VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

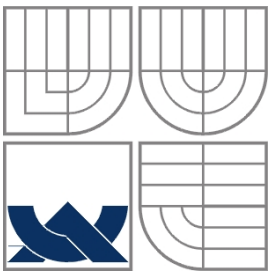
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA SYSTÉMOVÝCH ZÁZNAMOV

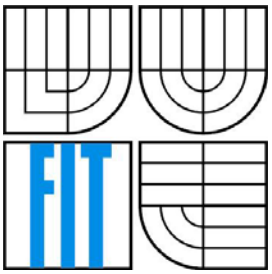
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE MARTIN GRACIK
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA SYSTÉMOVÝCH ZÁZNAMŮ

SYSTEM LOG ANALYSIS

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN GRACIK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ALEŠ SMRČKA

BRNO 2008

Abstrakt

Táto práca sa zaoberá problematikou analýzy systémových záznamov a jej použitím na detekciu vniknutia do systému. Prvá časť je zameraná na oboznámenie sa s rôznymi technikami analýzy. Druhá časť sa zaoberá nástrojom OSSEC, ktorý túto analýzu využíva na detekciu vniknutia do systému. V poslednej časti práce je návrh a implementácia grafického nástroja pre jednoduchú konfiguráciu OSSEC.

Kľúčové slová

Analýza, systémové záznamy, logy, OSSEC, grafické rozhranie, GUI

Abstract

This thesis discusses system log analysis and it's usage for intrusion detection. First part is about different techniques used for log file analysis. Second part is about OSSEC, a tool which uses log analysis to detect intrusion into the system. The last part talks about design and implementation of a graphical user interface for easy configuration of OSSEC.

Keywords

Analysis, system log files, logs, OSSEC, graphical user interface, GUI

Citácia

Gracik Martin: Analýza systémových záznamov. Brno, 2008, bakalárska práca, FIT VUT v Brne.

Analýza systémových záznamov

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Aleša Smrčku. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Martin Gracik
20.1.2008

Poďakovanie

Chcel by som poďakovať vedúcemu práce Ing. Alešovi Smrčkovi za jeho pripomienky, rady a konzultácie a takisto Danielovi B. Cidovi, tvorcovi OSSEC za odpovede na moje otázky ohľadom OSSEC.

Obsah

Úvod 2
1 Analýza systémových záznamov 3
1.1 Analýza proxy záznamov 3
1.1.1 Vnútorný užívatelia skenujú, alebo útočia na vonkajšie systémy 4
1.1.2 Vnútorný užívatelia s červami, trojanmi alebo vírusmi 4
1.1.3 Neplatný užívatelia v sieti 5
1.1.4 Zneužitie proxy 5
1.1.5 Porušenie firemnej politiky 6
1.2 Analýza internetových záznamov 6
1.2.1 Skenovanie servera alebo získavanie informácií 6
1.2.2 Úspešné a neúspešné útoky na internetové aplikácie 7
1.2.3 Problémy s internetovým serverom 7
1.3 Analýza autentifikačných záznamov 8
1.3.1 Užívatelia pristupujúci tam, kam nemajú 8
1.3.2 Prihlásenie systémového užívateľa 8
1.3.3 Viacero neúspešných prihlásení 9
1.3.4 Mnoho neúspešných pokusov o prihlásenie sa, nasledovaných úspešným prihlásením 9
2 OSSEC 10
2.1 Konfigurácia 10
2.2 Analýza záznamov a pravidlá 11
2.3 Správa agentov 11
2.4 Aktívna odozva 12
2.4.1 Konfigurácia aktívnej odozvy 12
3 Grafické rozhranie pre konfiguráciu OSSEC 14
3.1 Základné rozloženie GUI 15
4 Implementácia 19
4.1 Použité nástroje 19
4.2 Grafické rozhranie 19
4.3 Programová časť 20
4.3.1 Hlavný program 20
Záver 22
Literatúra 23
Zoznam príloh 24

Úvod

V moderných systémoch sú väčšinou všetky dôležité informácie o udalostiach, ktoré sa uskutočňujú v aplikáciách zaznamenávané do systémových záznamov (logov), ktoré nám umožňujú ich spätnú kontrolu. Ich prehliadanie a porozumenie nám môže poslúžiť v rôznych situáciách, ako napríklad pri vytváraní rôznych štatistík, ale aj pri ochrane pred nebezpečnými útokmi.

Cieľom tejto práce je oboznámenie so základnými princípmi analýzy systémových záznamov a jej využitie pri detekcii a prevencii pred prípadným vniknutím do systému.

V prvej kapitole sa budeme zaoberať samotnou analýzou systémových záznamov. Povieme si aké funkcie by mal zvládať nástroj pre automatickú analýzu systémových záznamov a aké rôzne útoky môžeme odhaliť pri analýze systémových záznamov generovaných rôznymi aplikáciami.

V druhej kapitole sa oboznámime s už existujúcim nástrojom OSSEC, ktorý patrí do skupiny nástrojov nazývaných HIDS (Host-based Intrusion Detection System) a pomocou monitorovania systémových záznamov dokáže odhaliť prípadné útoky na systém, alebo aj možné chyby v aplikáciách. Povieme si o funkciách, ktoré dokáže vykonávať a taktiež o možnostiach jeho konfigurácie tak, aby vyhovoval našim požiadavkám.

V tretej a štvrtej kapitole je popísaný vlastný návrh a implementácia grafického užívateľského rozhrania, ktoré umožňuje jednoduchú konfiguráciu nástroja OSSEC. Sú tu uvedené spôsoby a nástroje, ktoré boli použité pre implementáciu výslednej aplikácie.

V poslednej kapitole sú uvedené námety na ďalšie rozšírenia a možnosti ďalšieho vývoja.

1 Analýza systémových záznamov

Analýza systémových záznamov je jeden z najviac prehliadaných aspektov detekcie vniknutia do systému. Dnes už má skoro každý domáci počítač aspoň antivírusový program a aj najbežnejší užívatelia si kupujú najnovšie bezpečnostné nástroje. Avšak málokto pravidelne prezerá a monitoruje všetky informácie, ktoré tieto nástroje generujú a dokonca niekto neprezerá ani záznamy internetových serverov, poštových serverov, alebo autentifikačné záznamy. Nemám teraz na mysli pekné štatistiky prístupov na stránku, ale kritické bezpečnostné informácie, ktoré majú len určité udalosti a napriek tomu zostávajú nepovšimnuté. Veľa útokov by sa vôbec neuskutočnilo, alebo by boli oveľa skôr zastavené, keby administrátori kontrolovali svoje systémové záznamy.

Analýza systémových záznamov nie je ľahká a každodenne manuálne prezerať všetky systémové záznamy by bolo nemožné. A práve kvôli ich zložitosti a väčšinou veľkému obsahu je potrebná automatická analýza systémových záznamov.

Utilita na analýzu systémových záznamov by mala zvládať nasledujúce funkcie:

- Pochopiť systémové záznamy. Vedieť čo je dobré a čo je zlé.
- Korelovať zlé udalosti hľadajúc vzorky, ktoré môžu naznačovať útok alebo vniknutie.
- Korelovať dobré udalosti so zlými udalosťami (napríklad veľa neúspešných prihlásení, po ktorých nasleduje úspešné prihlásenie).
- Korelovať dobré udalosti (napríklad príliš mnoho úspešných prihlásení pre jedného užívateľa z rôznych zariadení počas malého časového intervalu).
- Hľadať nezvyklé vzorky, ktoré nie sú v zozname ani zlých, ani dobrých udalostí.

Samozrejme, že vykonávanie všetkých týchto krokov nie je jednoduché. V nasledujúcich kapitolách bude vysvetlené, ako môžu byť niektoré hrozby odhalené koreláciou špecifických vzoriek z internetových, proxy alebo autentifikačných systémových záznamov.

V príkladoch bude použitý nástroj OSSEC, ktorého funkcionalita nám umožňuje vykonávať všetku spomínanú analýzu záznamov.

1.1 Analýza proxy záznamov

Pri štúdiu proxy záznamov bola väčšinou použitá ako príklad aplikácia squid, pretože je to asi najpoužívanejšia internetová proxy. Ak je squid implementovaný správne, tak všetky dáta prejdú cez, bez nejakej extra konfigurácie na strane užívateľa. Vďaka tomu máme plný prístup ku každej stránke, ktorú užívateľ navštívi. To, čo odhalíme pri analýze proxy záznamov, je väčšinou veľmi dôležité, pretože to pochádza z vnútra našej siete. V nasledujúcich podkapitolách bude popísané aké problémy môžeme odhaliť pri monitorovaní proxy záznamov.

1.1.1 Vnútorný užívatelia skenujú, alebo útočia na vonkajšie systémy

Prakticky vždy, keď sa užívateľ snaží pripojiť na neexistujúcu stránku, squid zapíše do logu správu HTTP error code (po väčšinou s číslom 404 alebo 403). Ak vidíte viacero chybových hlásení s číslom 400 z rovnakej zdrojovej IP adresy za malý časový interval, malo by sa s tým niečo vykonať.

Ak užívateľ navštevuje stránku s nesprávnymi odkazmi, vygenerujeme falošné nálezy, takže ignorujeme prípony ako .jpg, .gif, .png a iné. Pri tomto prístupe môžeme odhaliť vnútorných užívateľov, ktorý sa pokúšajú skenovať, alebo získať informácie z vonkajších systémov alebo stránok. NIDS (Sieťový systém detekcie vniknutia) by tento problém nezachytil.

1.1.2 Vnútorný užívatelia s červami, trojanmi alebo vírusmi

Mnoho červov ma špecifické spôsoby pristupovania k internetovým alebo externým stránkam. Odhalenie týchto prístupov môže indikovať infikovaného vnútorného používateľa. Napríklad v prípade W32.Beagle červa, infikovaný systém sa bude pokúšať pristupovať na stránky s príponou "xxx3.php" alebo "blst.php", aby kontaktoval tvorcu červa. Ak vidíme tieto správy v záznamoch, vieme, že je niečo zle.

Toto je príklad OSSEC hlásenia infikovaného užívateľa:

```
OSSEC HIDS Notification.
2006 May 11 11:00:00

Received From: (web-proxy) 192.168.2.1->/usr/local/squid/var/log/access.log
Rule: 5054 fired (level 12) -> "Infected machine with W32.Beagle.DP.'"
Portion of the log(s):

524 192.168.2.204 TCP_MISS/404 590 GET http://www.ordendeslichts.de/intern/xxx3.php? -
DIRECT/81.201.107.6 text/html
3571 192.168.2.204 TCP_MISS/404 470 GET http://www.levada.ru/htmlarea/images/xxx3.php? -
DIRECT/62.118.252.213
466 192.168.2.204 TCP_MISS/404 543 GET
http://www.etype.hostingcity.net/mysql_admin_new/images/xxx3.php? - DIRECT/217.158.10.80
text/html
516 192.168.2.204 TCP_MISS/404 423 GET http//www.deadlygames.de/DG/BF/BF-
Links/clans/xxx3.php? - DIRECT/81.169.145.95 text/html
528 192.168.2.204 TCP_MISS/404 423 GET http://stroyindustry.ru/service/construction/xxx3.php?
- DIRECT/217.16.16.135 text/html
950 192.168.2.204 TCP_MISS/404 686 GET http://service6.valuehost.ru/images/xxx3.php? -
DIRECT/217.112.42.95 text/html
505 192.168.2.204 TCP_MISS/404 1368 GET http://schiffsparty.de/bilder/uploads/xxx3.php? -
DIRECT/212.227.94.133 text/html
```

1.1.3 Neplatný užívatelia v sieti

Niektoré proxy požadujú autentifikáciu od užívateľa. Ak užívateľ nemá platné údaje, bude do systémového záznamu zapísaná správa o chybe autentifikácie. Ak vidíme jednu alebo dve chyby autentifikácie, je možné, že užívateľ len zabudol svoje heslo. Keď ale vidíme viac chýb z rovnakého zdroja, je to signál, že sa deje niečo zlé (hlavne, ak sú tieto pokusy pre rôzne užívateľské mená).

Tu sú dva príklady autentifikačnej chyby, ktorú vygeneroval squid:

```
1134746808.068 34 10.1.2.3 TCP_DENIED/407 2675 GET http://www.test.com/ user NONE/- text/html
1096907971.215 4 10.1.2.4 TCP_DENIED/407 3715 GET http://www.microsoft.com/isapi/redir.dll? -
NONE/- text/html
```

1.1.4 Zneužitie proxy

Pri vytvorení proxy pravdepodobne chcete, aby bola používaná len na internetovú komunikáciu. Avšak, niektorí užívatelia sa môžu pokúsiť zneužiť niektoré podobnosti protokolov na odosielanie externých emailov, alebo prenos iných protokolov, ktoré by inak boli blokované. Napríklad, veľmi bežným problémom so squid-om je, že užívatelia sa pokúšajú použiť ho na prenos emailov. Toto môže byť jednoducho ošetrené zakázaním portu 25, ale aj potom je dobré vedieť, kto sa o tento prenos pokúša. Monitorovaním prístupov na zakázané porty môžeme odhaliť útočiacich užívateľov.

Nasledujúce hlásenie ukazuje vnútorného užívateľa, ktorý sa snažil preniesť email cez proxy:

```
OSSEC HIDS Notification.
2006 May 12 07:05:12

Received From: (web-proxy) 192.168.2.1->/usr/local/squid/var/logs/access.log
Rule: 5051 fired (level 10) -> "Multiple attempts to access forbidden file or directory from
same source ip.'"
Portion of the log(s):

0 192.168.2.135 TCP_DENIED/403 1382 CONNECT 65.54.245.104:25 - NONE/- text/html
2 192.168.2.135 TCP_DENIED/403 1378 CONNECT 4.79.181.14:25 - NONE/- text/html
0 192.168.2.135 TCP_DENIED/403 1390 GET http://www.ebay.com/ - NONE/- text/html
3 192.168.2.135 TCP_DENIED/403 1378 CONNECT 4.79.181.14:25 - NONE/- text/html
5 192.168.2.135 TCP_DENIED/403 1392 GET http://www.yahoo.com/ - NONE/- text/html
6 192.168.2.135 TCP_DENIED/403 1384 CONNECT 66.135.192.123:80 - NONE/- text/html
2 192.168.2.135 TCP_DENIED/403 1380 CONNECT 66.94.230.75:80 - NONE/- text/html
420 192.168.2.135 TCP_DENIED/403 1390 GET http://www.ebay.com/ - NONE/- text/html
6 192.168.2.135 TCP_DENIED/403 1384 CONNECT 66.135.192.123:25 - NONE/- text/html
```

1.1.5 Porušenie firemnej politiky

Niektoré spoločnosti nepovoľujú prístup k externým internetovým poštovým klientom alebo stránkam s pornografickým obsahom v práci. So squid-om môžete priamo zakázať tieto prístupy, ale potrebujete vedieť, kto sa pokúša pristupovať k týmto zakázaným stránkam. S použitím analýzy systémových záznamov môžete nastaviť zoznam nepovolených internetových stránok alebo IP adries, ktoré budú zablokované, alebo pri pokuse o prístup na ne, sa vygeneruje hlásenie. V OSSEC nie je táto funkcionalita predvolená, pretože každá spoločnosť má svoju vlastnú politiku.

1.2 Analýza internetových záznamov

Niektorý ľudia veria NIDS (Network-based IDS), ako napríklad Snort, že odhalia útoky na ich internetové aplikácie. Väčšina NIDS ale nezachytí veľa dôležitých informácií, ako napríklad vnútorné chyby, návratové hodnoty aplikácií atď. A ak vaša stránka používa SSL, sú NIDS úplne nepoužiteľné. Monitorovaním internetových záznamov môžeme odhaliť nasledujúce problémy.

1.2.1 Skenovanie servera alebo získavanie informácií

Keď sa niekto pokúša o neautorizovaný prístup k vašim systémom, bude ich pravdepodobne skenovať a hľadať zraniteľné aplikácie (ako napríklad staré verzie phpbb alebo awstats). To bude mať za výsledok, že váš internetový server vygeneruje mnoho chybových hlásení s číslom 400. Ak ich odhalíme mnoho za krátky časový interval z rovnakej zdrojovej IP adresy, mali by sme s tým niečo robiť. Niekedy sa môže stať, že to bude len falošný poplach, ak je na stránke veľa neplatných odkazov, preto ignorujeme prípony ako .gif, .jpg, .png (takisto ako to robíme pri squid záznamoch). Použitím tohto typu korelácie môžeme odhaliť nové červy alebo zero-day vulnerability v našich internetových aplikáciách.

Nasledujúcich niekoľko chybových hlásení s číslom 404 (hľadajúcich xmlrpc) sú príklady internetového skenovania:

```
100.149.117.1 - - [13/Jan/2006:01:03:30 -0200] "POST /blog/xmlrpc.php HTTP/1.0" 404 288
100.149.117.1 - - [13/Jan/2006:01:03:31 -0200] "POST /blog/xmlsrv/xmlrpc.php HTTP/1.0" 404
295
100.149.117.1 - - [13/Jan/2006:01:03:32 -0200] "POST /blogs/xmlsrv/xmlrpc.php HTTP/1.0" 404
296
100.149.117.1 - - [13/Jan/2006:01:03:33 -0200] "POST /drupal/xmlrpc.php HTTP/1.0" 404 290
100.149.117.1 - - [13/Jan/2006:01:03:35 -0200] "POST /phpgroupware/xmlrpc.php HTTP/1.0" 404
296
100.149.117.1 - - [13/Jan/2006:01:03:36 -0200] "POST /wordpress/xmlrpc.php HTTP/1.0" 404 293
100.149.117.1 - - [13/Jan/2006:01:03:44 -0200] "POST /xmlrpc/xmlrpc.php HTTP/1.0" 404 290
```

1.2.2 Úspešné a neúspešné útoky na internetové aplikácie

NIDS prezerajú len určitý obsah predtým ako oznámia, že sa niečo deje. Pri analýze záznamov môžeme vidieť či sa útok podaril alebo nie. Čo je ešte lepšie je to, že môžeme dokonca vidieť aj záznamy SSL pripojení, ktoré NIDS nie sú schopné vidieť. S malou množinou pravidiel môžeme odhaliť SQL injekcie, problémy s prechádzaním adresárov, pokusy o spustenie príkazov a rôzne iné útoky a vidieť naisto, či sa podarili alebo nie.

Napríklad, pre odhalenie SQL injekcií hľadáme niektoré SQL príkazy ako SELECT, WHERE alebo FROM. To isté platí aj pre odhalenie pokusov o spustenie príkazov. Každý systém ma podmnožinu príkazov, ktoré musíme monitorovať, ako napríklad cat, grep, wget, dir, ls a podobne. Takisto hľadáme medzery, nové riadky, alebo nulové ukončovače, pretože sú často používané (a potrebné) pre väčšinu pokusov o spustenie príkazov.

Na nasledujúcich dvoch útokoch na awstats vidíme niektoré bežné systémové príkazy, vidíme separátory a niektoré nezvyčajné znaky v adrese url. Pri pozeraní sa na výsledný HTTP kód, vieme, že jeden bol úspešný a druhý nie (chyba č. 404 a 200). S prístupom k týmto informáciám môžeme zvýšiť vážnosť toho, ktorý bol úspešný a minimalizovať vážnosť druhého, ktorý úspešný nebol. V OSSEC sa toto deje pre bežné internetové útoky. Ak sú úspešné zvýšime ich vážnosť a okamžite to oznámime administrátorovi a spustíme aktívnu odozvu.

```
a.b.c.d - - [13/Jan/2006:01:07:21 -0200] "GET
/awstats/awstats.pl?configdir=|echo;echo%20YYY;cd%20%2ftmp%3bwget...;echo%20YYY;echo|HTTP/1.0
" 404 291
a.b.c.d - - [14/Jan/2006:01:01:25 -0200] "GET /cgi-
bin/awstats.pl?configdir=|echo;echo%20YYY;cd%20%2ftmp%3bwget...;echo%20YYY;echo|HTTP/1.0" 200
291
```

1.2.3 Problémy s internetovým serverom

Veľa problémov s internetovým serverom môže byť odhalených pri prezeraní jeho záznamov. Napríklad nasledujúce chyby by boli prehliadnuté, keby sme ich nemonitorovali (OSSEC oznámenie poslané e-mailom):

```
OSSEC HIDS Notification.
2006 May 12 04:40:17

Received From: (web-server) 10.1.1.25->/var/log/apache/error_log
Rule: 102 fired (level 7) -> "Unknown problem somewhere in the system.'"
Portion of the log(s):

*** glibc detected *** corrupted double-linked list: 0xb7daca0c ***
```

```
OSSEC HIDS Notification.
2006 May 10 16:41:31

Received From: (intra-server) 10.1.2.41->/var/log/apache/error_log
Rule: 102 fired (level 7) -> "Unknown problem somewhere in the system.'"
Portion of the log(s):

[client 201.25.30.140] PHP Fatal error: Allowed memory size of 31457280 bytes exhausted
(tried to allocate 39518206 bytes) in /home/site/htdocs/components/com_search/search.php on
line 172, referer:
http://www.mysite.com.br/index.php?option=Itemid=5&searchword=+SNORT+%2B+MYSQL+%2B+APA
```

1.3 Analýza autentifikačných záznamov

Analýza autentifikačných záznamov je extrémne dôležitá. Za prvé preto, lebo môžeme vidieť kto kam pristupoval a kedy. Za druhé môžeme vidieť či niekto nepristupuje niekam kam by nemal. Takisto môžeme zistiť vnútorné zneužívanie tak, že vidíme v akom čase a na aké systémy sa užívatelia snažia pristupovať. Navyše môžu byť ešte odhalené útoky typu hrubej sily a iné problémy so skúšaním hesla.

1.3.1 Užívatelia pristupujúci tam, kam nemajú

Väčšina utilít na analýzu systémových záznamov upozorňuje len na neúspešné pokusy o prihlásenie sa. Čo sa ale stane, keď sa legitímny užívateľ snaží pristupovať na zariadenie, na ktoré by nemal mať prístup? Alebo sa o to snaží v čase, keď by nemal byť v práci?
Pri analýze si musíme vytvoriť základný zoznam všetkých užívateľov a zariadení, na ktoré majú prístup. Túto techniku používa OSSEC a volá sa FTS (First Time Seen). Vždy keď sa užívateľ pokúsi pristupovať na zariadenie, na ktorom ešte nikdy predtým nebol, OSSEC vygeneruje hlásenie. Prvých pár dní sa OSSEC učí, ktorý užívatelia pristupujú na ktoré zariadenia, čo spôsobuje niekoľko extra hlásení. Po nejakom čase sa však vytvorí základný zoznam a budú oznamované už len nepovolené prístupy. S FTS môžete odhaliť nelegálnych užívateľov v sieti alebo užívateľov, ktorý pristupujú na zariadenia, na ktoré by pristupovať nemali.

1.3.2 Prihlásenie systémového užívateľa

Systémové účty sú používané len na vnútorné účely a nemali by sme nikdy vidieť pokus o ich prihlásenie sa do systému. Ak vidíme niektorý z nich ako sa pokúša pripojiť pomocou ssh, telnetu, ftp alebo podobnej metódy, musíme si na to dať pozor. Môže to totiž znamenať, že bola kompromitovaná

nejaká aplikácia. OSSEC má zoznam užívateľov (napríklad apache, mysql, nobody, portmap, www, bin), ktorý je nápomocný pri ich identifikácii.

Príklad OSSEC hlásenia pri prihlásení sa užívateľa nobody:

```
OSSEC HIDS Notification.
2006 May 12 08:59:45

Received From: (auth1) 192.168.20.55->/var/log/messages
Rule: 1601 fired (level 12) -> "System user sucessfully logged on the system.'"
Portion of the log(s):

sshd[23410]: Accepted password for nobody from 10.1.2.3 port 42802 ssh2
```

1.3.3 Viacero neúspešných prihlásení

Útoky hrubou silou a slovníkové útoky sa stávajú stále častejšími, ale je možné ich zablokovať monitorovaním autentifikačných záznamov. Za prvé, ak vidíme mnoho neúspešných pokusov o prihlásenie sa z rovnakej zdrojovej IP adresy počas niekoľkých minút, je to pravdepodobne útok. Za druhé, ak vidíme mnoho neúspešných pokusov o prihlásenie sa z rôznych IP adries, je to pravdepodobne distribuovaný útok, alebo má náš systém nejaký vnútorný problém.

Toto je OSSEC hlásenie útoku hrubou silou na SSHD:

```
OSSEC HIDS Notification.
2006 May 11 21:17:07

Received From: /var/log/messages
Rule: 1512 fired (level 10) -> "SSHD brute force trying to get access to the system.'"
Portion of the log(s):

sshd[9370]: Failed password for invalid user admin from 200.30.175.162 port 58257 ssh2
sshd[9370]: Invalid user admin from 200.30.175.162
sshd[9368]: Failed password for invalid user fluffy from 200.30.175.162 port 58212 ssh2
sshd[9368]: Invalid user fluffy from 200.30.175.162
sshd[9366]: Failed password for invalid user slasher from 200.30.175.162 port 58109 ssh2
sshd[9366]: Invalid user slasher from 200.30.175.162
sshd[9364]: Failed password for invalid user sifak from 200.30.175.162 port 58030 ssh2
```

1.3.4 Mnoho neúspešných pokusov o prihlásenie sa, nasledovaných úspešným prihlásením

Toto je vážna udalosť. Ak vidíte niekoľko pokusov pre viacero užívateľov a rôzne heslá z rovnakej zdrojovej IP adresy nasledovaných úspešným prihlásením, mal pravdepodobne útočník šťastie. Môže

to ale takisto byť legitímny užívateľ, ktorý zabudol svoje heslo a po niekoľkých pokusoch si spomenul, čo bude znamenať falošný poplach. Pri pokusoch rôznych užívateľoch ale táto šanca klesá.

2 OSSEC

OSSEC je multi-platformný systém detekcie vniknutia s otvoreným zdrojovým kódom napísaný v jazyku C. Medzi podporované operačné systémy patrí Linux, OpenBSD, FreeBSD, MacOS, Solaris a Windows.

Vykonáva nasledujúce funkcie:

- analýza systémových logov
- kontrola integrity súborov
- monitorovanie Windows registrov
- detekcia rootkitov
- upozornenia v reálnom čase
- aktívna odozva

OSSEC môže bežať v dvoch rôznych módoch. V "local" móde, pri monitorovaní len jedného systému, napríklad domáceho počítača alebo malého servera, kde sa všetky funkcie vykonávajú priamo a len na lokálnom systéme. Alebo v "server-agents" móde, ktorý sa používa pri monitorovaní viac systémov v sieti z jedného centralizovaného systému. Pri tomto móde je OSSEC nainštalovaný na jeden systém ako "server" a na všetky ostatné ako "agent". Agenti potom posielajú všetky udalosti, ktoré zaznamenajú na server, ktorý ich spracúva a vyhodnocuje. Pri tejto inštalácii máte všetky nastavenia pravidiel na jednom mieste, čo umožňuje jednoduchšiu správu.

2.1 Konfigurácia

Celá konfigurácia OSSEC prebieha pomocou XML súborov. Hlavné nastavenia systému sú v súbore ossec.conf, ktorý má nasledujúce sekcie:

- global (štandardné nastavenia používané v celom systéme)
- email_alerts (nastavenia e-mailových upozornení)
- rules (zoznam súborov s pravidlami)
- syscheck (konfigurácia kontroly integrity)
- rootcheck (konfigurácia detekcie rootkitov
- alerts (nastavenia e-mail a log upozornení)

- localfile (nastavenia monitorovania lokálnych logov)
- remote (konfigurácia vzdialených spojení)
- client (konfigurácia agentov)
- database_output (nastavenia výstupu do databázy)
- command (konfigurácia aktívnej odozvy)
- active-response (konfigurácia aktívnej odozvy)

Nastavenia sa líšia podľa použitého módu, niektoré sú platné len pre agentov, niektoré len pre server, alebo lokálnu inštaláciu.

2.2 Analýza záznamov a pravidlá

Pravidlá analýzy sú taktiež uložené v XML súboroch, v ktorých môžte dynamicky špecifikovať, čo sa udeje s každou udalosťou, ktorá sa na systéme vyskytne.

Príklad syntaxe pravidla:

```
<rule id="1608" level="13" timeframe="120">
    <regex>^sshd[d+]: fatal: Local: crc32 compensation attack</regex>
    <if_matched_regex>^sshd[d+]: .+Corrupted check by bytes on</if_matched_regex>
    <comment>SSH CRC-32 Compensation attack</comment>
    <info>http://www.securityfocus.com/bid/2347/info/</info>
</rule>
```

OSSEC už pri inštalácii prichádza s mnohými prednastavenými pravidlami, ktoré môžete editovať, alebo si samozrejme nové pravidlá vytvoriť a pridať ku stávajúcim.

2.3 Správa agentov

Pri server-agents inštalácii prebieha komunikácia medzi serverom a agentmi šifrovane cez UDP port 1514. Preto treba povoliť tento port vo firewall-e a vytvoriť a importovať kľúče pre všetkých agentov. Kľúče sa generujú na serveri, z ktorého sa potom exportujú a následne importujú na agenta. Túto funkciu obstaráva príkaz manage_agents. Až po úspešnom importovaní kľúča bude môcť tento agent komunikovať so serverom.

2.4 Aktívna odozva

Aktívna odozva umožňuje automaticky spúšťať príkazy alebo akcie, keď sa vyskytne špecifická udalosť. OSSEC umožňuje spúšťať príkazy aj na strane servera, aj na strane agentov.

Výhodou tohto riešenia je rýchla odozva na možné útoky. Pri odhalení možného útoku môže byť okamžite vykonaná obranná reakcia. Je to extrémne výhodne hlavne voči rôznym skenovaniam portov, útokom hrubou silou a iným útokom, ktoré sa snažia zbierať údaje.

Avšak s tým prichádzajú aj riziká. Napríklad pri falošnom odhalení útoku môže byť zablokovaný oprávnený užívateľ, alebo ak útočník zistí, že je používaná aktívna odozva, môže sa pokúsiť o DoS útok.

Proti tomuto sa OSSEC snaží chrániť špecifikovaním zoznamu hostov, ktorý by nemali byť zablokovaní nikdy. Ďalej možnosťou nastavenia blokovania len pri pravidlách, u ktorých je malá šanca falošného odhalenia a takisto časovým vypršaním blokovania, kedy aj po chybnom zablokovaní užívateľa, získa po určitej dobe prístup znova.

2.4.1 Konfigurácia aktívnej odozvy

Rozdeľuje sa na dve časti. Vytvorenie príkazov aktívnej odozvy a priradenie týchto príkazov k pravidlám udalostí.

Vytvorenie príkazov má nasledujúci formát:

```
<command>
        <name>The name (A-Za-Z0-9)</name>
        <executable>The command to execute (A-Za-z0-9.-)</executable>
       <expect>Comma separated list of arguments (A-Za-z0-9)</expect>
       <timeout_allowed>yes/no</timeout_allowed>
</command>
```

Kde "name" je meno príkazu, ktoré slúži neskôr pri priradení ku pravidlám.

"executable" je názov spúšťacieho súboru, ktorý musí byť uložený v adresári /var/ossec/active-response/bin.

"expect" je zoznam argumentov, ktorý spúšťaný podprogram očakáva (možnosti sú srcip a username)

"timeout_allowed" určuje či tento príkaz podporuje časové vypršanie.

V druhej časti priradíme vytvorené príkazy k pravidlám udalostí následne:

```
<active-response>
    <disabled>Completely disables active response if "yes"</disabled>
    <command>The name of any command already created</command>
    <location>Location to execute the command</location>
    <agent_id>ID of an agent (when using a defined agent) </agent_id>
    <level>The lower level to execute it (0-9)</level>
```

```
    <rules_id>Comma separated list of rules id (0-9)</rules_id>
    <rules_group>Comma separated list of groups (A-Za-z0-9)</rules_group>
    <timeout>Time to block</timeout>
</active-response>
```

Štandardne prichádza OSSEC s nasledujúcimi pomôckami pre aktívnu odozvu:

- host-deny.sh - pridá IP adresu do súboru /etc/hosts.deny
- firewall-drop.sh - pridá adresu do iptables (ipfilter, ipfw, ipsec, pf) deny list

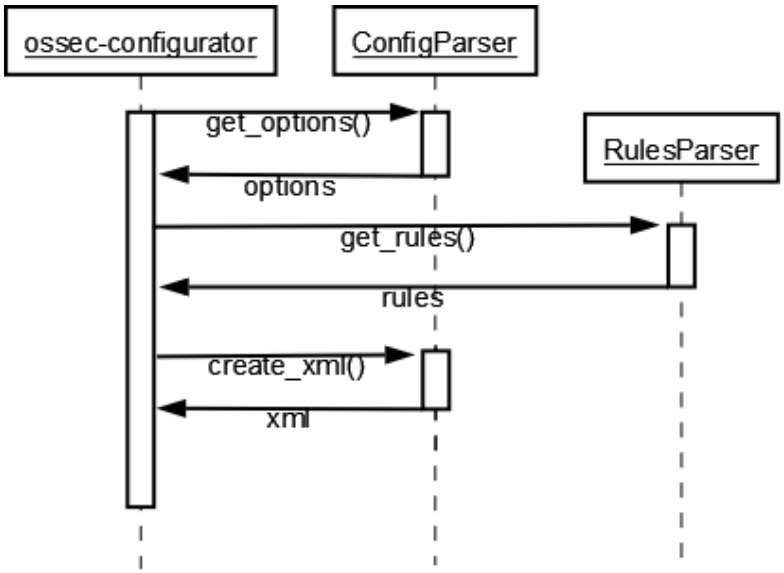
3 Grafické rozhranie pre konfiguráciu OSSEC

Grafické rozhranie má slúžiť na uľahčenie konfigurácie aplikácie OSSEC pre bežného užívateľa a napomôcť tak pri nastavovaní správania sa aplikácie, bez potreby editovania XML súborov textovým editorom.

Sú kladené nasledujúce požiadavky:

- rýchly prístup ku všetkým dôležitým nastaveniam OSSEC pre lokálnu inštaláciu na linuxovom operačnom systéme
- zrozumiteľnosť pre neskúseného užívateľa
- možnosť ponechať niektoré premenné nenastavené
- možnosť upravovať pravidlá analýzy
- možnosť pridať aktívnu odozvu založenú na pravidlách
- jednoduché pridanie ďalších možností nastavenia

Vzhľadom na požiadavky bude grafické rozhranie obsahovať grafickú možnosť nastavenia všetkých dôležitých nastavení, ktoré je možné nastaviť v XML súbore. Každá takto nastavovaná premenná bude mať detailný popis, aby bolo zrozumiteľné na čo slúži. Pri ponechaní niektorého nastavenia prázdneho bude do výsledného XML súboru vložená štandardná hodnota, čím zaručíme správne chovanie aplikácie ak si užívateľ nepraje meniť jej prednastavené správanie sa. Okrem možnosti nastavenia premenných aplikácie bude grafické rozhranie obsahovať takisto aj grafickú možnosť editácie súborov s pravidlami analýzy. Ďalej bude aplikácia obsahovať jednoduchú možnosť pridania aktívnej odozvy, kde si užívateľ bude môcť vybrať množinu pravidiel a príkaz, ktorý sa má vykonať po nájdení týchto útokov, špecifikovaných v pravidlách a jednoducho toto nastavenie pridať do XML súboru. Aplikácia bude navrhnutá tak, aby pri pridaní nových premenných nastavenia do aplikácie OSSEC bolo možné jednoducho pridať toto nastavenie aj do grafického rozhrania.

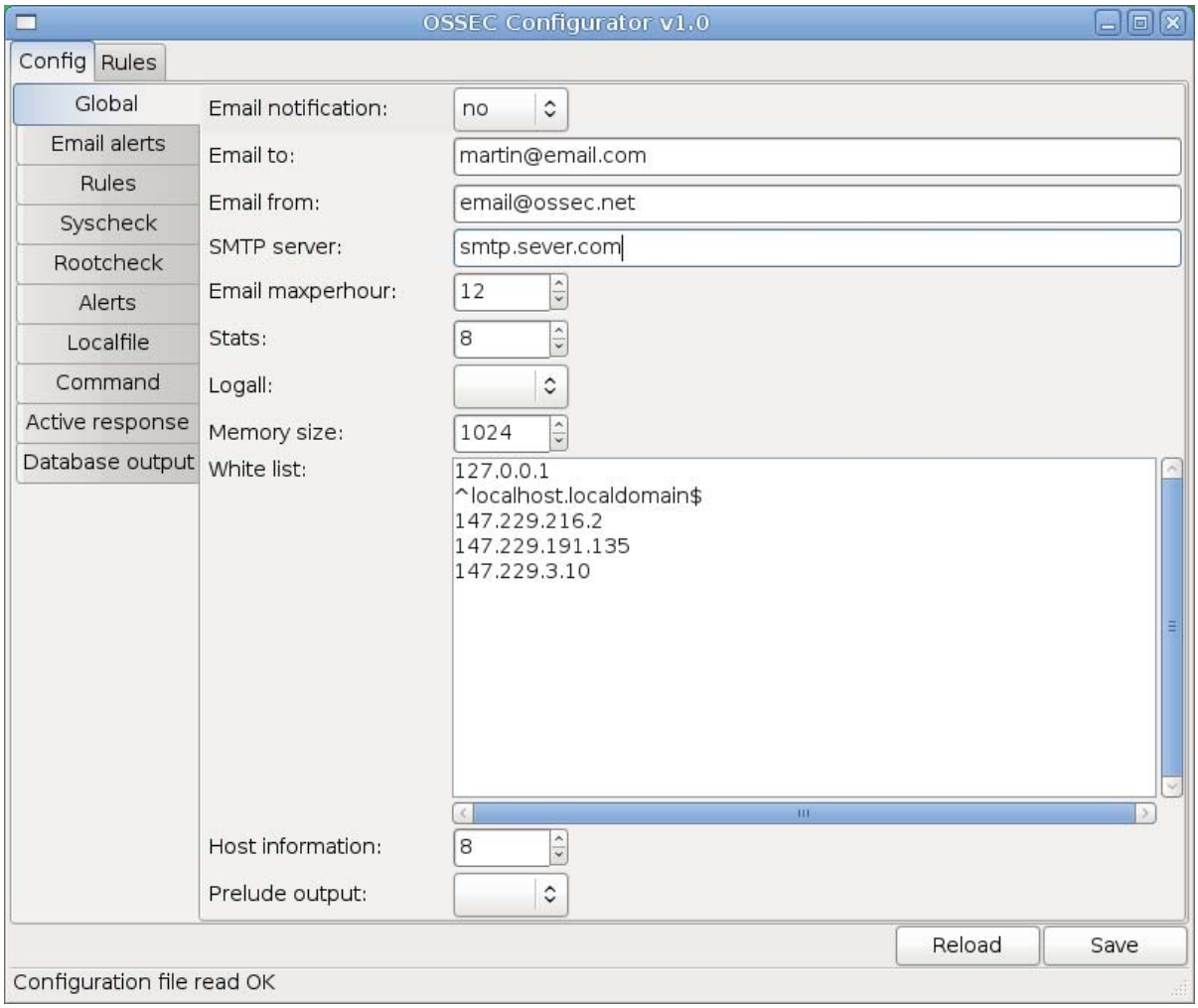


Obr. 1 Sekvenčný diagram návrhu

3.1 Základné rozloženie GUI

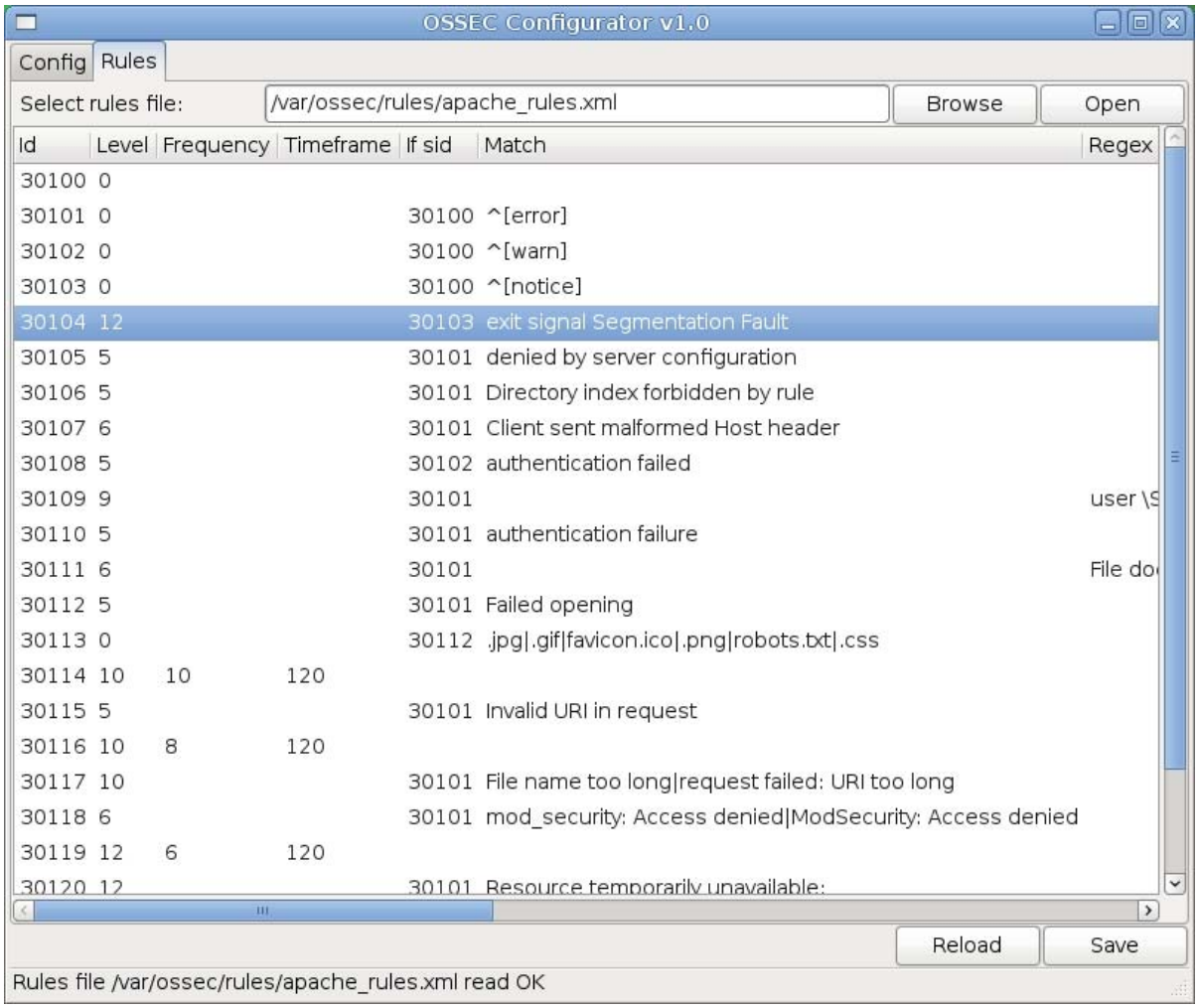
Aby bolo grafické rozhranie jednoduché a jednotné, prevádzajú sa všetky nastavenia v jednom aplikačnom okne, ktoré je rozdelené na niekoľko záložiek, podľa častí, ktorých sa týkajú.

Pri spustení sa zobrazí záložka Config, kde sa nastavujú všetky nastavenia, ktoré sa týkajú fungovania OSSEC a môžeme ich nájsť v súbore ossec.conf.



Obr. 2 Zobrazenie záložky Config

Druhou záložkou je záložka Rules, kde môžeme editovať pravidlá pre analýzu systémových záznamov dodávaných s OSSEC. Po zadaní cesty k súboru s pravidlami, alebo jeho nájdení v okne selektora súborov, ktoré otvoríme pomocou kliknutia na tlačidlo Browse a po následnom kliknutí na tlačidlo Open sa jeho obsah načíta do tabuľky, kde môžeme jednotlivé políčka editovať a následne uložiť do pôvodného XML súboru s pravidlami, ktorý sme otvorili.



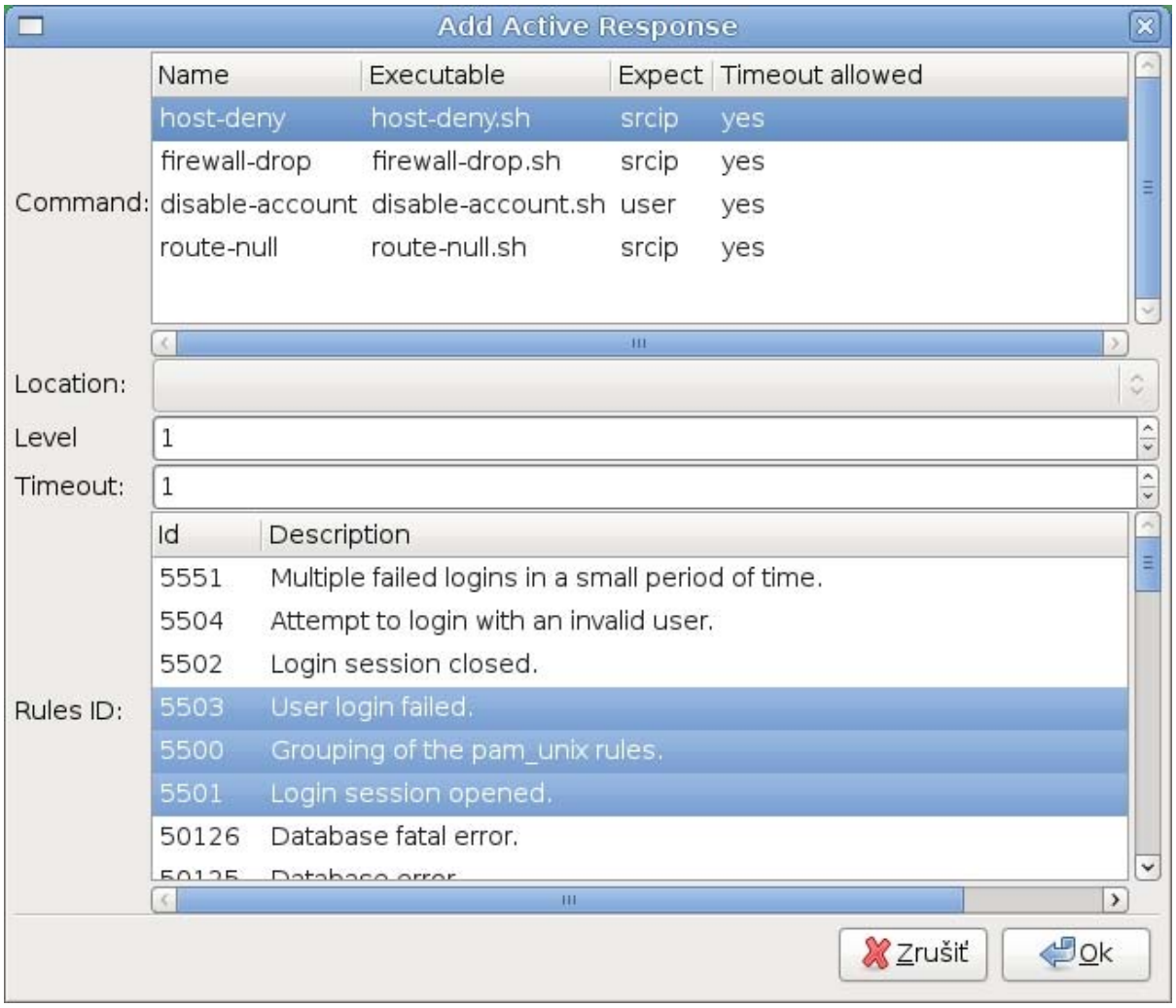
Obr. 3 Zobrazenie záložky Rules

Aby užívatelia, ktorí už sú zoznámení s formátom XML konfiguračných súborov, nemali problém rýchlo nájsť hľadanú hodnotu nastavenia, sú tieto záložky ďalej rozdelené na podsekcie, podľa odpovedajúcich podsekcií v textových konfiguračných súboroch, ktoré sú ale aj pre neskúseného užívateľa intuitívne pomenované, takže by aj skúsený, aj neskúsený užívateľ nemal mať problém nájsť hodnotu, ktorú chce zmeniť.

Takisto pri popise hodnôt nastavenia som sa snažil dodržiavať názvovú konvenciu z textových súborov, ktorá je podľa môjho názoru zrozumiteľná pre nového užívateľa, ktorý súbor s nastaveniami nepozná a na druhú stranu, znalému užívateľovi umožňuje priamo z grafického rozhrania vidieť, ktorú konkrétnu hodnotu nastavuje a kde ju potom prípadne môže vo výslednom textovom súbore nájsť. Ak chceme zistiť podrobnejšie informácie o tom čo vlastne ktorá hodnota znamená, stačí podržať kurzor myši na jej popisku a objaví sa nám vyskakovacie okno s konkrétnejším popisom nastavenia.

V podsekcii Active Response je oproti ostatným podsekciám navyše tlačítko Add Active Response, po stlačení ktorého sa otvorí nové aplikačné okno, kde môže užívateľ interaktívne pridať

novú aktívnu odozvu do OSSEC vybraním príkazu, ktorý sa ma spustiť a pravidiel zo zoznamu, pri nájdení ktorých sa tento príkaz ma vykonať. Po stlačení tlačidla OK sa táto odozva pridá do zoznamu k aktuálnym odozvám.



Obr. 4 Pridanie aktívnej odozvy

V spodnej časti aplikačného okna sú potom ešte 2 tlačítka, ktoré slúžia na znovu načítanie pôvodných hodnôt z konfiguračného súboru a na uloženie prevedených zmien. Nechýba samozrejme ani stavový riadok, ktorý nás informuje o funkciách vykonávaných aplikáciou, ako napríklad o úspešnom otvorení súboru s pravidlami, o pokuse o otvorenie nevalidného XML súboru, alebo o úspešnom uložení našich zmien do súboru.

4 Implementácia

Pri implementácii som dbal na to, aby použité nástroje umožňovali dobrú prenositeľnosť medzi platformami v prípade budúceho rozšírenia aj pre iné operačné systémy, na ktorých OSSEC dokáže bežať. Takisto som sa snažil použiť výhradne nástroje s otvoreným zdrojovým kódom, ktoré sú voľne k dispozícii.

4.1 Použité nástroje

Pre implementáciu grafického rozhrania bola použitá grafická knižnica GTK+. GTK+ je multi-platformná grafická knižnica pre tvorbu grafických užívateľských rozhraní. Patrí medzi najpopulárnejšie grafické knižnice pre X Window System. Pôvodne bola vytvorená pre GIMP (GNU Image Manipulation Program) v roku 1997. Je vytvorená v jazyku C, má otvorený zdrojový kód a je súčasťou GNU Projektu. Je šírená pod licenciou LGPL.

Rozhranie bolo navrhnuté v nástroji Glade, ktorý slúži na návrh grafických rozhraní v GTK+. Glade je utilita pre rýchlu tvorbu aplikácií, ktorá umožňuje rýchlu a jednoduchú tvorbu grafických užívateľských rozhraní využívajúcich grafickú knižnicu GTK+. Glade umožňuje navrhnúť grafické rozhranie pomocou grafického nástroja a následne toto rozhranie uložiť ako XML súbor, v ktorom sú uložené všetky informácie o rozhraní potrebné pre výslednú aplikáciu, ktorá si tento súbor pomocou libglade môže počas behu dynamicky načítavať.

Programová časť GUI je vytvorená v programovacom jazyku Python. Python je dynamický, objektovo orientovaný programovací jazyk, ktorý môže byť využitý na rôzne účely. Má veľké množstvo štandardných knižníc, vďaka ktorým je vývoj v ňom veľmi rýchly. Je dostupný pre väčšinu populárnych operačných systémov ako Unix/Linux, Windows, Mac OS. Je takisto vyvíjaný ako projekt s otvoreným zdrojovým kódom a môže byť použitý zdarma aj v komerčných aplikáciách.

Všetky použité nástroje nám zaručujú dobrú prenositeľnosť medzi rôznymi operačnými systémami.

4.2 Grafické rozhranie

Pri implementácii grafického rozhrania v nástroji Glade som použil záložky, ktoré rozdeľujú nastavenia do rôznych skupín. Každé nastavenie má svoj vlastný popisok a pri podržaní kurzoru myši nad ním sa nám zobrazí detailnejší popis vo vyskakovacom okne. Pre nastavenia, ktoré môžu mať viacero hodnôt a takisto aj pre editáciu pravidiel som použil gtk.TreeView, ktorý je vyplnený dátovou

štruktúrou gtk.ListStore a v konečnom dôsledku vytvára editovateľnú tabuľku, kde v hlavičke vidíme názov nastavenia a kliknutím do políčka tabuľky môžeme žiadanú hodnotu zmeniť na nami zvolenú novú hodnotu.

4.3 Programová časť

Programová časť využíva 2 vlastné moduly.

Modul ConfigParser slúži na prácu s XML súborom ossec.conf. Obsahuje metódy, ktoré tento súbor načítajú a vytvoria z jeho obsahu vlastnú dátovú štruktúru typu slovník, s ktorou potom hlavná aplikácia pracuje. Načítanie tohto XML súboru je implementované pomocou parseru typu SAX, v ktorom je postupne prechádzaný celý dokument a hodnoty sú prideľované do premenných v závislosti na elemente, v ktorom sa nachádzajú. Tento štýl načítania som sa rozhodol použiť pre jeho rýchlosť a menšie pamäťové nároky voči iným typom a pre jednoduchšiu prácu s vytvorenou vlastnou dátovou štruktúrou, ktorá je jednotnejšia ako štruktúra XML súboru. Takisto obsahuje metódu na spätné vytvorenie XML súboru z hodnôt v tejto dátovej štruktúre, čo nám umožňuje potom v jednoduchosti túto štruktúru uložiť do súboru.

Modul RulesParser slúži na načítavanie XML súborov s pravidlami analýzy pre OSSEC. Obsahuje metódy na jednoduché prečítanie ID pravidiel a ich popisov, ktoré potom využívame napríklad pri pridávaní aktívnej odozvy. Tento modul používa na prácu s XML súbormi parser typu DOM, ktorý vytvorí z celého XML súboru stromovú štruktúru, do ktorej môžeme následne pristupovať podľa mien elementov a takisto meniť ich hodnoty.

Pri implementácii som sa snažil vytvoriť generické triedy a funkcie, ktoré by dokázali jednotne spracúvať všetky možné nastavenia, ktoré sa nachádzajú v XML súboroch. Tieto sú ale natoľko rôzne a špecifické, že som v konečnom dôsledku musel použiť veľa výnimiek, aby bolo možné bezchybne všetko načítať.

Všetky mená elementov sú uložené zoznamoch na začiatku zdrojového kódu, ktoré potom využívam v triedach spracúvajúcich XML súbory, čo nám umožňuje uľahčenie pridávania nových možností nastavenia v prípadnom budúcom rozšírení OSSEC.

4.3.1 Hlavný program

Hlavný program ossec-configurator.py používa spomínané moduly na načítanie potrebných informácií zo súborov a následné doplnenie týchto hodnôt do políčok v grafickom rozhraní. A samozrejme pri ukladaní nastavení naopak prečíta hodnoty z políčok grafického rozhrania a uloží ich do výsledných textových súborov.

Keďže ku konfiguračným súborom OSSEC a takisto aj k súborom s pravidlami má v systéme prístup len užívateľ root, sú jeho práva potrebné aj pri spúšťaní grafického užívateľského rozhrania. Ak nie je prihlásený užívateľ root, bude pri spustení grafického rozhrania vyžadované heslo užívateľa root a aplikácia sa spustí až po jeho úspešnom zadaní. Ak je zadané nesprávne heslo aplikácia sa nespustí a tak nie je možné editovať nastavenia OSSEC. Týmto je zabezpečené bezpečné použitie grafického rozhranie, pretože neautorizovaný užívateľ nemá možnosť nastavenia nie len meniť ale ani prehliadať.

Pre správnu funkcionalitu programu je potrebné, aby konfiguračný súbor a takisto aj súbory s pravidlami, ktoré chceme editovať, boli validné XML súbory. Táto požiadavka vyplýva z použitia XML parserov jazyku python.

Záver

Výsledkom tejto práce bolo oboznámenie sa s analýzou systémových záznamov a jej využitím pri detekcii vniknutia do systému. Ďalej oboznámenie sa s nástrojom OSSEC, ktorý túto analýzu využíva a odhaľuje tak možné útoky na systém. Poslednou časťou bolo vytvorenie grafického rozhrania pre konfiguráciu nástroja OSSEC, ktorý umožní aj neskúseným užívateľom tento nástroj konfigurovať a tým pádom lepšie využívať.

Do budúcnosti by sa mohli do grafického nástroja pridať rôzni sprievodcovia, ešte viac uľahčujúci nastavenia rôznych funkcií. Ďalej by bolo možné implementovať automatickú aktualizáciu pravidiel z internetu na najnovšie pravidlá. Vďaka použitiu GTK+ knižnice a jazyka python je možné rozšírenie tohto GUI aj pre inštalácie OSSEC na platformu MS Windows alebo iné. A v neposlednej rade môžeme pridať funkcionalitu zabezpečujúcu korektné nastavenie aj pre inštalácie v móde server-agent, nie len local.

Takisto možným vylepšením by mohlo byť vytvorenie Qt klonu, pre užívateľov, ktorý by chceli toto grafické rozhranie využívať natívne v prostredí KDE.

Literatúra

[1] CID, Daniel B. *Log analysis for intrusion detection*. Máj 2006.
Dostupné z: <http://www.ossec.net/en/loganalysis.html>.

[2] CID, Daniel B. *OSSEC Manual*. August 2007.
Dostupné z: <http://www.ossec.net/en/manual.html>.

[3] ROSSUM, Guido van. *Python Reference Manual*. September 2006.
Dostupné z: <http://docs.python.org/ref/ref.html>.

[4] PILGRIM, Mark. *Dive Into Python*. Máj 2004.
Dostupné z: <http://www.diveintopython.org>.

[5] *PyGtk: GTK+ for Python* [online]. Posledná modifikácia: 7. apríla 2006 [cit. 2008-01-10].
Dostupné z: <http://www.pygtk.org>.

[6] *GTK+ - The GIMP Toolkit* [online]. Posledná modifikácia: 28. apríla 2007 [cit. 2008-01-10].
Dostupné z: <http://www.gtk.org>.

[7] *Glade User Interface Builder* [online]. Posledná modifikácia: 18. december 2007
[cit. 2008-01-10].
Dostupné z: <http://glade.gnome.org>.

[8] *Squid: Optimising Web Delivery* [online]. Posledná modifikácia: 21. december 2007
[cit. 2008-01-10].
Dostupné z: <http://www.squid-cache.org>.

[9] *Snort – the de facto standard for intrusion detection/prevention* [online]. Posledná modifikácia:
10. január 2008 [cit. 2008-01-10]
Dostupné z: <http://www.snort.org>.

Zoznam príloh

Príloha 1. CD so zdrojovými textami a manuálom k výslednej aplikácii