Logging and processing logs from Windows 7

Timber!

Windows 7 is pretty good at logging, but what do you do with all those log files? We look at some monitoring tools that can help you get the most out your logging data.

By Kurt Seifried

This month, I'm talking about Windows 7. I must admit I spent some time trying to come up with a good security topic related to Windows 7 that I haven't covered before. I've already done cross-platform host-based intrusion detection systems (OSSEC [1]), and I'll leave IPsec setup with Windows 7 and Linux to someone else. I tried to figure that out once, but gave up and went with a Shrew Soft IPsec client instead [2].

So, what to write about? Logging! The good news is that Windows 7 (and Windows in general since Windows 2000) has pretty good logging facilities. By default, it will catch most security-related events. And, if you're paranoid, you can enable logging for successful or failed attempts to access certain directories and files, run programs, and so on – not to mention all the third-party applications you need on Windows 7 machines (antivirus, etc.) that can also log events locally. But log files sitting on a remote system are a lot like trees that fall in a forest with no one to hear: basically useless.

Windows syslog Clients

I'll start by collecting all those Windows logs centrally onto a Unix system. The good news is that syslog [3] (and its replacements rsyslog and syslog-ng) won the standards battle, and almost every remote logging product supports the syslog protocol. Dozens of Windows syslog clients and several free clients are available: NTsyslog [4] (last updated 2009), eventlog-to-syslog [5], Snare [6], and Datagram SyslogAgent [7] (Figure 1).

Generally speaking, these clients work, but they do not offer much in the way of features beyond logging to a syslog server via UDP (no SSL encryption) and possibly mirroring (sending logs to more than one server, which is good in case one goes down). Thus, it's important to remember that by allowing remote UDP logging you are opening up your syslog server to spoofed messages, so you should restrict network access to it as much as possible and use ingress filtering to avoid spoofed network data.

Getting Unix systems to log to a central syslog server is pretty trivial; just specify a line containing something like `*.* @192.168.0.1`, and you're done. You can, of course, get fancy and use SSL/etc. if you want.

Performance and Databases

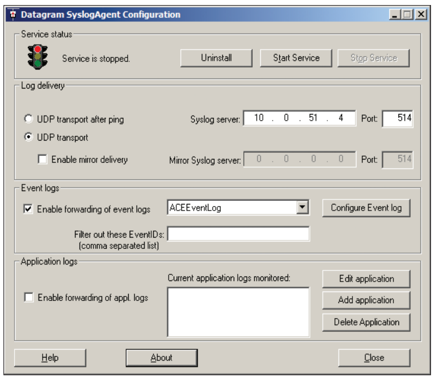
Now you have multiple systems all logging centrally to one or more big machines. For performance reasons, or for dealing with remote locations that may have spotty network connectivity, you could create multiple syslog servers. One quick note on performance: Syslog servers create heavy disk write loads (essentially you're churning out a huge text file). The easiest way to address this issue is to use a solid state disk or a PCIe card with solid state memory.

For a few hundred dollars, you can get something that will handle several hundred IOPS (Input/Output Operations per Second) with high writing capacity, and for a few thousand dollars, you can easily get something that will handle hundreds of thousands of IOPS. If you're using text-based logs, just rotate the logs often enough so that the device never gets full. But, what if you have multiple servers or want to actually use the data at a later date? You can put it into a database, of course. Rsyslog comes with the ability to log to a database, as does syslog-ng. The further advantage here is that you can build relatively low-end syslog servers and have a single dedicated high-end database server to collect all the logs.

With rsyslog, installation and configuration largely consists of installing the rsyslog-mysql package (or rsyslog-pgsql package, if you prefer PostgreSQL). Once that's done, you can log into the database and add a user and tables to hold the data (consult the `createDB.sql` file for the commands). In regard to performance with MySQL: Use InnoDB, or it will all go sideways. The default MyISAM table type uses table-level locking

KURT SEIFRIED

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.


Datagram SyslogAgent Configuration

Service status
Service is stopped.
Uninstall | Start Service | Stop Service

Log delivery
( ) UDP transport after ping
(•) UDP transport
[ ] Enable mirror delivery
Syslog server: 10 . 0 . 51 . 4 Port: 514
Mirror Syslog server: 0 . 0 . 0 . 0 Port: 514

Event logs
[✓] Enable forwarding of event logs ACEEventLog
Configure Event log
Filter out these EventIDs:
(comma separated list)

Application logs
[ ] Enable forwarding of appl. logs
Current application logs monitored:
Edit application | Add application | Delete Application

Help | About | Close

Figure 1: Installation and configuration of SyslogAgent on Windows 7.

for inserts, meaning that every time you insert a record, the entire database gets locked. Thus, if a number of events are being logged (even on a single host, you can easily generate several syslog messages per second), the table will be pretty much constantly locked for writing. By contrast, the InnoDB type uses row-level locking and will handle higher levels of inserts much more gracefully. Once you've created the database table, you can simply add

```
$ModLoad ommysql
*.* :ommysql:127.0.0.1,Syslog,↩
   Syslog,password
```

and then restart rsyslog [8].

What About All This Data?

Beyond setting up central logging so that attackers can't cover their tracks by simply wiping the logs on a local machine, what can you do with all this data? A number of tools are available to monitor syslog data, and they all fall into essentially two groups: "real-time" monitoring and "near real-time" or "after the fact" monitoring.

An example of an "after the fact" monitoring tool would be logwatch. Logwatch is run periodically (usually daily). It parses the syslog log files looking for suspicious events, such as failed logins, and then sends out a warning email on the basis of these events.

Although nothing is inherently wrong with logwatch, by definition, it's pretty much too late to do anything by the time the tool informs you of an event. On a daily schedule, the bad guy may have broken in 23 hours before you got the warning.

Real-Time Monitoring

The two best examples of real-time syslog monitoring are fail2ban and Sagan. Fail2ban was covered in a previous article [9]: The tool watches for failed login attempts, which indicate either a brute force attack or a drunk user (either way, not someone that should be logging in). Then, it bans that system from further access either temporarily or permanently, thereby reducing the chances of a bad guy getting in.

Sagan

Sagan [10] is named after a dog, who in turn was named after Carl Sagan, and is licensed under the GPLv2 license. Installation of Sagan is pretty easy. It's your typical `./configure; make; make install`, with the only caveat being that you'll need several external dependencies, such as pcre-devel, libpcap-devel, libesmtp-devel, and libdnet-devel. Additionally, on Red Hat-based systems (CentOS, Fedora, Scientific Linux, etc.), the MySQL client libraries will not be found, so you can use the option `--with-mysql-libraries=/usr/lib64/mysql/` (or wherever your `libmysqlclient_r.so` file is).

Once the tool has been installed, you'll need to configure Sagan and download the rules. The `sagan.conf` file is self-explanatory; however, if you want to monitor the log messages (especially with rsyslog) as they come in, you will need to create a FIFO (First In, First Out) buffer, into which your syslog server will write all incoming messages, which then get picked up by Sagan. You can start by adding a user called "sagan":

```
adduser sagan
mkfifo /var/run/sagan.fifo
chown sagan:sagan /var/run/sagan.fifo
```

You'll then need to download and install the latest rules [11] and unpack them into the *`RULE_PATH`* directory (as specified in `sagan.conf`).

One thing I really like about Sagan is that it can log events into your Snort database. That means you can log network- and host-based events across all your systems in one interface if you already use Snort. Sagan also supports Prelude (a large-scale security monitoring framework supported by many other products) and Logzilla (which, sadly, is no longer free software). Finally, you can configure Sagan to send email alerts. However, I advise you do so carefully, or you may end up with a flood of email every time someone does a brute force login attempt, for example.

A Note on Time Synchronization

Centralized logging has one advantage, in that the time a message came in is (roughly) when the event happened, making time synchronization slightly less important. However, if you need to compare log entries from multiple systems that are not synchronized, you'll be in a world of hurt. Also, messages may be delayed and queued for later delivery, and if this happens, time synchronization is critical. So, I recommend installing NTP and configuring it on all your systems – it's just a good idea. ■■■

INFO

[1] "Automated detection and response to attacks – OSSEC" by Kurt Seifried, *Linux Magazine*, June 2009: *http://www.linuxpromagazine.com/Issues/2009/103/Security-Lessons/%28kategorie%29/0*

[2] Shrew Soft: *http://www.shrew.net/*

[3] "Secure logging" by Kurt Seifried, *Linux Magazine*, June 2010: *http://www.linuxpromagazine.com/Issues/2010/115/SECURE-LOGGING/%28kategorie%29/0*

[4] NTsyslog: *http://sourceforge.net/projects/ntsyslog/*

[5] Eventlog-to-syslog: *http://code.google.com/p/eventlog-to-syslog/downloads/list*

[6] Snare: *http://www.intersectalliance.com/projects/SnareWindows/*

[7] Datagram SyslogAgent: *http://syslogserver.com/syslogagent.html*

[8] rsyslog-mysql: *http://www.rsyslog.com/doc/rsyslog_mysql.html*

[9] "Fighting dictionary attacks with Sshutout and Fail2ban" by Charly Kühnast, *Linux Magazine*, June 2008: *http://www.linuxpromagazine.com/Issues/2008/91/Sshutout-and-Fail2ban/%28kategorie%29/0*

[10] Sagan: *http://sagan.softwink.com/*

[11] Sagan rules: *http://sagan.softwink.com/rules/*