poskytovaných služeb. Dosažené výsledky poukazují na důležitost implementace těchto typů bezpečnostních prvků a jejich začlenění do celkové ochranné strategie daného subjektu. Při nasazení robustních bezpečnostních systémů však vystávají různá úskalí. Jedním z hlavních je dostatečné množství kvalitních pravidel.

LITERATURA

[1] Daniel B. *Log Analysis using OSSEC* [online]. 2007, [cit. 2009-12-08], s. 46. Dostupný z WWW: <http://www.ossec.net/ossec-docs/auscert-2007-dcid.pdf>.

[2] Drum R. *IDS and IPS placement for network protection* [online]. 2006, [cit. 2009-12-08], s. 8. Dostupný z WWW: <www.infosecwriters.com/text_resources/pdf/IDS_Placement_RDrum.pdf>.

[3] *Foofus.Net* [online]. 2010 [cit. 2010-05-02]. Dostupný z WWW: <http://www.foofus.net/>.

[4] GITE, V. *ProFTD* [online]. 2009 [cit. 2010-04-20]. Dostupný z WWW: <http://www.cyberciti.biz/tips/linux-installing-configuring-proftpd-ftp-server.html>.

[5] Gracik, M. *Analýza systémových záznamov* Brno, 2008, bakalárska práca, FIT VUT v Brně [cit. 2010-05-04]. Dostupný z WWW: <http://www.fit.vutbr.cz/study/DP/rpfile.php?id=7248>.

[6] *Hping* [online]. 2010 [cit. 2010-04-02]. Dostupný z WWW: <http://www.hping.org/>.

[7] *Intrusion detection system* [online]. 2009 [cit. 2009-12-03]. Dostupný z WWW: <en.wikipedia.org/wiki/Intrusion_detection_system>.

[8] *Intrusion prevention system* [online]. 2009 [cit. 2009-12-07]. Dostupný z WWW: <en.wikipedia.org/wiki/Intrusion_prevention_system>.

[9] *LibDNET* [online]. 2010 [cit. 2010-04-21]. Dostupný z WWW: <http://libdnet.sourceforge.net/>.

[10] *LibNET* [online]. 2010 [cit. 2010-04-12]. Dostupný z WWW: <ftp://ftp.eenet.ee/pub/gentoo/distfiles/libnet-1.0.2a.tar.gz>.

[11] MESSER, J. *A Comprehensive Guide to nmap* [online]. 2010 [cit. 2010-05-02]. Dostupný z WWW: <http://www.networkuptime.com/nmap/>.

[12] *NMAP* [online]. 2010 [cit. 2010-04-21]. Dostupný z WWW: <http://nmap.org/>.

[13] NSS Group *Intrusion Prevention Systems (IPS)* [online]. 2004, [cit. 2009-12-08], s. 6. Dostupný z WWW: <hosteddocs.ittoolbox.com/BW013004.pdf>.

[14] NSS Group *SecureWorks iSensor 850 V5.3* [online]. 2006, [cit. 2009-12-08], s. 56. Dostupný z WWW: <nsslabs.com/grouptests/ips/edition3/pdf/IPSED3-0601-SW.pdf>.

[15] *OSSEC* [online]. 2009 [cit. 2009-12-09]. Dostupný z WWW: <www.ossec.net>.

[16] *ProFTPD* [online]. 2010 [cit. 2010-04-12]. Dostupný z WWW: <http://www.proftpd.org/>.

[17] PUST, R. *Cvičení předmětu MNSB* [online]. 2010-04-04 [cit. 2010-04-22]. Dostupný z WWW: <https://www.vutbr.cz/elearning/file.php/86959/cv10-2010-mail.pdf>.

[18] RIDEN, J. *Responding to a Brute Force SSH Attack* [online]. 2008-12-03 [cit. 2010-05-03]. Dostupný z WWW: <http://www.securityfocus.com/print/infocus/1903>.

[19] ROOTBSD *SFTP support in ProFTPD* [online]. 2009-04-12 [cit. 2010-04-21]. Dostupný z WWW: <http://www.directadmin.com/forum/showthread.php?t=30607>.

[20] Schultz, G.; Endorf, C.; Mellander, J. *Intrusion Detection and Prevention* [online]. 2005, [cit. 2009-12-08], s. 355. Dostupný z WWW: <http://books.google.cz/books?id=AWYduASeDlEC&lpg=PP1&pg=PP1 #v=onepage&q=&f=false>.

[21] *Snort* [online]. 2009 [cit. 2009-12-04]. Dostupný z WWW: <en.wikipedia.org/wiki/Snort_(software)>.

[22] *Snort* [online]. 2010 [cit. 2010-04-10]. Dostupný z WWW: <www.snort.org>.

[23] Sourcefire, Inc. *Snort User Manual* [online]. 2010 [cit. 2010-05-04]. Dostupný z WWW: <http://www.procyonlabs.com/snort_manual>.

[24] *TCPDUMP / LIBPCAP* [online]. 2010 [cit. 2010-05-04]. Dostupný z WWW: <http://www.tcpdump.org>.

[25] Haugsness, K. *Intrusion Detection FAQ* [online]. 2010 [cit. 2010-05-04]. Dostupný z WWW: <http://www.sans.org/security-resources/idfaq/polymorphic_shell.php>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

API	Application Programming Interface
APIDS	Application protocol – based IDS
BASE	Basic Analysis and Security Engine
BSD	Berkeley Software Distribution
CIDR	Classless Inter-Domain Routing
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone
DNS	Domain Name System
DoS	Denial of Service
EGP	Exterior Gateway Protocol
FTP	File Transfer Protocol
GIDS	Gateway IDS
GNU	GNU's not UNIX
GPL	General Public License
HIDS	Host – Based IDS
HIPS	Host – based IPS
HTTP	Hypertext Transfer Protocol
HTTPS	Zabezpečený HTTP
HW	Hardware
ICMP	Internet Control Message Protocol
IDS	Intrusion – Detection System
IGP	Interior gateway protocol
IMAP	Internet Message Access Protocol

IP	Internet Protocol
IPS	Intrusion – Prevention System
ISO	International Organization for Standardization
LAN	Local Area Network
LIDS	Log – based Intrusion Detection
MBR	Master boot record
NIDS	Network–Based IDS
NIPS	Network – Based IPS
NOOP	No Operation Performed
OS	Operating system
OSI	Open Systems Interconnection
OSSEC	Open Source Host – based Intrusion Detection System
PIDS	Protocol – based IDS
POP3	Post Office Protocol version 3
QoS	Quality of Service
SFTP	Secure FTP
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSH	Secure Shell
SW	Software
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TOS	Type of Service
TTL	Time To Live
UDP	User Datagram Protocol

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WAN	Wide Area Network
WWW	World Wide Web
XML	Extensible Markup Language