Securing Apache Web Servers with Mod Security & CIS Benchmark

Ralph Durkee, CISSP, GSEC, GCIH, GSNA, GPEN
Principal Security Consultant
rd@rd1.net

About Ralph Durkee

- 25+ years of experience
  - Systems and Network Security
  - Software Development and Systems Administration
- Independent Consultant and Trainer since 1996
- SANS GIAC Certified since 2000
  - GSEC, GCIH, GSNA, GPEN
- Lead Developer, Author and Maintainer for the Center for Internet Security: RedHat Linux, DNS BIND, Apache
- Community Instructor for SANS
- CISSP Certified CISSP Instructor
- Rochester OWASP President & ISSA VP

Agenda

- Need A Secure Foundation
- Minimizing the Attack Surface
- Limiting HTTP Request Methods
- Access Control
- Mod_Security –
  Web Application Fire Wall
- Logging and Monitoring

Center for Internet Security Benchmarks

- Center for Internet Security
  - Non-profit Organization
  - Develops Technical Security Standards
  - Uses Consensus of Industry Experts
  - www.CISecurity.org
- Benchmarks for:
  - Most Unix and Windows Operating Systems
  - Several Servers such as Apache and BIND
  - Oracle and MS SQL Server Databases
  - Others applications are in the works

Need A Secure Foundation

Secure Foundation – OS Security

- Start with a Security Hardened OS
  - Unix or Linux recommended for Internet
  - Apply appropriate CIS OS Benchmark
  - Don’t mix other high risk, or critical services
- Regularly Apply OS and Apache updates

Secure Foundation – DNS Cache Poisoning Attacks

- DNS Level attacks against your clients /customers
- Secure your Authoritative and Caching DNS Servers with CIS BIND Benchmark
- DNS Pharming Attacks
  - Uses DNS Cache poisoning to harvest victims
  - Bogus IP Addresses provided to Vulnerable DNS Cache
  - Typically requires guessing DNS Query-ID and port
  - Clients resolve domain name are directed to a spoofed hostile website instead of trusted website

Dan Kaminsky's - DNS Attack

- Much more effective than traditional DNS cache poisoning. Uses:
  - Requests many random nonexistent host names
  - Send many negative responses with guessed QID
  - Response: Go to server NAME & IP has the answer.
  - Victim caches the IP address of "*DNS*" server
  - Game over the "*DNS*" server was the target
- Only Complete Prevention requires DNSSEC
- Securing the Caching DNS Server helps

Apache User Account

- Don’t run Apache as root
  - Use dedicated locked Account
  - Account with Invalid Shell such as /dev/null
  - Locked, with no valid password

```
Example Server Configuration
User apache
Group apache

# grep apache /etc/passwd /etc/shadow
apache:x:48:48:Apache:/var/www:/dev/null
apache:!!:14428:0:99999:7:::
```

Set Minimal Permissions

Ownership and Permissions

- Apache Configuration Files
  - Read-write by group Web Admin
  - Owned by Root
  - No access for Other
  - Apache reads these as root, before starting
- Document Root (and most sub-directories)
  - Read-write by group Web Development
  - Readable by Other
  - Owned by root

Set Minimal Permissions (2)

More Ownership and Permissions

- CGI-BIN Directories
  - Read-write by group Web Admin
  - Readable & Executable by Other
  - Owned by root
- Apache bin files (apachectl and httpd)
  - Read & Execute by Wed Admin
  - Read & Execute by root

Subscribe to Security Advisories

- Web Admin and System Admin should subscribed to appropriate advisories
- Apache
  `http://httpd.apache.org/lists.html`
- CERT
  `https://forms.us-cert.gov/maillists/`
- Sun
  `https://subscriptions.sun.com`
- Fedora Core
  `https://www.redhat.com/mailman/listinfo/fedora-announce-list`

Minimize the Attack Surface

Disable Unnecessary Modules

- Modules you probably DON'T need
  - **mod_dav -** Distributed Authoring and Versioning (WebDAV) functionality
  - **mod_dav_fs** – File System for mod_dav
  - **mod_status** – Provide Web Server status info.
  - **mod_proxy** – HTTP Proxy
  - **mod_autoindex -** Directory listings
  - **mod_cern_meta -** CERN HTTPD Meta file semantics (old not used)

Use only Necessary Modules

- **Modules you might need**
  - **mod_log_config** – Provides flexible for Logging of Requests
  - **mod_logio** – Provides I/O bytes per request
  - **mod_mime** – Determines MIME type / Handler by file extension
  - **mod_env** – Controls environment passed to CGI
  - **mod_expires** - Generation of `Expires` and `Cache-Control` HTTP headers

Check Config Include Directories

- Check any *config* include directories
  - Red Hat Linux uses /etc/httpd/conf.d
  - All *.conf files are auto included
  - Remove the rpm, not just the file
  - Or comment out the file content
- Example:

```
rpm –qf /etc/httpd/conf.d/manual.conf
    httpd-manual-2.2.xx-xx.x
rpm -e httpd-manual
```

Remove Any Default Files

- Default HTML Files
  - Manual
  - Welcome page
  - Directory Index icons
- Sample CGI files (e.g. printenv)
- Apache source code files
- Apache user files (.bashrc etc)

Other Resources for Modules

- Modules list available On-line

  http://httpd.apache.org/docs/2.0/mod/

  http://httpd.apache.org/docs/2.2/mod/

- Also Review Module recommendations in CIS Benchmark Appendix
- Some Modules have their own website, (such as modsecurity.org) check your favorite search engine.

Options Directive
Apache 2.2 docs

Description: Configures what features are available in a particular directory

Syntax: Options [+|-]option [[+|-]option] ...

Default: Options All

Context: server config, virtual host, directory, .htaccess

Override: Options

Module: core

Options Directive

Example 1 - Top Level Root

```
<Directory />
    . . .
    Options None
</Directory>
```

Example 2 – cgi-bin Directory

```
ScriptAlias /mailman/ /usr/lib/mailman/cgi-bin/
<Directory /usr/lib/mailman/cgi-bin/>
    . . .
    Options ExecCGI
</Directory>
```

Options Directive

- Options
  - **All** – Everything except Multiviews
  - **ExecCGI –** Execution of CGI scripts
  - **FollowSymLinks –** Will follow symbolic links
  - **SymLinksIfOwnerMatch –**only if owner matches
  - **Includes** - Enables Server Side include
  - **IncludesNOEXEC –** SSI without #exec
  - **AllowOverride –** Allow usage of .htaccess files.
  - **Multiviews** - Content negotiation (e.g. Language)

Access Controls

Auth and Authz Modules

- mod_authz_host (was mod_access) - Access based on IP address or hostname.
- mod_authz_user , mod_authz_groupfile Mod_auth - user authentication using text files

Access Control Directives (1)

- Protecting Root (httpd.conf)

```
<Directory />
    Options None
    AllowOverride None
    deny from all
</Directory>
```

- Allowing All Access

```
<Directory "/var/www/html/">
     Order allow,deny
     allow from all
</Directory>
```

Access Control Directives (2)

- Allowing Limited Access
- Usage of IP Address or partial IP Address

```
<Directory "/var/www/html/">
    Order allow,deny
    deny from all
    allow from 10.10.2.
</Directory>
```

- Domain and Host names also work

HTTP Basic Authentication

- Requires mod_auth enabled
- Send base64 encoded username and password sent with every request.
- Needs SSL to protect username/password
- No password guessing protection built-in
- Sample Configuration

```
<Directory /var/www/html/members>
AuthType Basic
AuthName “Memebers Access"
AuthUserFile /path/to/passwordfile
Require valid-user
</Directory>
```

HTTP Basic Authentication (2)

- Setup Apache Password file

```
htpasswd -c /path/to/passwordfile jsmith
  New password: password
  Re-type new password: password
  Adding password for user jsmith
```

- Don’t place Password file in the DocRoot
- Apache needs Read-only access
- Don’t allow other read access.

HTTP Digest Authentication

- Requires mod_auth and mod_digest enabled
- Uses Challenge – Response
- Response is encrypted with the password
- Does not protect data, still needs SSL
- No password guessing protection built-in
- Sample Configuration

```
<Directory /var/www/html/members>
AuthType Digest
AuthName “Members Access"
AuthUserFile /path/to/passwordfile
Require valid-user
</Directory>
```

New ChrootDir Directive

Description: Directory for apache to run chroot(8) after startup.

Syntax: ChrootDir /path/to/directory

Default: none

Context: server config

Module: event, prefork, worker

Compatibility: Available in Apache 2.2.10 and later

Example:

```
ChrootDir  /var/www/chroot
```

New ChrootDir Directive (2)

Apache Disclaimer:

Note that running the server under chroot is not simple, and requires additional setup, particularly if you are running scripts such as CGI or PHP. Please make sure you are properly familiar with the operation of chroot before attempting to use this feature.

New ChrootDir Directive (3)

- Makes chroot easier, but still work required.
- Some typical directories required:

```
CHR=/var/www/chroot/
mkdir –p $CHR/var/www
mv /var/www/* /var/www/chroot/var/www/
mkdir $CHR/var/run
mkdir $CHR/tmp
mkdir –p $CHR/ /var/lib/php/session
```

- Usually others? Your Mileage **Will** vary!

Apache and SELinux an Alternative to chroot

- A different (easier?) approach to chroot
- Implements Mandatory Access Controls
- Use SELinux in targeted mode
- In /etc/selinux/config, set
  `SELINUXTYPE=targeted`
- To test, start with
  `SELINUX=permissive`
- Switch to
  `SELINUX=enforcing`

Apache SELinux Polices

- **httpd_selinux**(8) man page defines contexts types:
  - **httpd_sys_content_t** - all content access
  - **httpd_sys_script_exec_t** – for scripts
- /etc/selinux/targeted/contexts/files/ file_contexts – labels directories with types
  - **/var/www/cgi-bin**(/.*)? system_u:object_r:**httpd_sys_script_exec_t:**s0
  - **/var/www(/.*)?** system_u:object_r:**httpd_sys_content_t**:s0

Checking SELinux Labels

- Use –Z option on ls to see SELinux labels

`ls -Z /var/www`

```
drwxr-xr-x  root      root
   system_u:object_r:httpd_sys_script_exec_t cgi-bin
drwxr-xr-x  root      root
   system_u:object_r:httpd_sys_content_t error
drwxr-xr-x  root      root
   system_u:object_r:httpd_sys_content_t html
drwxr-xr-x  root      root
   system_u:object_r:httpd_sys_content_t icons
drwxr-xr-x  webalizer root
   system_u:object_r:httpd_sys_content_t usage
```

Limiting HTTP Request Methods

HTTP Request Methods?

- RFC 2616 defines HTTP/1.1 Methods
  - **GET** - Most used – retrieves content
  - **HEAD** – Doesn't return body, used to check for existence and updates
  - **POST** – Typically used for FORM submissions
  - **PUT** – Push a resource up to the server
  - **DELETE** – Remove a resource
  - **TRACE** – For Debugging
  - **CONNECT** – for SSL Proxy connections

Limiting HTTP Request Methods

- Limit Methods to HEAD, GET and POST

```
<Directory "/var/www/html">
    Order allow,deny
    Allow from all
    <LimitExcept GET POST>
        deny from all
    </LimitExcept>
    Options None
    AllowOverride None
</Directory>
```

- TRACE is not limited by this!
- HEAD is included with GET

Deny HTTP Trace Mod_Rewrite Technique

- TRACE method part of RFC HTTP protocol
- Reflects the request back to the client
- Intended for Debug
- Used for XST (Cross-Site Tracing vulnerabilities)
- Use mod_rewrite to deny TRACE Method
- [F] Flag returns 403 Forbidden

```
RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

Deny HTTP Trace
New TraceEnable Directive

Description: Determines the behavior on TRACE requests

Syntax: TraceEnable [on|off|extended]

Default: TraceEnable on

Context: server config

Module: core

Compatibility: Available in Apache 1.3.34, 2.0.55 and later

Example:

```
TraceEnable off
```

Mod Security – The Web Application Firewall

modsecurity

Mod_Security Features

- **Open Source Web Application Firewall**
- **Features:**
  - Request filtering
  - Anti-evasion techniques - paths and parameters are normalized
  - Understands the HTTP protocol
  - Performs very specific and fine grain filtering.
  - POST payload analysis

Mod_Security Features (2)

- **More Features:**
  - Audit logging - Full details can be logged for later analysis
  - HTTPS – Analysis performed after decryption
  - Inspect and Filter Any Headers
  - Buffer Overflow Protection
  - Attack Detection and Prevention

Mod_security Configuration

- Easily Installed via package, or build from source.
- Configuration mod_security.conf
- Rename file if using include conf.d/

```
LoadModule security_module modules/mod_security.so
<IfModule mod_security.c>

# Turn the Filtering and Audit engine, On
SecFilterEngine On
SecAuditEngine RelevantOnly
```

Mod_security Configuration (2)

- More Basic Feature Configuration

```
# Make sure that URL encoding is valid
SecFilterCheckURLEncoding On
# Unicode encoding check
SecFilterCheckUnicodeEncoding On
# Only allow bytes from this range
SecFilterForceByteRange 1 255
# Cookie format checks.
SecFilterCheckCookieFormat On
# The name of the audit log file
SecAuditLog logs/audit_log
# Should mod_security inspect POST payloads
SecFilterScanPOST On
# Default action set
SecFilterDefaultAction "deny,log,status:406"
```

Mod_security Filters (1)

Basic Recommended Filters

```
# Require HTTP_USER_AGENT and HTTP_HOST headers
 SecFilterSelective "HTTP_USER_AGENT|HTTP_HOST" "^$"

 # Only accept request encodings we how handle
 # we exclude GET requests because some (automated)
 # clients supply "text/html" as Content-Type
 SecFilterSelective REQUEST_METHOD "!^GET$" chain
 SecFilterSelective HTTP_Content-Type    "!(^$|
   ^application/x-www-form-urlencoded$|^multipart/form-
   data)"
```

Mod_security Filters (2)

More Basic Recommended Filters

```
# Require Content-Length to be provided with
# every POST request
SecFilterSelective REQUEST_METHOD "^POST$" chain
SecFilterSelective HTTP_Content-Length "^$"


# Don't accept transfer encodings we don't handle
SecFilterSelective HTTP_Transfer-Encoding "!^$"
```

Logging and Monitoring

Logging Directives

- LogLevel
  - Controls Verbosity
  - Values are emerg, alert, crit, error, warn, notice, info and debug
  - Notice is recommended
- ErrorLog – File name for logging errors
- LogFormat – Defined format of log entries
- CustomLog logs/acces_log combined

Logging Directives (2)

- Sample Logging Configuration

```
LogLevel notice
ErrorLog logs/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b \"%
  {Accept}i\" \"%{Referer}i\" \"%{User-Agent}i\""
  combined
CustomLog logs/access_log combined
```

- Combined format is fairly standard and handled well by log analysis software
- Use Swatch or LogWatch for log monitoring.

Log Monitoring

- Sample LogWatch output with Web Attacks

```
Requests with error response codes
    404 Not Found
        //README: 2 Time(s)
        //chat/messagesL.php3: 1 Time(s)
        //graph_image.php: 1 Time(s)
        /PhpMyChat//chat/messagesL.php3: 1 Time(s)
        /horde-3.0.5//README: 2 Time(s)

    406 Not Acceptable
        /: 2 Time(s)
        /robots.txt: 1 Time(s)
```

Log Monitoring (2)

- More Samples of Web Scans / Attacks
- Looking for open proxy & phone apps?

```
400 Bad Request
  http://www.wantsfly.com/prx.php?hash=457F6 ...
404 Not Found
  /apple-touch-icon.png: 1 Time(s)
  /iphone/: 2 Time(s)
  /mobi/: 2 Time(s)
  /mobile/: 2 Time(s)
  /pda/: 2 Time(s)
  /sql/: 1 Time(s)
```

Abuse Reports

- **Why Report Attacks on your Servers?**
  - Makes it a more difficult for the attacker (Yeah, mostly for the script kiddies)
  - Educates organizations on the state of their system and their need for response
  - Helps make the Internet a better place
- Choose your “favorites” to report
- Use whois on IP address of the source IP to abuse email contact
- Reporting to questionable organizations may not be helpful, or helpful in the wrong way.

Abuse Reports – How to (2)

- Keep it Simple Just the facts.

To: abuse@example.com

Subject: web vulnerability attack from IP xx.xx.xx.xx

Logs are included below of a web vulnerability attack from the above address. This system may have been compromised or infected. Please take action to prevent further abuse. An e-mail reply is appreciated. Thank you for taking action on this.

-- Ralph Durkee, CISSP, GSEC, GCIH, GSNA, GPEN
Information Security Consultant
USA 585-624-9551

Logs are NTP time synced in USA EDT TZ

Abuse Reports (2)

- **Send Sample of Access Web Logs**

xx.xx.xx.xx - - [03/Sep/2009:06:26:31 -0400] "GET /scripts/setup.php HTTP/1.1" 404 215 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

xx.xx.xx.xx - - [03/Sep/2009:06:26:31 -0400] "GET /scripts/setup.php HTTP/1.1" 404 215 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

xx.xx.xx.xx - - [03/Sep/2009:06:26:31 -0400] "GET */phpMyAdmin/* HTTP/1.1" 404 209 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"

xx.xx.xx.xx - - [03/Sep/2009:06:26:31 -0400] "GET */sql/* HTTP/1.1" 404 202 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)”

Abuse Reports (3)

- **Some Recent Interesting User Agent in Logs**

xx.xx.xx.xx - - [03/Sep/2009:20:04:50 -0400] "GET / HTTP/1.0" 200 67 "-" "**Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A543a Safari/419.3**“

xx.xx.xx.xx - - [03/Sep/2009:20:05:01 -0400] "GET /apple-touch-icon.png HTTP/1.0" 404 218 "-" "**Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A543a Safari/419.3**"

Abuse Responses

From: Amazon EC2 Abuse ec2-abuse-team@amazon.com

Thank you for submitting your abuse report.
We have received your report of Intrusion Attempts originating from our network.

We have completed an initial investigation of the issue and learned that the activity you noticed did indeed originate from an Amazon EC2 instance. These intrusion attempts that you report were not, however, initiated by Amazon.

One of the biggest advantages of Amazon EC2 is that developers are given complete control of their instances. . . .

That said, we do take reports of unauthorized network activity from our environment very seriously. It is specifically forbidden in our terms of use. This instance has since been terminated.

OSSEC.net

- OSSEC – Open Source HIDS, central logging and monitoring solution – aka SIM/SEM/SIEM
- Supports most platforms Linux/Unix/Windows/Mac
- Real-time alerting
- Active response - blocking of attacks
- Agent and Agentless monitoring
- File Integrity Monitoring
- Rootkit detection

Questions?

Durkee Consulting, Inc.
www.rd1.net rd@rd1.net