

# Event Log Management: Demos

Presented by  
**Anthony Lai**

# Agenda

- Demonstrations
  - Logs from web attack with WebGoat
    - Apache Web Server Log Configuration & Possible Exploits
    - You can't simply ignore it. Check your web server!
  - Logs filtering with LogParser
    - I am tired to review thousands of log entries, what can I do?
- Role & Responsibility (R&R) of logs review
  - There are many devices and servers to monitor....
- Logs review best practices and its challenges

# What will you see from web server logs if the web application is under attack?

- Let me show you a simple demonstration with WebGoat, which is a standalone web site for people to testing web hacking.
- It is released from OWASP.
- Please be liable for your unethical behavior. Do not try to “test” another company’s logging mechanism or fault detection with your hacking skills!☺

# Demonstration with WebGoat

- Apart from WebGoat, it needs another tool called WebScarab to act as a proxy between the browser and the targetted web site?
- Why? You will know it later ☺



## After providing demos.....

- What do you think? Have you checked your web server logs?
- I would like to provide information about web server logging configuration. I pick up Apache.
- It is a popular web server running on both Win32 and Linux/Unix platforms.
- Web server could be accessed in any time and any place.

## Apache Web Server Log Configuration (1)

- I will give an overview of how to configure log files in Apache. Remember that this is not a comprehensive explanation, and for more information you should look at Apache's official documentation: <http://httpd.apache.org/docs/logs.html>.
- *Normal (Classic) Configuration* There are two types of log information in Apache: the *access log* (handled by the module `mod_log_config`) and the *error log*.
- The access log records every request sent to the web server. A typical configuration is:  

```
LogFormat "%h %l %u %t \"%r\" %<s %b" common
CustomLog logs/access_log common
```

## Apache Web Server Log Configuration (2)

- Apache is instructed to log access information in the file `logs/access_log`, using the format defined in the previous line (`common`). To find out the exact meaning of each parameter, check Apache's documentation. You will find out that Apache can log almost anything pertaining to a request, including the client's address and the type of request itself.
- Apache server's error messages are logged separately, using a different file. In this case, there is no definite format for the messages, and these directives are defined:  

```
ErrorLog logs/error_log
LogLevel warn
```

## Error Levels

- The first directive, `ErrorLog`, instructs Apache to log all the errors in `logs/error_log`. The second directive sets the minimum importance for a message to be logged (the "level" of the message).
- Remember that if you decide to set the log level to `crit`, the messages for more important levels will be logged as well (in this case, `alert` and `emerg`).
- **NOTE** Notice level messages are *always* logged, regardless of the `LogLevel` setting.

Level	Description
Emerg	Emergencies - system is unusable
alert	Action must be taken immediately
Crit	Critical Conditions
Error	Error conditions
Warn	Warning conditions
Notice	Normal but significant condition
Info	Informational
Debug	Debug-level messages

# Apache Log Format

%h	Logs the remote host
%l	Remote logname, if supplied
%u	Remote user (mostly useful if logging behind authentication)
%t	The date and time of the request
%r	The request to your web site
%s	The status of the request (201, 301, 404, 500, etc.), the > in front of the "s" insures only the last status is logged.
%b	Bytes sent for the request (tracks http bandwidth use)
%i	Tracks items sent in the HTML header. So by adding (Referer) and (User Agent), we are capturing the referring url and the browser type in the combined log format.

# Apache Logging Practice

Logging appears to be a simple process, and you might wonder why security is involved at all. There are some very basic security problems connected to logging. For example:

- Logs are written as root, and permission problems can be dangerous. (\*)
- Logs are written in plain text, and can be easily modified and forged.
- Logging programs are executed as root; if they are vulnerable, an attacker may gain root access.
- Logs can cause a DOS if they run out of disk space (an attacker might do this deliberately).
- Logging can be unreliable; if Apache dies (for example after an attack), they could be incomplete.

# Logs and Root Permissions

- Apache is normally started by the root user, in order to be able to listen to **port 80** (non-root processes can only listen to ports higher than 1024). After starting up, Apache opens the log files, and only *then* drops its privileges. This allows the Apache server to write to files that no other user may access (if the permissions are set properly), protecting the log files. If the log files were opened after dropping privileges, they would be a lot more vulnerable.
- This implies that if the directory where the logs are stored is writable by common users, then an attacker can do this (note the wrong permissions for the logs directory).

```
[merc@localhost merc]$ cd /usr/local/apache2/
[merc@localhost apache2]$ ls -l
total 52
drwxr-xr-x 2 root root 4096 Oct 4 14:50 bin
drwxr-xr-x 2 root root 4096 Sep 13 23:18 build
drwxr-xrwx 2 root root 4096 Oct 5 18:10 logs
[...]
drwxr-xr-x 2 root root 4096 Oct 4 18:50 modules
[merc@localhost apache2]$ cd logs
[merc@localhost logs]$ ls -l
total 212
-rw-r--r--1 root root 124235 Oct 5 18:11 access_log
-rw-r--r--1 root root 74883 Oct 5 18:10 error_log
-rw-r--r--1 root root 5 Oct 5 18:10 httpd.pid
[merc@localhost logs]$ rm access_log
rm: remove write-protected file 'access_log'? y
[merc@localhost logs]$ ln -s /etc/passwd_for_example
access_log
[merc@localhost logs]$ ls -l
total 84
lrwxrwxrwx 1 merc merc 23 Oct 5 19:26 access_log ->
/etc/passwd_for_example
-rw-r--r--1 root root 75335 Oct 5 19:27 error_log
-rw-r--r--1 root root 5 Oct 5 19:27 httpd.pid
[merc@localhost logs]$
```

## The next time Apache is run...

- the web server will append to `/etc/passwd`. This would make the system unstable and prevent any further login by users. The solution is to ensure that the logs directory is not writable by other users. Obviously, this can only be done if the attacker has login access to the web server.

## Error Log in Apache

- An ideal error log on a running server is an empty one (apart from information about the server starting and stopping), when the error level is set to notice. For example, a "File not Found" error probably means that there is a broken link somewhere on the Internet pointing to your web site. In this case, you would see a log entry like this:  

```
[Sat Oct 05 20:05:28 2003] [error] [client 127.0.0.1] File does not exist: /home/merc/public_html/b.html, referer: http://localhost/~merc/a.html
```
- The webmaster of the referrer site should be advised that there is a broken link on their site. If there is no answer, you might want to configure your Apache server so that the broken link is redirected to the right page (or, if in doubt, to your home page).

## Looking for exploits!

- If crackers are looking for possible exploits, they will generate "File not Found" entries in the error log, so keeping the error log as clean as possible will help to locate malicious requests more easily. Some exploit attempts are logged in the error\_log. For instance, you could find:  

```
[merc@localhost httpd]$ grep -i formmail access_log [Sun Sep 29 06:16:00 2003] [error] [client 66.50.34.7] script not found or unable to stat: /extra/httpd/cgi-bin/formmail.pl [merc@localhost httpd]$
```
- The formmail script is widely used, but it generates a number of security issues.

## DOS Attack on Apache?

- A segmentation fault problem needs attention as well. Apache should never die, unless there is a problem in one of the modules or an attack has been performed against the server. Here is an example:  

```
[Sun Sep 29 06:16:00 2002] [error] [notice] child pid 1772 exit signal Segmentation fault (11)
```
- If you see such a line in the log file, you will have to see what was going on at the time in the server's activity (possibly reading the access\_log file as well) and consider upgrading Apache and its modules as soon as possible. Because of Apache's extensive use and deployment, most such problems in the core Apache package have been eliminated. Therefore, a segmentation fault message usually implicates an after-market or third-party module failure, or a successful DOS attack.

## Access log in Apache

- The access log includes information about what the user requested. If the error log reports a segmentation fault, you can use the access log to find out what caused Apache to die. Remember that if the cause of death is really sudden, because of buffering issues, the latest log information might not be in the log file.
- You can also use the access log to check whether someone is trying to break into your system. Some attacks are easy to identify by checking for the right string in the log. You can find the entries for many Windows-aimed attacks just by looking for the exe string in the access log. For example:

```
[root@localhost logs]# grep -i exe access_log
200.216.141.59 - - [29/Sep/2003:06:25:22 +0200] "GET
/_vti_bin/shtml.exe HTTP/1.0" 404 288
200.216.141.59 - - [29/Sep/2003:06:31:33 +0200] "GET
/_vti_bin/shtml.exe HTTP/1.0" 404 288
193.253.252.93 - - [02/Oct/2003:02:17:53 +0200] "GET
/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
HTTP/1.1" 404 319
151.4.241.194 - - [02/Oct/2002:02:34:46 +0200] "GET
/scripts/..%255c%255c../winnt/system32/cmd.exe?/c+dir" 404 -
[root@localhost logs]#
```

## Encoded URL found from the log?!

- The main problem with using grep to look for attacks: URLs can be URL-encoded (see Appendix B for more information). This means that the last entry you saw in the access\_log shown above could be written as:

```
151.4.241.194 - -
[02/Oct/2003:02:34:46 +0200] "GET
/scripts/..%255c%255c../winnt/system
32/cmd.%65x%65?/c+dir" 404 -
```

## A Simple Script to Use As a Filter

```
#!/usr/bin/perl
use URI::Escape;
use strict;
# Declare some variables
#
my($space)="%20";
my($str,$result);
# The cycle that reads
# the standard input
while(<>){
# The URL is split, so that you have the
# actual PATH and the query string in two
# different variables. If you have
# http://www.site.com/cgi-bin/go.pl?query=this,
# $path = http://www.site.com/cgi-bin/go.pl
# $qstring = "query=this"
my ($path, $qstring) = split(/\?/, $_, 2);

# If there is no query string, the result string
# will be the path...
$result = $path;
# ..BUT! If the query string is not empty, it needs
# some processing so that the "+" becomes "%20"!
if($qstring ne ""){
    $qstring =~ s/\+/$space/ego;
    $result .= "?$qstring";
}
# The string is finally unescaped...
$str = uri_unescape($result);
# ...and printed!
print($str);
}
```

## Search for decoded URL...

- Note that the script is slightly complicated by the fact that a + (plus) in the query string (and *only* in the query string) must be converted into %20 (\$qstring =~ s/\+/\$space/ego;), which is then translated into a space once the string is URL-decoded:
 

```
$str = uri_unescape($result);
```
- You should call this script urldecode, place it in /usr/local/bin, and give it executable permission (chmod 755 /usr/local/bin/urldecode). To test it, just run it:
 

```
[root@localhost logs]# urldecode
hello
hello
this is a test: .%65x%65
this is a test: .exe
[root@localhost logs]#
```
- The script acts as a filter as it echoes information to the standard output. The command to test your logs should now be:
 

```
[root@merc root]# cat access_log | urldecode | grep
exe
```
- You can change exe into anything you want to look for in your log.

## Store your logs in another machine?

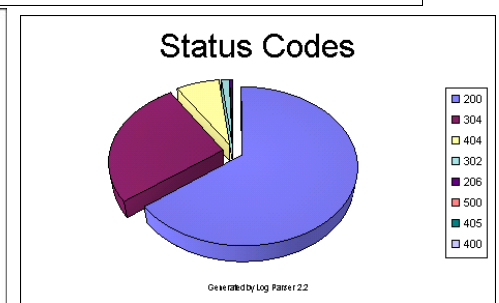
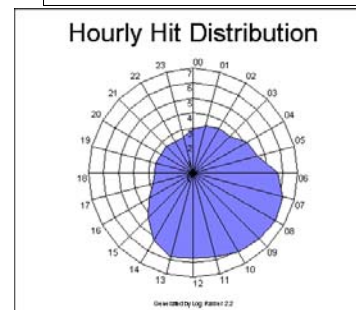
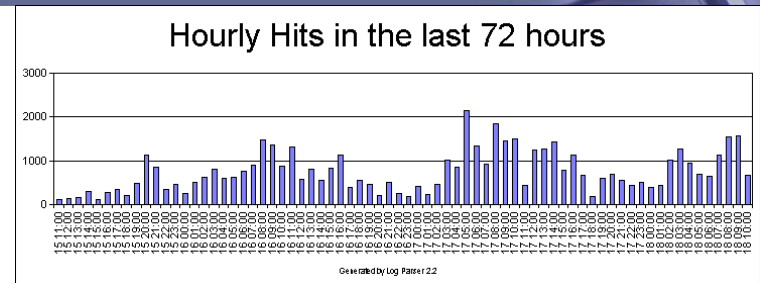
- In some cases, you'll want to store your logs on a separate, secure server on your network dedicated to logging. This means that your server won't be responsible for holding the logs, and if some crackers gain access to it, they won't be able to delete their tracks (unless they crack the log server as well).
- There are two ways of doing this.
  1. To instruct Apache to send all the log messages to the standard Unix log server, syslogd.
  2. To build a custom-made logger script that sends the log entries to a remote server. You can implement this in several ways, and it might prove to be better for security and simplicity.

## What is LogParser?!

- Just a free tool from Microsoft.
- A kind of noise reduction.
- It could be integrated with other tools to output graph and chart!
- It could get logs from text files, csv, XML files, or from database.
- It is used for both investigation and business analysis.

## Log Filtering and Noise Reduction

- This topic will be discussed by Sam NG deeply. However, I would like to recommend this tool for your log review and provide some demos to use.
- If you understand basic SQL query, it is advantageous.





## Log Review Role & Responsibility

<u>Roles</u>	<u>Servers</u>	<u>Devices</u>
<i>System Administrator</i>	Mail server, web server, file server, domain server, backup server, proxy server, terminal server, application server, log server, patch server, VoIP server, (fax server?), Antivirus/Antispyware, Web Server etc.	Routers, switches
<i>Database Administrator</i>	Database Server	Nil
<i>System Development Team</i>	Program version, system migration, system libraries access, application server	Nil
<i>Security Administrator (Technical and Business Streams)</i>	Firewall, web application firewalls, Anti-Virus/Anti-Spyware, Intrusion Detection Server, log server, specific application systems, user account, Log server, forensic server, authentication server, certificate server, encryption key server	Nil
<i>System Operator</i>	Batch/Job processing	Nil
<i>Physical security officer</i>	Nil	Door/Gate/Vault/ CCTV/Voice Recording

## Log Review Role & Responsibility

### From the matrix....

- The matrix may not cover full range of technologies and servers as well as devices.
- We found that it is hard to implement separation of duties for user registration and logs review.
- Technologies and various servers come to enterprise continuously.
- It is tough for a system administrator/security administrator to monitor every servers/devices with thorough understanding of criticality of logs entries.
- After finding suspicious logs, you need to proceed on investigation, you are required to interview with relevant personnel.

## Log Review Program

- Compliant with internal and external security policy, it is not your preference.

## Log Review Best Practice

*Presented by*

**Anthony Lai**

## Logs Review Best Practice

- Prerequisites:
  - Understand the business workflow picture
    - What departments are involved? You have got their contact information
    - What is the business hours of the business function?
    - Who is the system owner?
  - Getting sufficient technical background information
    - Error code and description
    - Criticality of errors (identified by and discussed with system owners)

## Log Review Best Practice (1)

- When allocating the responsibility for log review, a separation of roles should be considered between the persons undertaking the review and those whose activities are being monitored or....
  - With supervisor review if you are involved in operation task as well.
  - Then who will review the logs when your supervisor is on leave? Finding another managerial personnel to review?

## Log Review Best Practice (2)

- Particular attention should be given to the security of the logging facility because if tampered with it can provide a false sense of security. Controls should aim to protect against unauthorized changes and operational problems including:
  - The logging facility being de-activated
  - Alterations/Additions to the message types that are recorded
  - Log files being edited or deleted
  - Log file media becoming exhausted and either failing to record events or overwriting itself.

## Log Reviews Best Practice From BS7799

“System logs often contain a large volume of information, much of which is extraneous to security monitoring. That is the reason, we copy appropriate message types automatically to a second log, and/or the use of suitable system utilities or audit tools to perform file interrogation should be considered. Original set of logs will be kept in server for future reference and further investigation. “

## Log Review Checklist (1) – In a networked environment

- System startup: are there multiple run levels? If so, system should record which level is starting in some way that a human can make sense of it
- System shutdown: are there multiple modes of shutdown? Does the system have any capacity to send "oh my god i'm going down" messages in the case of an emergency crash or power loss? Are there distinctions between normal and abnormal shutdowns that can be differentiated in the logs?
- File system full: including thresholds (default or user defined) • boy wouldn't it be nice if the logs "automagically" included the three (or however many) biggest culprits in terms of file size or space consumed by a directory or folder in an error message?

## Log Review Checklist (2) – In a networked environment

- Hardware failures: power supplies, network interfaces, etc. I am relatively uneducated about hardware diagnostics, other than Cisco gear...
- Logins: failed and successful; console, remote (what protocol if remote); anonymous account, unprivileged user account, privileged user account, including switches to other users (unprivileged, privileged) from user accounts
- Account creation: failed and successful; adding new user ID, assigning rights and privileges to new user, adding password to new user

## Log Review Checklist (3) – In a networked environment

- Account modification: failed and successful; assigning or removing rights and privileges, resetting password; privileged user or unprivileged user
- Account removal: failed and successful
- Account disabled: too many failed logins, account expired, etc.
- Password/security information copied: failed and successful
- System configuration change: failed and successful; including access control, network addressing, audit policy; who made change, what changed, from system kernel on out to user-level applications

## Log Review Checklist (4) – In a networked environment

- Operating system patch applied: who applied patch, what system components changed, source of patch (?)
- Network connections: failed and successful connection attempts; anonymous service, user-specific service, access to administrative tools or control connection; DNS zone transfers, etc.
- Audit logs: failed and successful attempts to modify or clear audit logs
- Object access: failed and successful attempts to read files, start or stop processes, etc (understanding that most organizations will not need or want this level of detail)

## Log Review Challenges

- Log format are not standardized. Some system provides log but some don't. There is no explicit standard for vendor.
- Audit log facility may be deactivated/changed without noticed.
- Platform change/upgrade leads to review of existing logs monitoring.
- Logs integrity and retention.
- When incidence happens, then people will think of the logs standard.
- Ensure that it is a continuous process. There is no day-off for the review.

## Log Review Challenges

- No soft copy of the logs, no filtering could be engaged.
- Selecting critical logs is not a rocket science but your ignorance may neglect some kind of critical events/suspicious activities. People are willing to go for printing out all of the logs to review, of course, auditor knows that it is a joke to review a big heap of paper with a X-Ray eyes.

## Follow up

- Create standard and identify critical logs. Refer to latest security and audit log manual.
- Reference to log review best practice from vendors and global security organizations.
- Do not think of the response from Auditor in an extreme way, the most important point is that your selection of logs are sufficient for your control objective.
- Even you may get a heap of logs, trying to filtering out the noise with tools to review the logs from another view (like grouping by critical error code)
- Create your log review report template and score sheet.

## Resources

- Configure Apache Web Log
  - <http://www.webhostgear.com/69.html>
- WebGoat and WebScarab
  - <http://www.owasp.org>
- Hardening Apache by [Tony Mobily](#)
  - Search from online book stores!
- MS LogParser Official Site
  - Search from Google or simply click into:  
<http://www.microsoft.com/technet/scriptcenter/tools/logparser/default.mspx>
- Unofficial LogParser (Many tools and tips from there)
  - <http://www.logparser.com/>
  - [http://www.logparser.com/site\\_statistics.htm](http://www.logparser.com/site_statistics.htm)
- Forensic Parsing using LogParser
  - <http://www.securityfocus.com/infocus/1712>
- Microsoft Log Parser Toolkit
  - <http://www.syngress.com/catalog/?pid=3110>