IDS/IPS

Klasifikace IDS

Podle chráněné vrstvy

- síťové (protokoly až do transportní vrstvy)
- aplikační (WEB, Databáze)

V poslední době dochází k prolínání a snaze analyzovat všechno, nebo nabídnout modulární řešení (F5).
Existují i specializovaná řešení (ARBOR).

Podle metod detekce

- IDS založené na pravidlech (signaturách)
- IDS založená na profilu (detekce anomálií)

Architektura

Senzor, agent,

řídící centrum (konzole).

- správa dat
- generování výstrahy
- korelace událostí
- monitorování ostatních komponent
- řídící konzole

Způsoby zapojení do sítě

- zrcadlení portu (mirroring, SPAN)
- větvení sítě (network TAP)

Způsoby tvorby pravidel

- negative security model (blacklist)
- positive security model (whitelist)

Positive security model zpravidla méně zatěžuje sondu a lépe chrání před dosud neznámými útoky.

Problémy

- falešné pozitivní detekce
- vysokorychlostní prostředí
- potíže s detekcí neznámých hrozeb

Příklad implementace (Snort)

Příklazová řádka

(http://manual.snort.org/node4.html)

Sniffer mode:
./snort -vde

Packet logger mode:
./snort -vde -l ./log

IDS mode:

./snort -vde -l ./log -c snort.conf

-A definuje způsob alertování (např -A console zasílá alerty typu “fast” na konzoli)

Konfigurační soubor

(http://manual.snort.org/node15.html)

Konfigurační soubor obsahuje 9 sekcí:

1) síť
2) dekodér
3) detekční modul
4) dynamické knihovny
5) preprocesory
6) výstupy
7) pravidla
8) pravidla pro preprocesor a dekodér
9) pravidla pro sdílené objekty

1)
var HOME_NET
var EXTERNAL_NET
var _SERVERS $HOME_NET

var RULE_PATH
var PREPROC_RULE_PATH
var SNORT_PATH

2)

config logdir

3)

dynamicpreprocessor directory
dynamicengine

6)

output unified2: filename merged.log, limit 128

include classification.config
include reference.config

7)

include $RULE_PATH/local.rules

Pravidla

(http://manual.snort.org/node28.html)

Struktura pravidla:
akce protokol zdrojová_adresa port → cílová_adresa port hlášení (volby{flags: content: offset: depth: nocase: reference: classtype})

alert tcp $EXTERNAL_NET any → $HOME_NET 80 (msg:”Test”; sid:1000001;)

http://manual.snort.org/

Aplikační firewall (WEB)

F5, Imperva SecureSphere WAF, ModSecurity, Iron Bee

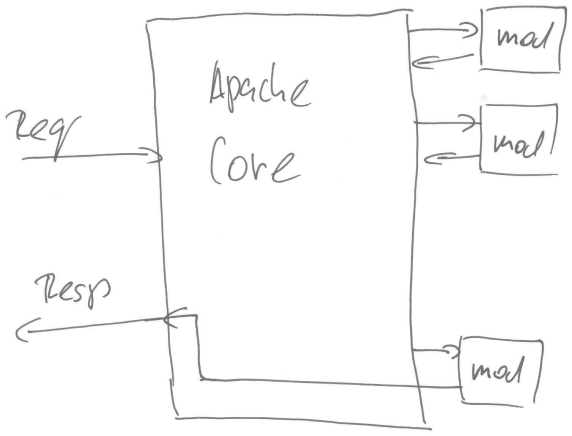
Slouží k detekci/blokování útoků hlavně na aplikační úrovni.
Analyzují HTTP požadavky i odpovědi včetně těla a alertují, nebo blokují podezřelé požadavky (odpovědi).
Zpravidla jsou zapojeny jako reverzní proxy k chráněným serverům, některé implementace je ale možné zapojit i na mirror port jako pouze detekční. WAF v režimu reverzní proxy může být samozřejmě nakonfigurován tak, aby útoky pouze detekoval a neblokoval (ladění).

Požadavky (výběrová kritéria) na WAF

Schopnost ochránit web server proti nejčastějším útokům (OWASP Top Ten).
Chránit proti útokům hrubou silou na autentizační mechanizmy.
Snadná, resp. automatická aktualizace databáze útoků.
Seznam typů útoků před kterými je systém schopen chránit.
Možnost nakonfigurovat WAF tak, aby chránil před libovolným specifickým problémem.
Schopnost pracovat ve třech režimech: Blokace a logování, Pouze logování, Pasivní režim.
Detekovat chráněné informace v odpovědích web serveru.
Možnost monitorování HTTPS spojení.
Podpora různých systémů kódování znaků.
Podpora XML.
Schopnost “učit se” resp. automatizovaně definovat správné/nesprávné chování uživatelů .
Schopnost automaticky identifikovat strukturu a logiku chráněné aplikace.
Možnost generování poplachů, tvorby reportů a analýzy útoků.
Schopnost propagace poplachů do SIEM/ISMS.
Možnost maskování citlivých dat v logu.
Jednoduché a intuitivní uživatelské rozhraní.
Velmi malá latence (milisekundy).
Řešení musí být samo o sobě bezpečné a musí umožňovat správu založenou na systému rolí.
Podpora řešení s vysokou dostupností
Obsahuje API umožňující rozšiřování systému.
Podpora systému.

ModSecurity

implementován jako modul začleněný do zpracování HTTP requestu/response



Funkce

- monitorování v reálném čase a řízení přístupu
- virtuální záplatování bezpečnostních nedostatků
- logování kompletního HTTP provozu
- pasivní bezpečnostní analýza
- zvýšení zabezpečení web aplikace

Pravidla (rules.conf)
- co provést se zpracovanými daty

Příklad pravidla:

SecRule ARGS "<script>" log,deny,status:404

- co hledáme ve vstupních datech <script>
- co se stane, pokud nalezneme zadaná data (log, deny, status:404)

Obecná struktura pravidla s vysvětlivkami relevantními k výše uvedenému příkladu:

SecRule VARIABLES OPERATOR ACTIONS

VARIABLES: kde hledat (ARGS = všechny vstupní parametry)
OPERATOR: jak hledat (regexp, který je aplikován na vstupní parametry)
ACTIONS: co dělat, pokud je nalezena shoda (zalogovat, a blokovat se statusem 404)

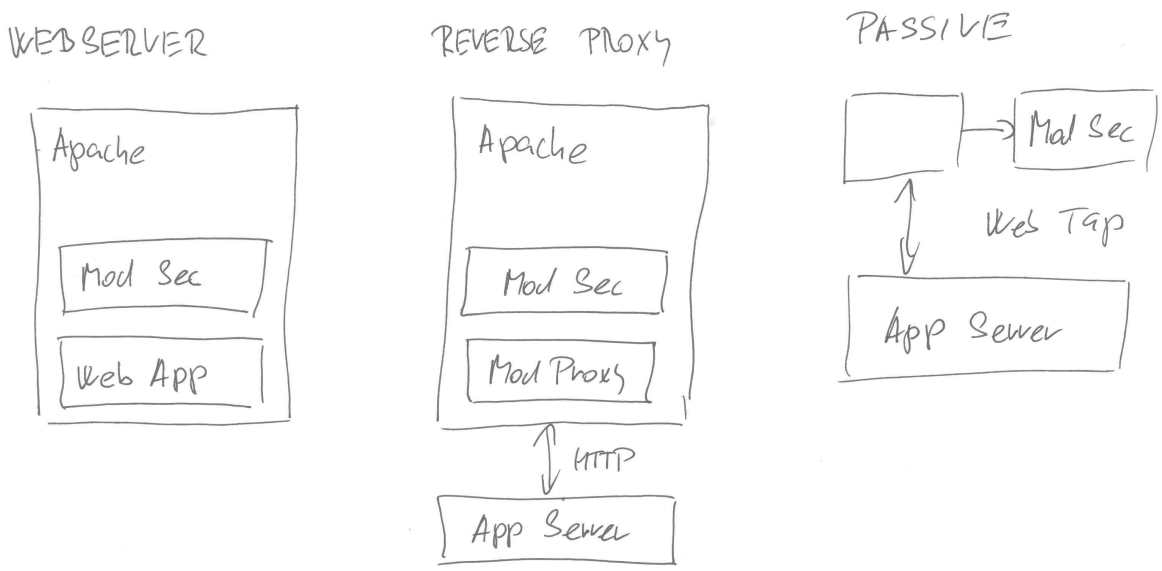
Pravidla je možné tvořit nad:

- hlavičkami požadavků
- tělem požadavků
- hlavičkami odpovědí
- tělem odpovědí
- logovací pravidla

v podstatě se jedná o pět fází zpracování.

Scénáře nasazení:

- webserver
- reverzní proxy
- pasivní režim



Konfigurace a spuštění:

V Apache konfiguraci uvést:

Include /opt/modsecurity/etc/modsecurity.conf

modsecurity.conf:

<IfModule mod_security2.c>
Include /opt/modsecurity/etc/main.conf
Include /opt/modsecurity/etc/rules-first.conf
Include /opt/modsecurity/etc/rules.conf
Include /opt/modsecurity/etc/rules-last.conf
</IfModule>

Aktivovat moduly:

```
  # Load libxml2
  LoadFile /usr/lib/libxml2.so
  # Load Lua
  LoadFile /usr/lib/liblua5.1.so
  # Finally, load ModSecurity
  LoadModule security2_module modules/mod_security2.so
```

Enable ModSecurity v pasivním režimu (neblokuje):

```
# Enable ModSecurity, attaching it to every transaction.
SecRuleEngine DetectionOnly
```

Chceme prohledávat i těla požadavků:

```
# Allow ModSecurity to access request bodies. If you don't,
# ModSecurity won't be able to see any POST parameters
# and that's generally not what you want.
SecRequestBodyAccess On
```

Buffering:

```
# Maximum request body size we will accept for buffering.
# If you support file uploads then the value given on the
# first line has to be as large as the largest file you
# want to accept. The second value refers to the size of
# data, with files excluded. You want to keep that value
# as low as practical.
SecRequestBodyLimit 1310720
SecRequestBodyNoFilesLimit 131072
```

```
# Store up to 128 KB of request body data in memory. When
# the multipart parser reaches this limit, it will start
# using your hard disk for storage. That is generally slow,
# but unavoidable.
SecRequestBodyInMemoryLimit 131072
```

Odpovědi.

```
# Allow ModSecurity to access response bodies. We leave
# this disabled because most deployments want to focus on
# the incoming threats, and leaving this off reduces
# memory consumption.
SecResponseBodyAccess Off

# Which response MIME types do you want to look at? You
# should adjust the configuration below to catch documents
# but avoid static files (e.g., images and archives).
SecResponseBodyMimeType text/plain text/html

# Buffer response bodies of up to 512 KB in length.
SecResponseBodyLimit 524288

# What happens when we encounter a response body larger
# than the configured limit? By default, we process what
# we have and let the rest through.
# nebo Reject
SecResponseBodyLimitAction ProcessPartial
```

Umístění pracovních souborů v souborovém systému.

```
# The location where ModSecurity will store temporary files
# (for example, when it needs to handle a multipart request
# body that is larger than the configured limit). If you don't
# specify a location here your system's default will be used.
# It is recommended that you specify a location that's private.
SecTmpDir /opt/modsecurity/var/tmp/

# The location where ModSecurity will keep its data. This,
# too, needs to be a path that other users can't access.
SecDataDir /opt/modsecurity/var/data/
```

Uploady souborů.

```
# The location where ModSecurity will store intercepted
# uploaded files. This location must be private to ModSecurity.
SecUploadDir /opt/modsecurity/var/upload/

# By default, do not intercept (nor store) uploaded files.
SecUploadKeepFiles Off
```

```
# Uploaded files are by default created with permissions that
# do not allow any other user to access them. You may need to
# relax that if you want to interface ModSecurity to an
# external program (e.g., an anti-virus).
SecUploadFileMode 0600
```

Logy

Debug:

```
# Debug log
SecDebugLog /opt/modsecurity/var/log/debug.log
SecDebugLogLevel 3
```

Audit:
(loguje kompletní data transakce, cca 1KB na požadavek)

```
# Log only what is really necessary.
SecAuditEngine RelevantOnly
# jenom error a warning
# On loguje vsechno, Off nic

# Also log requests that cause a server error.
SecAuditLogRelevantStatus ^5
# kody 500 az 599

# Log everything we know about a transaction.
SecAuditLogParts ABCDEFHKZ

# Use a single file for logging.
SecAuditLogType Serial
SecAuditLog /opt/modsecurity/var/log/audit.log

# Specify the path for concurrent audit logging.
SecAuditLogStorageDir /opt/modsecurity/var/audit/
```

Různé

```
SecArgumentSeparator &

# 0 je Netscape style cookies, neboli version 0 cookies
SecCookieFormat 0


# Implicitni chovani pravidel
# lze predefinovat v pravidlu
# pravidlo vsak neprepise chovani definovane v SecRuleEngine
SecDefaultAction "phase:1,log,auditlog,pass"
```

Ověření instalace.

- Vytvoříme testovací pravidlo, které bude hledat ve všech parametrech slovo “pokus” a pokud ho najde, odpoví kódem 503 (služba nedostupná):

SecRule ARGS pokus "phase:1,log,deny,status:503"

- restartujeme Apache
- pošleme požadavek s testovacím řetězcem “pokus” v některém parametru
- překontrolujeme error.log, debug.log a audit.log
- otestujeme POST
- otestujeme blokování spouštění scriptů
- odstraníme testovací pravidlo a restartujeme Apache

Honeypoty

honeyd, dionaea, honeynet project

Démoni simulující reálné služby a umožňující detekovat a analyzovat útoky.
Často nasazovány ve formě celých sítí, které jsou routované, ale neobsahují žádné jiné služby. Provoz směrovaný do těchto sítí tak lze z vyšší mírou pravděpodobnosti považovat za útoky.

honeyd

honeyd.conf:

```
create windows
set windows personality "Microsoft Windows XP Professional SP1"
set windows uptime 1728650
add windows tcp port 80 "sh /usr/share/honeyd/scripts/win32/web.sh"
set windows default tcp action reset
set windows ethernet "00:00:24:ab:8c:12"
dhcp windows on eth0

create default
set default default tcp action block
set default default udp action block
set default default icmp action block
```

```
# /usr/bin/honeyd -d -f ./honeyd.conf
```

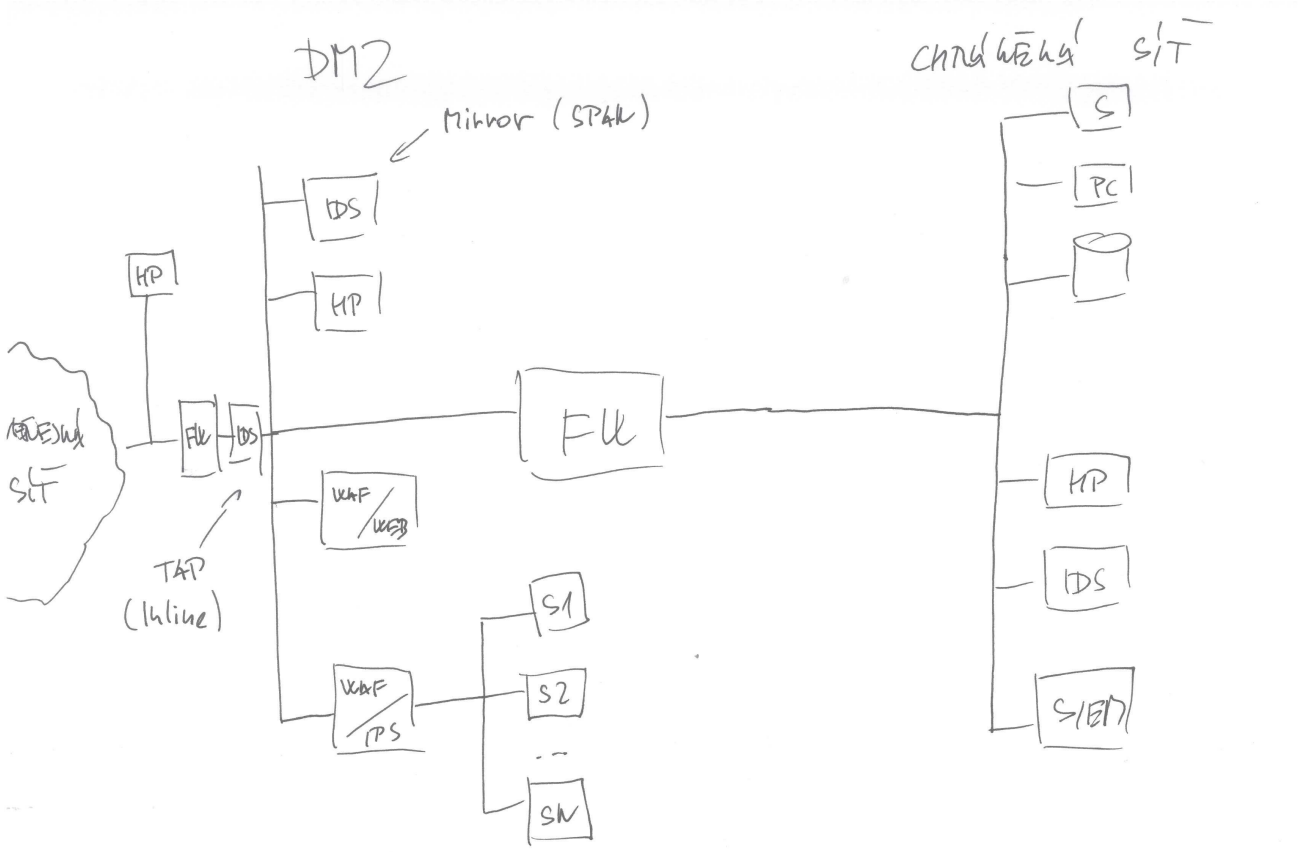
(http://travisaltman.com/honeypot-honeyd-tutorial-part-1-getting-started/)

SIEM

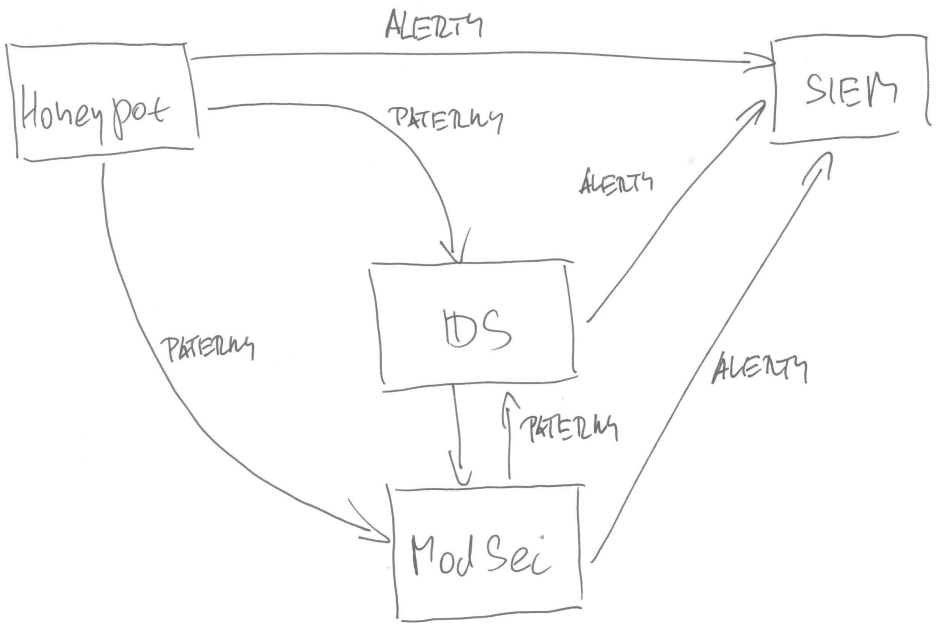
- agregace dat
- korelace dat
- generování poplachů - alerting
- shoda s požadavky (normy, zákonné požadavky) – compliance
- retence bezpečnostně relevantních dat
- forenzní analýza
- konzole

Návrh propojení popsaných komponent:

Síťově:



Logicky:


ALERTY
Honeypot
SIEM
PATERNY
ALERTY
IDS
PATERNY
PATERNY
ALERTY
ModSec

Zápočet:

Napsat snort pravidlo.

Napsat ModSecurity pravidlo.

Nakonfigurovat IIS web server honeypot.

Otázky:

1) Co to jsou IDS/IPS, jejich klasifikace, způsoby použití.
2) Uveďte základní prvky konfigurace IDS Snort.
3) Co to je aplikační firewall? Uveďte příklady a možnosti zapojení do síťové infrastruktury.
4) Uveďte základní prvky konfigurace WAF mod_security
5) Co to je Honeypot/Honeynet a k čemu slouží?
6) Uveďte základní prvky konfigurace honeyd.
7) Co to je SIEM a k čemu slouží?