Automated detection and response to attacks

OSSEC

Learn how to monitor and block attacks without lifting a finger.

BY KURT SEIFRIED

One of the first things I learned about computer security was logging [1]. If you don't have logs, then trying to reconstruct what happened when something breaks, or when you get broken into, is almost impossible. The second thing I learned was that you have to centralize your logging; this is the only way to get a complete picture and ensure that an attacker can't simply wipe the logs on a compromised host, leaving you nothing to work with. But none of this will alert you to an attacker or, even more importantly, stop an attacker from getting in. It will simply give you something to look at once you figure out you have been broken into. For this, you need a human being in the loop, right? Well, you either need a human being or some smart software.

Wouldn't it be great if you could monitor critical logfiles (like mail and web) and actually have something respond to attacks, notifying you and even blocking the attacker from further access if you so wished? Well you're not the only one. Daniel B. Cid is the lead developer of the OSSEC project, an effort to build an open source host-based intrusion detection system [2]. OSSEC uses a traditional server and agent approach: You install the agent software on each system you want to monitor, and a central server collects all the data and sends out alerts. Additionally, the OSSEC project has released a web-based interface; however, it is only capable of reporting. Unfortunately, it can't be used to configure the system.

Installing OSSEC

When installing OSSEC, you have three options. The server option allows you to have it monitor itself and collect alerts from other systems. The agent option simply monitors local events and fires anything interesting off to the server. The local option runs the monitoring locally and can send email alerts, but it does not listen for any remote agents (so if you have one server or want to test it, this is the option for you). Simply download the OSSEC package (*ossec-hids-2.0.tar.gz*") and unpack it to a directory:

```
# wget http://www.ossec.net/files/ossec-hids-2.0.tar.gz
```

```
# tar -zxf ossec-hids-2.0.tar.gz
# cd ossec-hids-2.0
# ./install.sh
```

Now you simply choose your language, your server type, and whether you want to run the integrity check daemon, run the rootkit detection engine, enable active response, and enable the firewall to block attacks. If you are setting the system up as an agent, you also need to point it to your server and paste in the agent key. The agent key is a long string used to secure communications between an agent and the server, preventing fake messages from being injected, and so on. Why is it important to prevent spoofed or fake messages from being sent to the server?

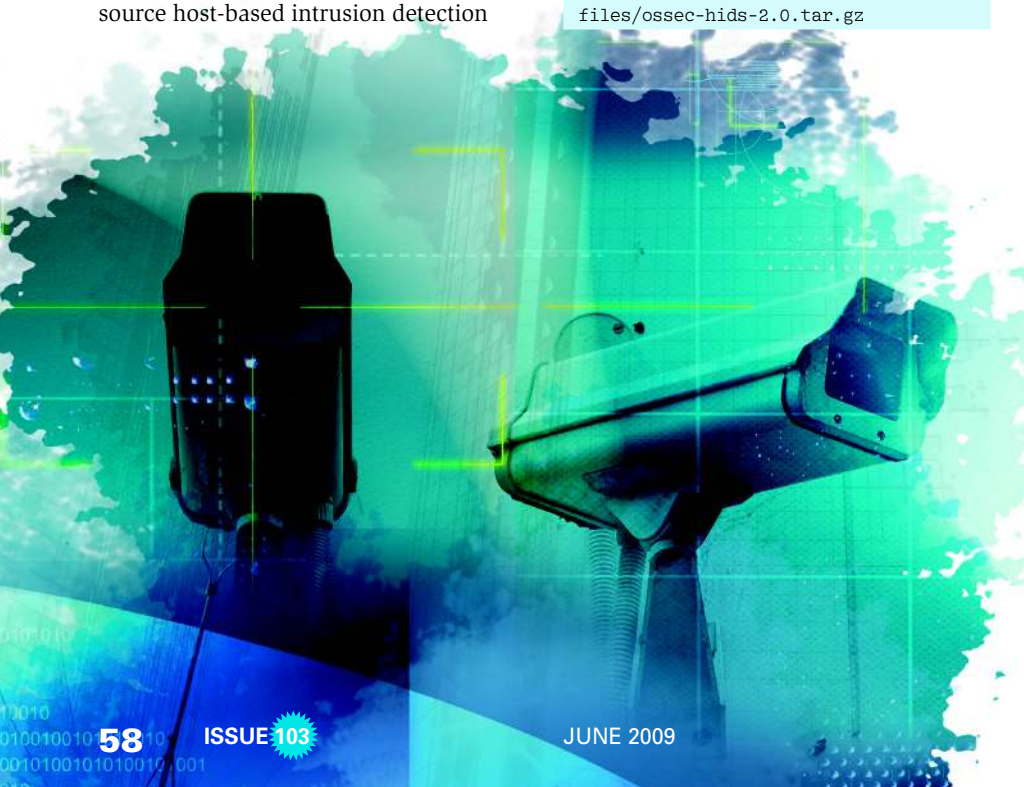
Beware

If an attacker can trigger fake or spoofed attacks and a system blocks IP addresses or users because of this, the attacker can easily block legitimate systems and lock users out. In a worst case scenario, you might have to break into your own system if your accounts are locked out, which is why most HIDS and NIDS support whitelisting (see the "HIDS vs. NIDS" box). Administrators simply create a list of hosts and networks that are critical. Of course, determining which hosts are critical depends on the exact setup (DNS, email, file servers, authentication servers, routers, etc. are all a good place to start). With OSSEC, the whitelist is held in the *ossec.conf* file (by default, this is kept in */var/ossec/etc/*), and you can specify individual hosts or networks:

```
<global>
 <white_list>127.0.0.1</white_list>
 <white_list>1.2.3.4</white_list>
 <white_list>10.0.0.0/8</white_list>
 <white_list>192.168.0.0/16
 </white_list>
</global>
```

Running OSSEC

The OSSEC program comes with its own control program called *ossec-control*. Additionally, when installed on Red Hat Linux or CentOS, a standard set of *rc.d/init* scripts will be added, allowing the OSSEC services to be control through the standard *chkconfig* utility. When OSSEC is running, you should see a number of programs running.



The monitoring processes generally need to run as root:

```
USER   PID   COMMAND
ossecm 17381 /var/ossec/bin/ossec-maild
root   17385 /var/ossec/bin/ossec-execd
ossec  17389 /var/ossec/bin/➥
ossec-analysisd
root   17393 /var/ossec/bin/➥
ossec-logcollector
root   17405 /var/ossec/bin/➥
ossec-syscheckd
ossec  17409 /var/ossec/bin/➥
ossec-monitord
```

OSSEC Agent

Once you have the server running, it's high time to get the rest of your herd reporting to it. Simply install the OSSEC software on any machines you want to monitor, choosing the agent installation option, of course.

During the install, you will be asked for the IP address of the server and standard options regarding which monitoring options you want. Once you have finished, you will need to create and import the agent key, which is done via the *manage_agents* program. On the server you simply add the agent.

Once finished you can extract the key for a particular agent, then you will need to cut and paste it (remote login via SSH is your best bet). Simply run *manage_agents* on the agent and import the key. The process is similar for Windows, but a graphical interface has been added as the default to make it easier (fortunately, the command-line versions of all the programs are available, which allows scripted management to be done remotely via the command line).

By default, OSSEC monitors all files in */etc*, */bin*, */sbin*, */usr/bin*, and */usr/sbin* (essentially the guts of almost any system) and a large number of network daemon logfiles (*named*, *smbd*, *mysql*, *telnetd*, etc.).

To modify which directories are monitored or to add new rulesets for monitoring services, you simply edit the *ossec.conf* file, which uses an XML-style format that is largely self-explanatory.

OSSEC WebUI

So now that you have OSSEC properly set up and it's protecting your network, what do you do now? One feature I love about OSSEC is the reporting. For example, you can generate text reports on the top activity for IP addresses, attempted login names, and so on.

Of course, a text-based report is unlikely to impress your boss; fortunately, there is a solution for this. The web user interface for OSSEC allows ad hoc queries, but unfortunately, it does not support configuration of the server or agents (for that, you have to stick to the command line).

Additionally, OSSEC WebUI allows you to see the state of your server and agents at a glance (Figure 1).

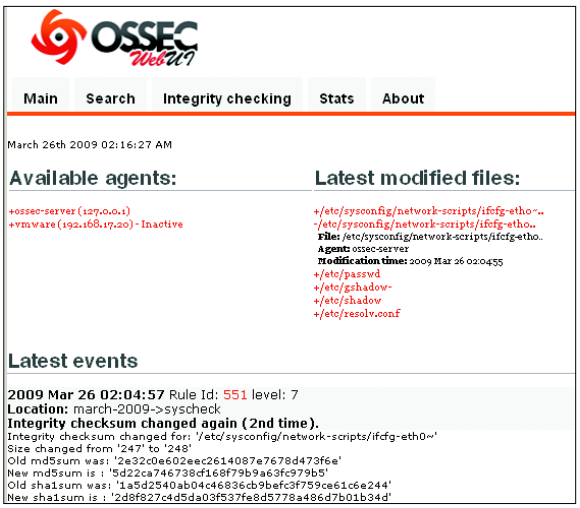


Figure 1: The Main tab of the OSSEC web interface shows some information about the latest modified files and events.

Tripwire

Of course, I would be amiss if I failed to mention Tripwire [3]. Tripwire is the granddaddy of HIDS, monitoring and reporting on file changes on Unix systems (and now on Windows), routers, and other devices.

Tripwire is still available as an open source package; however, it has not been updated in several years (although one could argue it is largely a finished project).

Conclusion

One of the biggest problems with security is the amount of setup effort and continuous maintenance it often requires. OSSEC provides a degree of assurance and active protection with a minimal setup cost and little maintenance. OSSEC is lacking in a few features I would really love to see (like telling me what changed within a file as opposed to just telling me that the file has changed) and lacks some ease of use features (like mass configuration and change management), but weighed against the simplicity of setup and management I think it's still worth it. ■

HIDS vs. NIDS

Host-based intrusion detection systems (HIDS) are generally defined as applications that run on specific systems and monitor local logfiles, inbound network activity, and other items to detect hostile behavior. The advantage of HIDS is that it has deeper access to a system and can correlate local events easily (e.g., a web application error followed by a new user being added). The disadvantage of HIDS is that you must install software on each system you want to protect and manage many endpoints.

Network intrusion detection systems (NIDS) typically consist of one or more network-based sensors deployed at network choke points (such as firewalls) or attached to switches that are configured to replicate traffic to the sensor. The advantage of NIDS is that you can cover large portions of a network and network traffic with a minimal number of sensors. The disadvantage of NIDS is that you could miss internal attacks that don't cross monitored networks, and you can't see deeply into a system.

INFO

[1] "Dive Deep" by Heike Jurzik, *Linux Pro Magazine*, April 2008, *http://www.linux-magazine.com/w3/issue/89/086-087_command.pdf*

[2] OSSEC: *http://www.ossec.net/*

[3] Tripwire: *http://sourceforge.net/projects/tripwire/*

THE AUTHOR

Kurt Seifried is an Information Security Consultant specializing in Linux and networks since 1996. He often wonders how it is that technology works on a large scale but often fails on a small scale.