OSSEC:
non solo log analysis

Roberto Cecchini - GARR

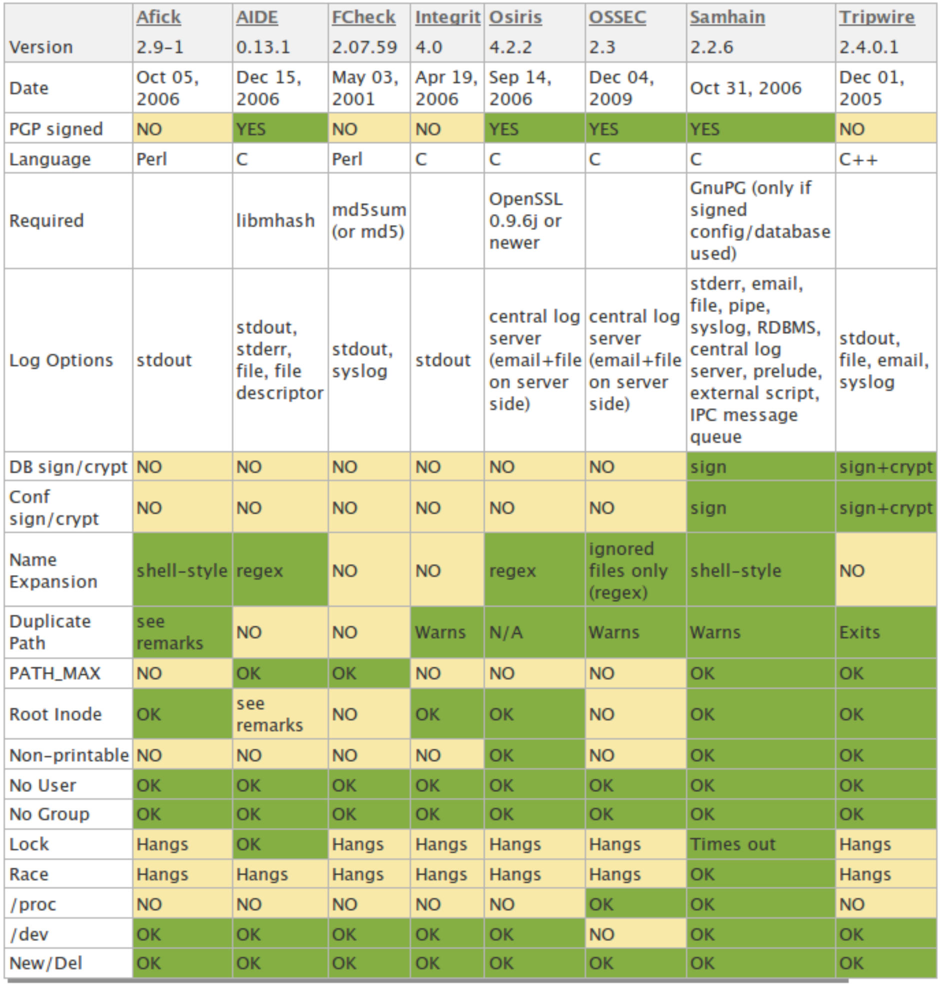
Meccanismi di protezione

- Firewall
- Network Intrusion Detection/Prevention
- Host Intrusion Detection
  - file integrity check
    - funziona anche se l'accesso è stato “regolare”
    - non richiede conoscenze a priori
  - analisi log
    - log server

File integrity check

- Copia di confronto
  - metadata?
- Metadata (mtime, log, ...)
  - modifiche del clock di sistema?
- Checksum ricalcolati periodicamente

Sistemi di integrity check



	Afick	AIDE	FCheck	Integrit	Osiris	OSSEC	Samhain	Tripwire
Version	2.9-1	0.13.1	2.07.59	4.0	4.2.2	2.3	2.2.6	2.4.0.1
Date	Oct 05, 2006	Dec 15, 2006	May 03, 2001	Apr 19, 2006	Sep 14, 2006	Dec 04, 2009	Oct 31, 2006	Dec 01, 2005
PGP signed	NO	YES	NO	NO	YES	YES	YES	NO
Language	Perl	C	Perl	C	C	C	C	C++
Required		libmhash	md5sum (or md5)		OpenSSL 0.9.6j or newer		GnuPG (only if signed config/database used)	
Log Options	stdout	stdout, stderr, file, file descriptor	stdout, syslog	stdout	central log server (email+file on server side)	central log server (email+file on server side)	stderr, email, file, pipe, syslog, RDBMS, central log server, prelude, external script, IPC message queue	stdout, file, email, syslog
DB sign/crypt	NO	NO	NO	NO	NO	NO	sign	sign+crypt
Conf sign/crypt	NO	NO	NO	NO	NO	NO	sign	sign+crypt
Name Expansion	shell-style	regex	NO	NO	regex	ignored files only (regex)	shell-style	NO
Duplicate Path	see remarks	NO	NO	Warns	N/A	Warns	Warns	Exits
PATH_MAX	NO	OK	OK	NO	NO	NO	OK	OK
Root Inode	OK	see remarks	NO	OK	OK	NO	OK	OK
Non-printable	NO	NO	NO	NO	OK	NO	OK	OK
No User	OK	OK	OK	OK	OK	OK	OK	OK
No Group	OK	OK	OK	OK	OK	OK	OK	OK
Lock	Hangs	OK	Hangs	Hangs	Hangs	Hangs	Times out	Hangs
Race	Hangs	Hangs	Hangs	Hangs	Hangs	Hangs	OK	Hangs
/proc	NO	NO	NO	NO	NO	OK	OK	NO
/dev	OK	OK	OK	OK	OK	NO	OK	OK
New/Del	OK	OK	OK	OK	OK	OK	OK	OK

Rainer Wichmann, *A comparison of several host/file integrity monitoring programs*

Perché un loghost?

- Troubleshooting
- (Early) warning
- Forensic data
- Registrazione accessi degli amministratori di sistema
- Unix: **syslog**, **syslog-ng**
- Windows: **eventlog-to-syslog**, **snare**, ecc..

syslog-ng?: yes, please!

- `source`
  - file, pipe, stream/dgram, tcp/udp
- `destination`
  - file, pipe, stream/dgram, tcp/udp, console, program
- `filter`
  - facility, priority, program, host, regexp, filter
- `log`
  - `source + destination + filter`

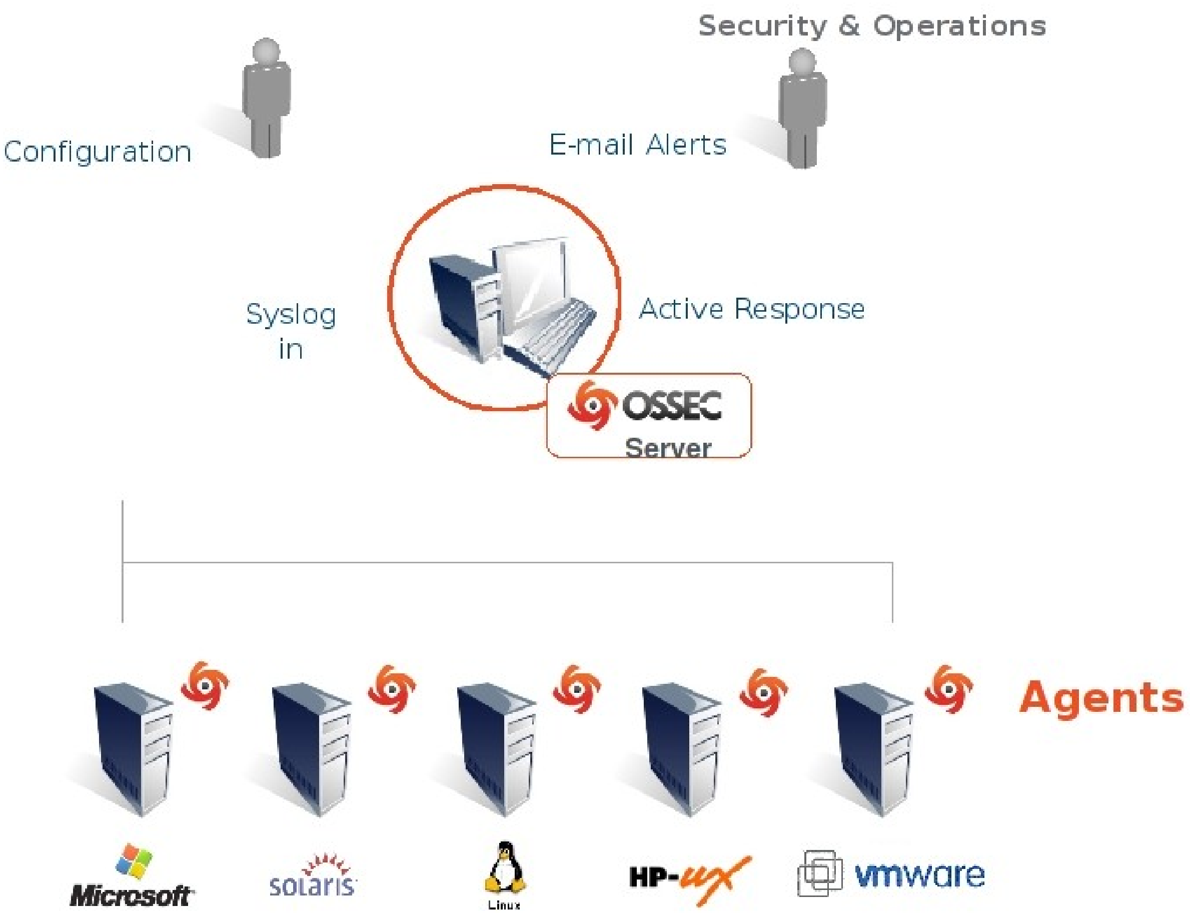
syslog-ng: un esempio

```
source s_loc {
  unix-stream("/dev/log"); internal( ); }
source s_tcpmessages {
   tcp( ip(192.168.190.190); port(10514);); };
destination d_dlog {
   file("/var/log/messages.$WEEKDAY"); };
destination d_mlog {
   file("/var/log/mlog" owner(mick) perm(0600)); };
filter f_mail { facility(mail); };
filter f_messages {
   level(info .. warn) and not
   facility(auth,authpriv,cron,daemon,mail,news); };
log { source(s_tcpmessages); destination(d_mlog); };
log { source(s_loc);
      filter(f_mail); destination(d_mlog); };
log { source(s_loc); filter(f_messages);
      destination(d_dlog); };
```

OSSEC

- Open Source Host Intrusion Detection System
  - Linux, OpenBSD, FreeBSD, OSX, Solaris and Windows (solo client)
- Integrity checking
- Rootkit detection
- Active response
  - whitelist, granulare, timeout
- **Log analysis**

OSSEC: architettura



OSSEC: integrity checking

- File / Directory
  - proprietà
  - permessi
  - checksum
- Windows Registry Monitoring
- File da escludere o ignorare
- Database sul server OSSEC
- Anche agentless

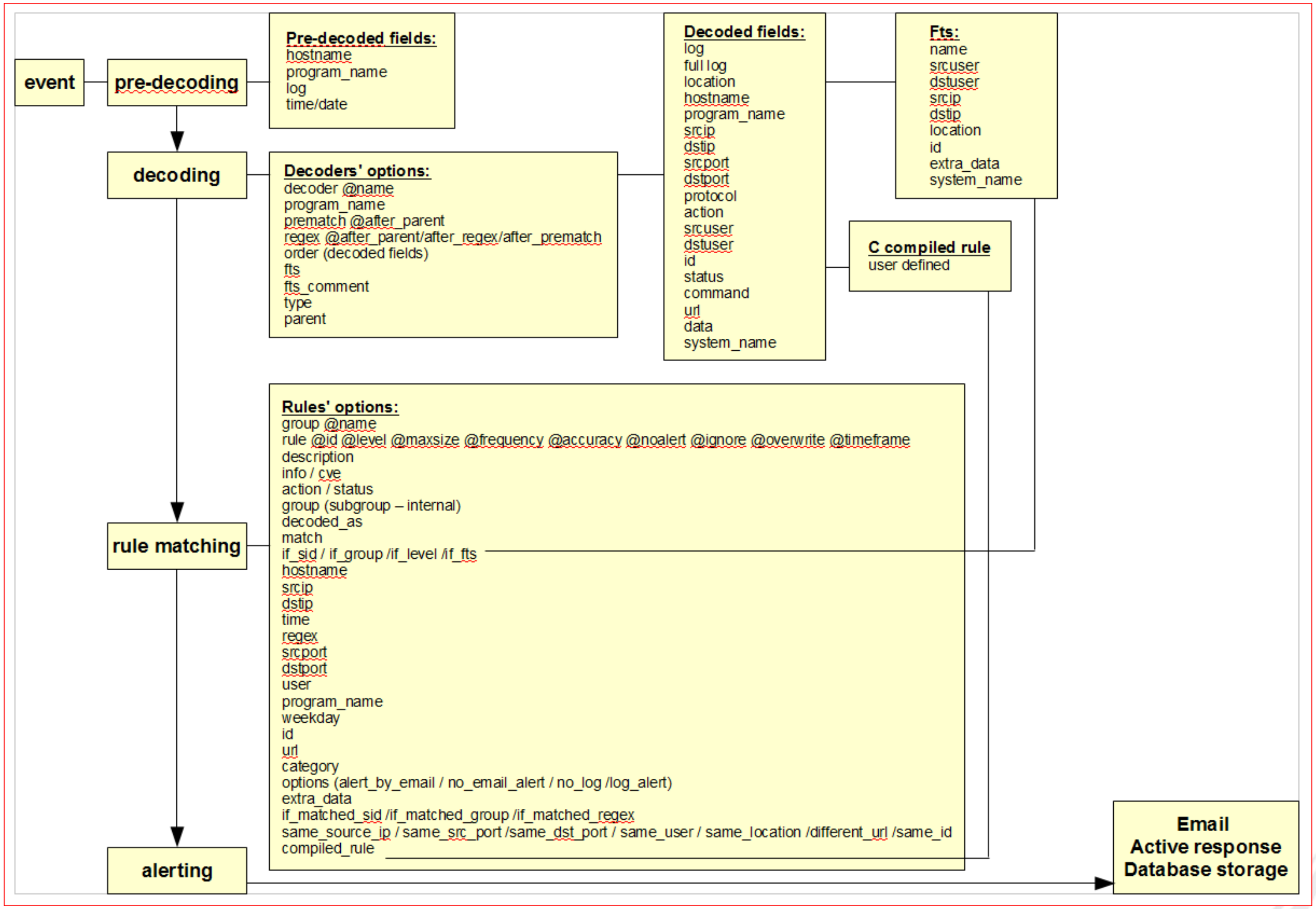
OSSEC: rootkit detection

- DB centrale di signatures
- File Database: file noti di rootkit via stats, fopen e opendir
- Trojan Database: binari usati dai rootkit
- Anomalie del fs: permessi, file di root, file nascosti, file SUID
- Processi nascosti: getsid() vs. ps
- Porte nascoste: bind() vs. netstat
- Interfacce promiscue

OSSEC: log analysis

- Analisi e correlazione di file di log
- Regole flessibili (xml)
- Molte regole preesistenti (> 400)
  - unix pam, sshd, telnetd, samba, su, sudo, proftpd, pure-ftpd, vsftpd, MS ftp server, solaris ftpd, imapd, postfix, sendmail, vpopmail, MS exchange, apache, IIS5/6, Horde IMP, iptables, pf, netscreen, Cisco PIX/ASA/FWSM, snort, Cisco IOS, nmap, Symantec AV, arpwatch, named, squid, Windows event log, ecc., ecc.

Log analysis



event

pre-decoding

Pre-decoded fields:
hostname
program_name
log
time/date

decoding

Decoders' options:
decoder @name
program_name
prematch @after_parent
regex @after_parent/after_regex/after_prematch
order (decoded fields)
fts
fts_comment
type
parent

Decoded fields:
log
full log
location
hostname
program_name
srcip
dstip
srcport
dstport
protocol
action
srcuser
dstuser
id
status
command
url
data
system_name

Fts:
name
srcuser
dstuser
srcip
dstip
location
id
extra_data
system_name

C compiled rule
user defined

rule matching

Rules' options:
group @name
rule @id @level @maxsize @frequency @accuracy @noalert @ignore @overwrite @timeframe
description
info / cve
action / status
group (subgroup – internal)
decoded_as
match
if_sid / if_group /if_level /if_fts
hostname
srcip
dstip
time
regex
srcport
dstport
user
program_name
weekday
id
url
category
options (alert_by_email / no_email_alert / no_log /log_alert)
extra_data
if_matched_sid /if_matched_group /if_matched_regex
same_source_ip / same_src_port /same_dst_port / same_user / same_location /different_url /same_id
compiled_rule

alerting

Email
Active response
Database storage

Aurora Mazzone, http://personalpages.to.infn.it/~mazzone/ossec/

Log flow

- Pre-decoding
  - estrae campi noti
- Decoding
  - anche con regole utente
- Rule matching
  - anche con regole utente
- Alerting
- Active response

Pre-decoding

```
Apr 14 17:32:06 ehi sshd[1025]: Accepted password
for root from 192.168.1.1 port 1618 ssh2
```

- time/date → Apr 14 17:32:06
- hostname → ehi
- program_name → sshd
- log → Accepted password for root from 192.168.1.1 port 1618 ssh2

Decoding

```
Apr 14 17:32:06 ehi sshd[1025]: Accepted password
for root from 192.168.1.1 port 1618 ssh2
```

- time/date → Apr 14 17:32:06
- hostname → ehi
- program_name → sshd
- log → Accepted password for root from 192.168.1.1 port 1618 ssh2
- srcip → 192.168.1.1
- user → root

Esempio di decoder

```
192.168.1.190 - - [18/Jan/2006:13:10:06 -0500]
"GET /index.html HTTP/1.1" 200 1732
```

```
<decoder name="web-accesslog">
  <type>web-log</type>
  <prematch>^\d+.\d+.\d+.\d+ </prematch>
  <regex>^(\d+.\d+.\d+.\d+) \S+ \S+ [\S+ \S\d+]</regex>
  <regex>"\w+ (\S+) HTTP\S+ (\d+) </regex>
  <order>srcip, url, id</order>
</decoder>
```

Regole

- XML
- Match in base alle informazioni decodificate
- Oltre 400 regole preinstallate
- Due tipi
  - singole
  - composite

Esempio: tentativi di login

```
<rule id="5700" level="0" noalert="1">
  <decoded_as>sshd</decoded_as>
  <description>SSHD messages grouped.</description>
</rule>

<rule id="122" level="7">
  <if_sid>5700</if_sid>
  <match>^Failed password</match>
  <description>Failed password</description>
</rule>

<rule id="133" level="13">
  <if_sid>122</if_sid>
  <hostname>^mainserver</hostname>
  <srcip>!192.168.2.0/24</srcip>
  <description>Tentativo su mainserver!</description>
</rule>
```

Esempio: login fuori orario

```
<rule id="153" level="5">
  <if_sid>5700</if_sid>
  <match>Accepted password</match>
  <description>Login ok</description>
  <group>login_ok</group>
</rule>

<rule id="154" level="10">
  <if_sid>153</if_sid>
  <time>6 pm – 8:30 am</time>
  <description>Login fuori orario
d'ufficio</description>
  <group>login_ok policy_violation</group>
</rule>
```

Esempio: login multipli

```
<rule id="133" level="7">
  <if_sid>5700</if_sid>
  <match>^Failed password</match>
  <description>Failed password attempt</description>
</rule>
```

```
<rule id="1050" level="11" frequency="5"
timeframe="120">
  <if_matched_sid>133</if_matched_sid>
  <same_source_ip />
  <description>Tentativi multipli dallo stesso ip!
</description>
</rule>
```

Alert via mail

```
OSSEC HIDS Notification.
2010 Apr 20 01:21:17

Received From: ercole->/var/log/secure
Rule: 5551 fired (level 10) -> "Multiple failed logins in a small period of time."

Portion of the log(s):

Apr 20 01:21:16 ercole sshd[5423]: (pam_unix) authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=174.121.213.181
Apr 20 01:21:16 ercole sshd[5414]: (pam_unix) authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=174.121.213.181  user=root
Apr 20 01:21:16 ercole sshd[5413]: (pam_unix) authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=174.121.213.181
Apr 20 01:21:16 ercole sshd[5421]: (pam_unix) authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=174.121.213.181
Apr 20 01:21:16 ercole sshd[5424]: (pam_unix) authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=174.121.213.181
Apr 20 01:21:16 ercole sshd[5422]: (pam_unix) authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost=174.121.213.181  user=root
```

Bibliografia

- *Misure e accorgimenti prescritti ai titolari dei trattamenti effettuati con strumenti elettronici relativamente alle attribuzioni delle funzioni di amministratore di sistema - 27 novembre 2008* (G.U. n. 300 del 24 dicembre 2008) http://j.mp/9SsI0b
- CERT-In, *Implementation of Central Logging server using syslog-ng*, http://j.mp/a2BG1B
- Rainer Wichmann, *A comparison of several host/file integrity monitoring programs* http://www.la-samhna.de/library/scanners.html
- http://www.loganalysis.org/
- **eventlog-to-syslog:** http://code.google.com/p/eventlog-to-syslog/
- **Snare:** http://www.intersectalliance.com/projects/SnareWindows/
- **OSSEC:** http://www.ossec.net/
- Andrew Hay, Daniel Cid e Rory Bray, *OSSEC Host-Based Intrusion Detection Guide*, Syngress (2008)
- Aurora Mazzone, http://personalpages.to.infn.it/~mazzone/ossec/