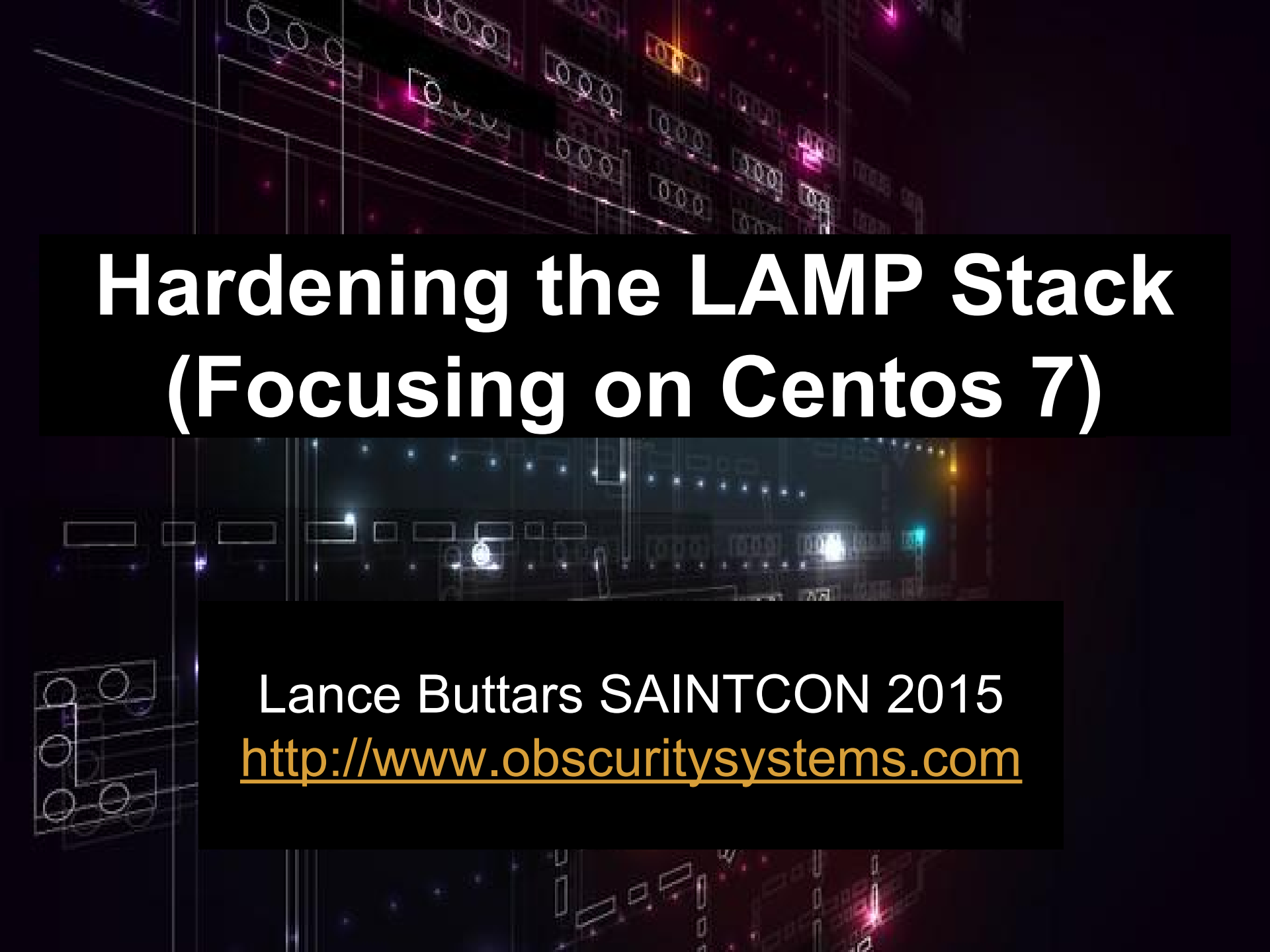


Hardening the LAMP Stack (Focusing on Centos 7)

Lance Buttars SAINTCON 2015
http://www.obscuritysystems.com

Who am I

Security Research / Software Developer

Twitter @Nemus801

I am a member of the local

Defcon Group **www.dc801.org**
Freenode #dc801. Nemus

I help organize and run 801 Labs which is a hackerspace located in downtown Salt Lake City.

801 Labs Hacker Space www.801labs.org

Defcon Talks www.introtobackdoors.com

Websecurity Warriors Podcast

http://websecuritywarriors.com/

What is Hardening?

My definition of **Hardening** is the process of securing a system by reducing its surface of vulnerability and reducing information leakage.

Security Definitions

Confidentiality

- Information is not disclosed to unauthorized parties.

Integrity

- Information remains unchanged in transit or in storage until it is changed by an authorized party.

Availability

- Authorized parties are given timely and uninterrupted access to resources and information.

Security Principles

Secure the weakest link -Identify the weakest link.

Defend in depth - Layers of defense use multiple tools and techniques.

Fail securely - When something fails have it fail securely and closed.

Grant least privilege - Need to know or need of use.

Separate privileges - Compartmentalize privileges.

Economize mechanism - Complexity is the enemy of security engineering and the friend of the attacker.

Do not share mechanisms - if two groups of user / systems / applications don't need to be together separate them.

http://searchsecurity.techtarget.com/opinion/Thirteen-principles-to-ensure-enterprise-system-security

Security Principles Part 2

Be reluctant to trust - Assume your environment is insecure.

Assume your secrets are not safe - Think as though your attacker knows what you know.

Mediate completely - Test every security mechanism you put in place.

Make security usable - if your security is difficult to follow people will circumvent or ignore it.

Promote privacy - Think about the privacy of your users.

Use your resources - If you're not sure whether your system design is secure, ask for help.

Threat Modeling

Threat modeling is a method for conceptualizing about threats. It includes what attack surfaces you have, who has interest in your assets and what your vulnerabilities are. With this information you can make informed decisions about which threats you want to mitigate.

Assessment Scope - By identifying all components and assets.

System Modeling - Model Components relation.

Identify Threats - Categorize possible attack vectors and components.

Identify Vulnerabilities - Identity vulnerabilities in your model.

Threat ranking - Rank threat priority

Mitigation - What are the costs for mitigating each threat.

https://www.owasp.org/index.php/Category:Threat_Modeling

Why attacks take place

Defacement - Website defacement or pranking.

Spamming - Directing users to spam links directing visitors to another site.

Spreading Malware - Hosting malicious code that installs itself onto a visitor's computer.

Credential Theft - Theft of other user session IDs (cookies).

Steal Userinfo - Stealing visitor information and browsing habits.

Information Theft - Information stored in the database

Access Restricted Content - Access content hidden on the site.

Prevent Access - Deny users access to resources.

Attacks on Web Servers

Denial of service

- Flooding system with traffic.
- Finding code or resources that use a lot of cpu.

Attacks on Clients

- Phishing

Session Attacks

- Attacks on cookies
- Session hijacking
  - Involuntary token leak
  - Voluntary token leak
  - Session fixation

Attacks continued

Injection Flaws

- Cross-Site Scripting
- Code Execution
- Command Execution
- Buffer Overflows

Information Disclosure

- Directory Listing
- Verbose Error Messages
- Debug Messages

Attacks yet again

Application Logic Flaws

- Attacking flaws in the application logic flow.
  - Doing sequential tasks out of order
  - Changing values of hidden values.

Path Traversal

- Path traversal occurs when the apache or php can be used to back reference files path outside of the web application path to gain access to the parent folder of a subfolder.


Linux

Centos 7 Hardening

- https://highon.coffee/blog/security-harden-centos-7/
- http://benchmarks.cisecurity.org/downloads/browse/index.cfm?category=benchmarks.os.linux.redhat

Iptables

sudo yum install -y iptables-services

sudo systemctl enable iptables.service

sudo /usr/libexec/iptables/iptables.init save

yum install net-tools

http://www.cipherdyne.org/LinuxFirewalls/ch01/

Server Setup

yum install httpd

yum install mariadb mariadb-server

yum install php php-mysql

yum install denyhosts

yum install fail2ban

yum install epel-release

yum install denyhosts

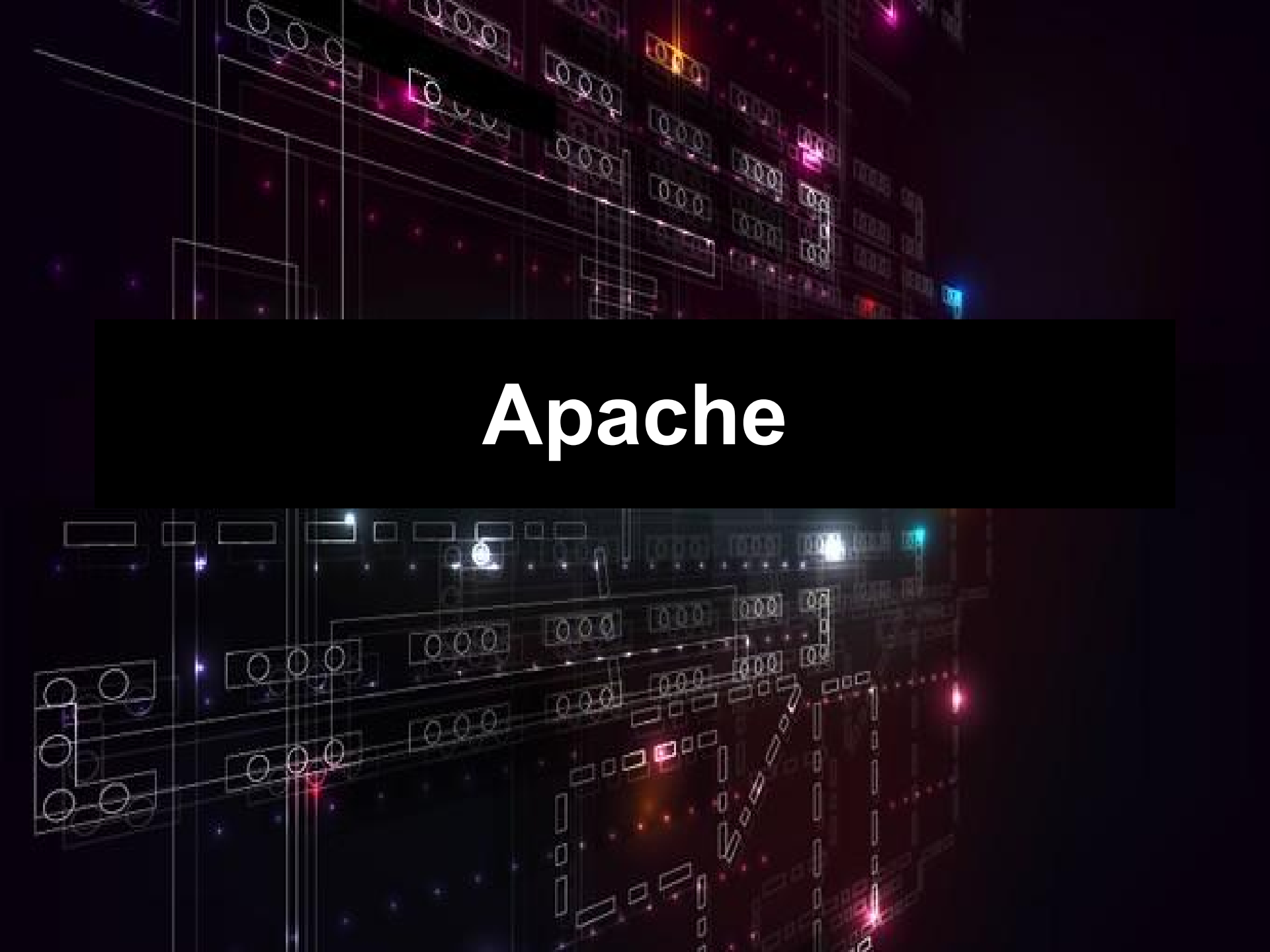
yum install fail2ban

yum install screen wget

yum install vim

yum install stunnel aide

yum install sudo


Apache

Configuration Hardening

Apache should not run as root

```
groupadd apache
useradd apache -g apache -d /dev/null -s /sbin/nologin
#/etc/httpd/conf/httpd.conf
User apache
Group apache

chown -R root:root  /etc/httpd
chmod -R 640 /etc/httpd
```

There is no need for any other users to be able to read the Apache configuration or the logs:

```
chmod -R 640 /etc/httpd/conf
chmod -R 640 /var/log/httpd
```

Secure Defaults

```
<Directory />
    Options None  # Options Directive
    Order Deny,Allow
    Deny from all
    AllowOverride None # AllowOverride Directive
</Directory>

<Directory /var/www/htdocs>
    Order Allow,Deny
    Allow from all
</Directory>
```

Options Directive

All - Allows all options listed below except MultiViews. This is the default setting.

None - None of the options will be enabled.

ExecCGI - Allows execution of CGI scripts.

FollowSymLinks - Allows symbolic links to be followed.

Includes - Allows server-side includes.

IncludesNOEXEC - Server-side includes are permitted,but the #exec cmd and #exec cgi are disabled

Options Directive 2

Indexes - Allows the server to generate the list of files in a directory when a default index file is absent.

MultiViews - Allows content negotiation.

SymLinksIfOwnerMatch - Allows symbolic links to be followed if the owner of the link is the same as the owner of the file it points to.

http://httpd.apache.org/docs/2.4/mod/core.html#options

Symbolic Links

If you need symbolic links use the Alias directive which Apache use to incorporate an external folder into the web server path. It serves the same purpose but is more secure. For example, it is used in the default configuration to allow access to the Apache manual:

The following configuration directive will disable symbolic link usage in Apache:

Options -FollowSymLinks

If you want to keep symbolic links then make sure to enable ownership verification on by setting the SymLinksIfOwnerMatch option. With this enabled Apache will follow symbolic links only if the target and the destination belong to the same user.

Options -FollowSymLinks +SymLinksIfOwnerMatch

mod_alias

```
Alias "/image" "/ftp/pub/image"
<Directory "/ftp/pub/image">
    Require all granted
</Directory>
```

http://httpd.apache.org/docs/current/mod/mod_alias.html#Alias

.htaccess

The only time you should only use .htaccess files is when you don't have access to the main server configuration file. There exists a common misconception that developers should use .htaccess files for mod_rewrite and other configurations options.

This is simply not the case. You can put any .htaccess file in the main server configuration. Additionally mod_rewrite directives work better, in many respects, in the main server configuration.

When AllowOverride is set to allow the use of .htaccess files, apache will look in every directory for .htaccess files. Thus, permitting .htaccess files causes a performance hit even if you're not using them. Additionally the .htaccess file is loaded every time a document is requested.

http://httpd.apache.org/docs/2.2/howto/htaccess.html

AllowOverride directive

This directive declares which directives in distributed .htaccess files can override directives from httpd.conf

AuthConfig - Allows use in .htaccess files of the authorization directives.

FileInfo - Allows use of the directives controlling document types

Indexes - Allows use of the directives controlling directory indexing

Limit - Allows use of the directives controlling host access

Options - Allows use of the directives controlling specific directory functions (the Options and

XbitHack directives)

All - Allows all options listed

None - Ignores .htaccess configuration files

Information Leakage

Edit /etc/httpd/conf/httpd.conf

#add these values

ServerSignature Off

ServerTokens ProductOnly

ServerName www.example.com:80

#commit server tokens out

#ServerTokens OS

TraceEnable Off

ServerAdmin you@somedomain.com

Information Leakage from HTTP Headers

curl -v http://127.0.0.1

* About to connect() to 127.0.0.1 port 80 (#0)

*   Trying 127.0.0.1... connected

* Connected to 127.0.0.1 (127.0.0.1) port 80 (#0)

> GET / HTTP/1.1

> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.16.2.3 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2

> Host: 127.0.0.1

> Accept: */*

< HTTP/1.1 200 OK

< Date: Wed, 06 May 2015 03:32:47 GMT

< Server: Apache/2.2.15 (CentOS)

< Last-Modified: Wed, 06 May 2015 03:32:43 GMT

< ETag: "c0653-e-5156172b104c1"

Rejected File Types

To prevent unintentional file disclosure, you should turn off automatic indexing and instruct Apache to reject all requests for files matching unwanted file types.

```
<FilesMatch "(^\.ht|~$|\.old$|\.git\.SVN\.svn $)">
Order Allow,Deny
Deny from all
</FilesMatch>
```

Removing Default Content

Remove all references to default content

/etc/httpd/conf.d/welcome.conf

```
#
# This configuration file enables the default "Welcome"
# page if there is no default index page present for
# the root URL.  To disable the Welcome page, comment
# out all the lines below.
#
#<LocationMatch "^/+$">
#    Options -Indexes
#    ErrorDocument 403 /error/noindex.html
#</LocationMatch>
```

Changing the name using mod_security

#/etc/httpd/conf/httpd.conf

Reveal full identity (standard Apache directive)

ServerTokens Full

#/etc/httpd/conf/mod_securutity/config.conf

Replace the servername (mod_security directive)

SecServerSignature "Microsoft-IIS/5.0"

More on mod_security later*

Remove unused Apache Modules

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

#LoadModule authn_anon_module modules/mod_authn_anon.so

#LoadModule cgi_module modules/mod_cgi.so

#LoadModule suexec_module modules/mod_suexec.so

#LoadModule dav_module modules/mod_dav.so

#LoadModule dav_fs_module modules/mod_dav_fs.so

#LoadModule ldap_module modules/mod_ldap.so

Chrooting

```
ldd /bin/bash
        linux-vdso.so.1 =>  (0x00007fff7e3ff000)
        libtinfo.so.5 => /lib64/libtinfo.so.5
(0x00007fad11e82000)
        libdl.so.2 => /lib64/libdl.so.2
(0x00007fad11c7e000)
        libc.so.6 => /lib64/libc.so.6
(0x00007fad118e9000)
```

Chrooting 2

```
mkdir /chroot/
cp /lib/linux-vdso.so.1 /chroot/lib/
mkdir /chroot/lib64
cp /lib64/libtinfo.so.5 /chroot/lib64
cp /lib64/libdl.so.2 /chroot/lib64/libtinfo.so.5
cp /lib64/libdl.so.2 /chroot/lib64/
cp /lib64/libc.so.6  /chroot/lib64/
cp /lib64/ld-linux-x86-64.so.2 /chroot/lib64/
chroot /chroot /bin/bash
```

Strace

Using chroot and ldd you will be able to put programs inside a chrooted jail. The only issue is when they crash you will not know why.

By using strace you can diagnose why they fail. For that reason, you will often need strace inside the jail itself.

strace uname

```
execve("/bin/uname", ["uname"], [/* 24 vars */]) = 0
brk(0)                                  = 0x215d000
mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fd6425d5000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat(3, {st_mode=S_IFREG|0660, st_size=17244, ...}) = 0
```

Chrooting Apache

ldd /usr/sbin/httpd

```
linux-vdso.so.1 =>  (0x00007ffff17ff000)
libm.so.6 => /lib64/libm.so.6 (0x00007fe53005f000)
libpcre.so.0 => /lib64/libpcre.so.0 (0x00007fe52fe33000)
libselinux.so.1 => /lib64/libselinux.so.1 (0x00007fe52fc13000)
libaprutil-1.so.0 => /usr/lib64/libaprutil-1.so.0 (0x00007fe52f9ef000)
libcrypt.so.1 => /lib64/libcrypt.so.1 (0x00007fe52f7b8000)
libexpat.so.1 => /lib64/libexpat.so.1 (0x00007fe52f58f000)
libdb-4.7.so => /lib64/libdb-4.7.so (0x00007fe52f21b000)
libapr-1.so.0 => /usr/lib64/libapr-1.so.0 (0x00007fe52efef000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007fe52edd1000)
libc.so.6 => /lib64/libc.so.6 (0x00007fe52ea3d000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007fe52e839000)
/lib64/ld-linux-x86-64.so.2 (0x00007fe530543000)
libuuid.so.1 => /lib64/libuuid.so.1 (0x00007fe52e634000)
libfreebl3.so => /lib64/libfreebl3.so (0x00007fe52e431000)
```

Is Chrooting Secure?

No not really it was a old stop gap method used before the days of selinux.

Chrooting was never intended to be a security feature.

http://www.linuxsecurity.com/content/view/117632/49/

SeLinux config for Apache

https://beginlinux.com/server_training/web-server/976-apache-and-selinux

Sooner or later you may run into situations where SELinux denies access to something and you need to troubleshoot the issue.

http://www.serverlab.ca/tutorials/linux/web-servers-linux/configuring-selinux-policies-for-apache-web-servers/

Selinux For Apache

setsebool -P httpd_can_network_connect_db 1

restorecon -rv /var/www/html

yum install policycoreutils-python

setsebool -P httpd_can_network_connect 1

yum install setroubleshoot setools

getsebool -a | grep httpd

https://www.centos.org/docs/5/html/Deployment_Guide-en-US/sec-sel-building-policy-module.html

audit2allow

yum update selinux-policy* libse* policycoreutils

getsebool -a

http://blog.endpoint.com/2012/05/selinux-local-policy-modules.html

SSL Setup

```
yum install mod_ssl openssl
http://wiki.centos.org/HowTos/Https
# Generate private key
openssl genrsa -out ca.key 4096
# Generate CSR
openssl req -new -key ca.key -out ca.csr
# Generate Self Signed Key
openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
# Copy the files to the correct locations
cp ca.crt /etc/pki/tls/certs
cp ca.key /etc/pki/tls/private/ca.key
cp ca.csr /etc/pki/tls/private/ca.csr
chmod 400 /etc/pki/tls/private/ca.key
```

SSL Configuration

SSLHonorCipherOrder on

SSLProtocol all -SSLv2 -SSLv3

SSLHonorCipherOrder on

SSLCipherSuite "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384 EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS

openssl s_client -host www.yourdomain.com -port 443

https://www.ssllabs.com/ssltest/

Fail Safe

Preventing configuration mistakes

If you are running a website that needs to be SSL only, then avoid a chance of having the sever host the same content available through a non-SSL port by creating virtual host that points to an empty folder. Use a RedirectPermanent directive to redirect users to the correct (secure) location:

```
<VirtualHost *:80>
    ServerName www.yourdomain.com
    DirectoryRoot /var/www/test
    RedirectPermanent / https://www.yourdomain.com/
</VirtualHost>
```

If the site contains SSL and non-SSL content, separating the content into two virtual hosts and separate directories decreases the chance of providing sensitive information without SSL. If the content must be put under the same directory tree, consider creating a special folder where the secure content will go. Then tell Apache to allow access to that folder only when SSL is used:

```
<Directory /var/www/htdocs/secure>
    # SSL must be used to access this location
    SSLRequireSSL
    SSLOptions +StrictRequirSSLOptions +StrictRequire
</Directory>
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule ^/secure(.*) https://%{SERVER_NAME}/secure$1 [R,L]
```

Useful OpenSSL commands

Check a Certificate Signing Request (CSR)

openssl req -text -noout -verify -in CSR.csr

Check a private key

openssl rsa -in privateKey.key -check

Check a certificate

openssl x509 -in certificate.crt -text -noout

Check a PKCS#12 file (.pfx or .p12)

openssl pkcs12 -info -in keyStore.p12

https://www.sslshopper.com/article-most-common-openssl-commands.html?jn29cd232d=2

Denial of Service Attacks

SYN Flood Attacks

A SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic - wikipedia

hping3 -S 192.168.56.2 -p 80 --flood

hping3 -S 192.168.56.2 -a 192.168.1.1 -p 80 --flood

temporary fix

iptables -A INPUT -p tcp -m state --state NEW -m recent --update --seconds 60 --hitcount 20 -j DROP

iptables -A INPUT -p tcp -m state --state NEW -m recent --set -j ACCEPT

#filter for odd typed syn packets

iptables -t mangle -I PREROUTING -p tcp -m tcp --dport 80 -m state --state NEW -m tcpmss ! --mss 536:65535 -j DROP

http://hakin9.org/syn-flood-attacks-how-to-protect-article/

Denial of Services Part 2

Distributed Denial of Service Attacks

As clarification, distributed denial-of-service attacks are sent by two or more people, or bots, and denial-of-service attacks are sent by one person or system. As of 2014, the frequency of recognized DDoS attacks had reached an average rate of 28 per hour - wikipedia

Reflection DoS Attacks

In computer security, a reflection attack is a method of attacking a challenge-response authentication system that uses the same protocol in both directions. That is, the same challenge-response protocol is used by each side to authenticate the other side. The essential idea of the attack is to trick the target into providing the answer to its own challenge - wikipeda

Denial of Services Part 2

Brute-Force Attacks

A brute force attack can manifest itself in many different ways, but primarily consists in an attacker configuring predetermined values, making requests to a server using those values, and then analyzing the response - Owasp

Poorly Designed Web Applications

Taking advantage of slow system resources such as pages that load slowly or run long running database queries that eat system resources.

Mod_evasive

mod_evasive is an module for Apache that provides evasive action in the event of a DoS attack or brute force attack.

```
yum install mod_evasive
setsebool -P httpd_can_sendmail 1
```

Mod Evasive Config

```
DOSEmailNotify you@yourdomain.com
LoadModule evasive20_module modules/mod_evasive20.so
DOSSystemCommand "sudo /etc/httpd/scripts/ban_ip.sh %s"
<IfModule mod_evasive20.c>
    DOSHashTableSize    3097
    DOSPageCount        2
    DOSSiteCount        50
    DOSPageInterval     1
    DOSSiteInterval     1
    DOSBlockingPeriod   10
</IfModule>
```

DOSHashTableSize:
The size of the hash table that is used to keep track of activity on a per-IP address basis. Increasing this number will provide a faster look up of the sites that the client has visited in the past, but may impact overall performance if it is set too high.

DOSPageCount:
The number of identical requests to a specific URI (for example, a file that is being served by Apache) a visitor can make over the DOSPageInterval interval.

DOSSiteCount:

similar to DOSPageCount, but refers to how many overall requests can be made to the site over the DOSSiteInterval interval.

DOSBlockingPeriod: If a visitor exceeds the limits set by DOSSPageCount or DOSSiteCount, he/she will be blacklisted for the DOSBlockingPeriod amount of time. During this interval, any requests coming from him/her will return a 403 Forbidden error.

Mod_evasive Script

```
#!/bin/sh
# Offending IP as detected by mod_evasive
IP=$1
# Path to iptables binary executed by user apache through sudo
IPTABLES="/sbin/iptables"
# mod_evasive lock directory
MOD_EVASIVE_LOGDIR=/tmp
# Add the following firewall rule (block IP)
$IPTABLES -I INPUT -s $IP -j DROP
# Unblock offending IP after 2 hours through the 'at' command;
echo "$IPTABLES -D INPUT -s $IP -j DROP" | at now + 2 hours
# Remove lock file for future checks
rm -f "$MOD_EVASIVE_LOGDIR"/dos-"$IP"
```

http://xmodulo.com/harden-apache-web-server-mod_security-mod_evasive-centos.html

Sudo File edit to allow the evasive command from apache

```
apache ALL=NOPASSWD: /usr/local/bin/scripts/ban_ip.sh Defaults:apache !requiretty
```

Fine Tuning Settings to prevent DOS

```
# wait up to 60 seconds for slow clients default is 300
TimeOut 60
# allow connections to be reused between requests
KeepAlive On
# allow a total of 100 requests per connection
MaxKeepAliveRequests 100
#Wait up to 15 seconds for the next request on an open connection
KeepAliveTimeout 15
```

Configuration Limits

The following directives enforce limits on various properties of HTTP request:

```
# impose no limits on the request body size default is  0 which is 2GB
LimitRequestBody 64 #if you're not supporting file upload
# allow up to 100 header fields in a request
LimitRequestFields 100
# each header may be up to 8190 bytes long
LimitRequestFieldsize 8190
# the first line of the request can be
# up to 8190 bytes long
LimitRequestLine 8190
# limit the XML request body to 1 million bytes(Apache 2.x only)
LimitXMLRequestBody 1000000
```

Multiprocessing Modules
MPM

All versions of Apache 2.0 and greater run MPMs. There are a different versions of MPMs for the various operating systems.

The MPMs work by modifying how Apache listens to the network connections, accepts and handles http requests.

For more information on the various MPMs go to

- http://codebucket.co.in/apache-prefork-or-worker/
- http://codebucket.co.in/apache-prefork-or-worker/

MPM - Settings

```
# the maximum number of processes
ServerLimit 16
# how many processes to start with
StartServers 2
# how many threads per process to create
ThreadsPerChild 25
# minimum spare threads across all processes
MinSpareThreads 25
# maximum spare threads across all processes
MaxSpareThreads 75
# maximum clients at any given time
MaxClients 150
```

Common Log Format

One of the advantages of Apache is its flexibility when it comes to log formatting. through the use of the LogFormat directive. Which referred to as the Common Log Format (CLF):

LogFormat "%h %l %u %t \"%r\" %>s %b" common

http://httpd.apache.org/docs/2.4/mod/mod_log_config.html

Apache Logging

TransferLog

TransferLog is the basic request logging directive, /var/log/httpd/access.log

CustomLog

#dynamic requests

CustomLog /var/www/logs/application_log combined env=!static_request

Error Logging

The Apache error log contains error messages and information about events unrelated to httpd requests. The error log contains what the access.log doesn’t.

- Startup and shutdown messages
- Various informational messages
- Errors that occurred during request serving http 400 500
- Critical process events
- Standard error output (stderr)

Web Application Firewalls (WAF's)

An application firewall is a form of firewall that controls input, output, and/or access from, to, or by an application or service. It operates by monitoring and potentially blocking the input, output, or system service calls that do not meet the configured policy of the firewall. The application firewall is typically built to control all network traffic on any OSI layer up to the application layer. It is able to control applications or services specifically, unlike a stateful network firewall, which is - without additional software - unable to control network traffic regarding a specific application. There are two primary categories of application firewalls, network-based application firewalls and host-based application firewalls - Wikipeda

Mod_security Setup

yum install mod_security

LoadModule security2_module modules/mod_security2.so

Mod_security Setup

```
mkdir /etc/httpd/mod_securityrules

<IfModule security2_module>
    Include crs/owasp-modsecurity-crs/modsecurity_crs_10_setup.conf
    Include crs/owasp-modsecurity-crs/base_rules/*.conf
</IfModule>
```

Spiderlabs Owasp Rules

mkdir /etc/httpd/owasp-modsecurity-crs

cd /etc/httpd/owasp-modsecurity-crs

wget https://github.com/SpiderLabs/owasp-modsecurity-crs/tarball/master

tar xzf master

mv SpiderLabs-owasp-modsecurity-crs-ebe8790 owasp-modsecurity-crs

cp modsecurity_crs_10_setup.conf.example modsecurity_crs_10_setup.conf

Mod Security Config

```
<IfModule mod_security2.c>
    SecRuleEngine On
    SecRequestBodyAccess On
    SecResponseBodyAccess On
    SecResponseBodyMimeType text/plain
text/html text/xml application/octet-stream
    SecDataDir /tmp
</IfModule>
```

Modsecurity Config Explained

SecRuleEngine On: Use the OWASP CRS to detect and block malicious attacks.

SecRequestBodyAccess On: Enable inspection of data transported request bodies (e.g., POST parameters).

SecResponseBodyAccess On: Buffer response bodies (only if the response MIME type matches the list configured with SecResponseBodyMimeType).

Modsecurity Config Part 2

SecResponseBodyMimeType text/plain text/html text/xml application/octet-stream: Configures which MIME types are to be considered for response body buffering. If you are unfamiliar with MIME types or unsure about their names or usage, you can check the Internet Assigned Numbers Authority (IANA) web site.

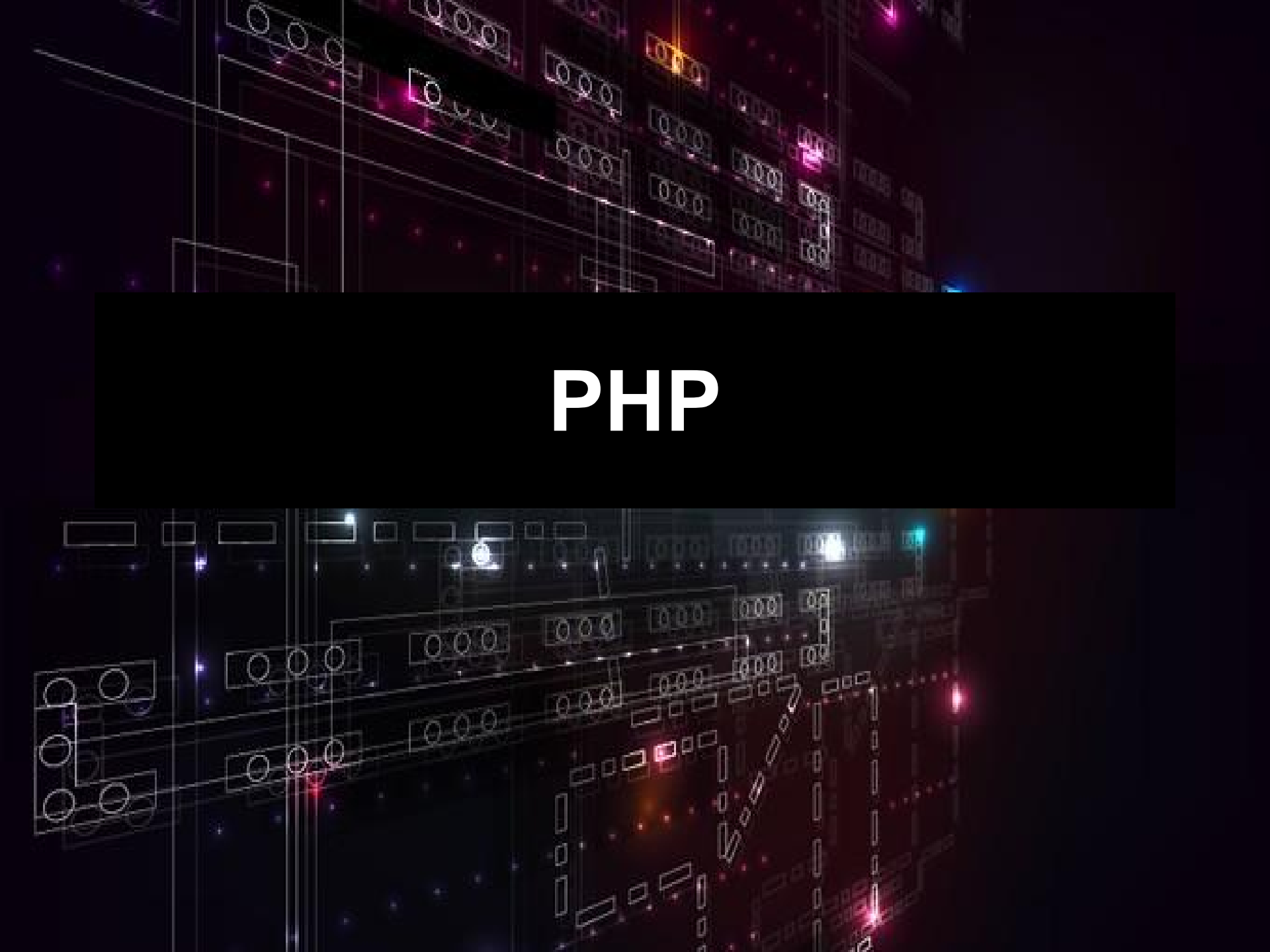
SecDataDir /tmp: Path where persistent data (e.g., IP address data, session data, and so on) is to be stored. Here persistent means anything that is not stored in memory, but on hard disk.

SecRuleEngine DetectionOnly

If you have a pre existing site run mod secure in detection mode first and read the log files to look for false positives.

SecRuleEngine On|Off|DetectionOnly

https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual


PHP

Securing php.ini

```
expose_php=Off
display_errors=Off
log_errors=On
error_log=/var/log/httpd/php_scripts_error.log
file_uploads=Off
file_uploads=On
upload_max_filesize=1M
allow_url_fopen=Off
allow_url_include=Off
sql.safe_mode=On
allow_url_fopen=Off
allow_url_include=Off
post_max_size=1K
```

php.ini config

safe_mode = On

safe_mode_gid = On

allow_url_fopen = Off

open_basedir = /path/to/web/root

register_globals = Off

enable_dl = Off

register_globals = Off

disable_functions = openlog

upload_tmp_dir = /var/php_tmp

upload_max_filezize = 2M

error_reporting = E_ALL

PHP Session Management

When logging into a PHP application you can view your cookies and likely identify a cookie with an name like 'phpsessid' and a value similar to 'bbbca6bb7a23bdc8de3baef2b506e654'. The cookie is composed of 32 hexadecimal characters.

PHP Session configuration

session.entropy_file = "/dev/urandom"
session.name = SESSID
session.save_path = /var/lib/php
session.cookie_httponly = 1
session.referer_check = your domain
session.cookie_lifetime = 0
session.cookie_secure = 1
session.cookie_httponly = 1
http://php.net/manual/en/session.configuration.php#ini.session.cookie-secure

Disable Dangerous functions

disable_functions = php_uname, getmyuid, getmypid, passthru, leak, listen, diskfreespace, tmpfile, link, ignore_user_abord, shell_exec, dl, set_time_limit, exec, system, highlight_file, source, show_source, fpaththru, virtual, posix_ctermid, posix_getcwd, posix_getegid, posix_geteuid, posix_getgid, posix_getgrgid, posix_getgrnam, posix_getgroups, posix_getlogin, posix_getpgid, posix_getpgrp, posix_getpid, posix, _getppid, posix_getpwnam, posix_getpwuid, posix_getrlimit, posix_getsid, posix_getuid, posix_isatty, posix_kill, posix_mkfifo, posix_setegid, posix_seteuid, posix_setgid, posix_setpgid, posix_setsid, posix_setuid, posix_times, posix_ttyname, posix_uname, proc_open, proc_close, proc_get_status, proc_nice, proc_terminate, phpinfo

Check Permissions remove undeed php modules

chmod -R 0444 /var/www/html/

php -m

mv /etc/php.d/sqlite3.ini /etc/php.d/sqlite3.disable

Use PHPIDS

- https://github.com/PHPIDS/PHPIDS


MySQL

MySQL Security Guidelines

- Do not all any account outside of root to access user table.
- Do not grant necessary privileges and grant privileges to all hosts.
- Keep MySQL behind a local firewall use stunnel or openvpn for outbound connection never expose it to system that don’t need access to it.
- Do not store password in clear text in MySQL databases
- Use Innodb over MyISAM

MySQL Security Guidelines Part 2

- Applications that access MySQL should not trust any data entered by users.
- Never run MySQL as system root
- Do not grant FILE privileges
- Do not permit use of symlinks to tables
- Use ip address instead of hostnames in grant tables
- Do not specify passwords on command line

Stunnel Client /etc/stunnel/stunnel.conf

cert = /etc/stunnel/stunnel.pem #https://www.stunnel.org/howto.html

client = yes

chroot=/var/run/stunnel

debug = 1

output=/var/run/stunnel.log

pid=/stunnel

sslVersion = all

options = NO_SSLv2

options = NO_SSLv3

ciphers = ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-DSS-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDH-RSA-AES256-GCM-SHA384:ECDH-ECDSA-AES256-GCM-SHA384:AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:ECDH-RSA-AES128-GCM-SHA256:ECDH-ECDSA-AES128-GCM-SHA256:AES128-GCM-SHA256

[mysql1]

accept = 127.0.0.1:3306

connect = 10.0.0.1:3307

Stunnel Server Config

```
cert = /etc/stunnel/stunnel.pem
chroot = /var/run/stunnel/
setuid = nobody
setgid = nobody
pid = /stunnel.pid
debug = 1
output = /var/log/stunnel.log
sslVersion = all
options = NO_SSLv2
options = NO_SSLv3
ciphers = ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-DSS-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDH-RSA-AES256-GCM-SHA384:ECDH-ECDSA-AES256-GCM-SHA384:AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:ECDH-RSA-AES128-GCM-SHA256:ECDH-ECDSA-AES128-GCM-SHA256:AES128-GCM-SHA256
[mysql]
accept  = 3307
connect = 3306
```

mysql_secure_installation

mysql secure installation will asked a series of questions, beginning with if you'd like to change the root password.

You should answer "Y" (for yes) to all of the remaining questions.

This will remove the ability for anyone to log into MySQL by default, disable logging in remotely with the administrator account, remove some test databases that are insecure, and update the running MySQL instance to reflect these changes.

My.ini Security

```
[MySQLD]
log=/var/log/mysql-logfile
bind-address = 127.0.0.1
local-infile=0
```

Implement Application-Specific Users

```
create database somedatabase;

CREATE USER 'someuser'@'localhost' IDENTIFIED BY 'agoodpassword';

GRANT SELECT,UPDATE,DELETE ON somedatabase.* TO 'someuser'@'localhost';
FLUSH PRIVILEGES;
show grants for 'someuser'@'localhost'

REVOKE UPDATE ON somedatasbase.* FROM 'someuser'@'localhost';
```

MySQL Stuff

View clear text queries

- tcpdump -l -i eth0 -w - src or dst port 3306 | strings

Clear MySql History

- cat /dev/null > ~/.mysql_history

Disable MySQL history

- ln -s /dev/null $HOME/.mysql_history

Resources

- http://www.tecmint.com/linux-server-hardening-security-tips/
- http://www.linuxweblog.com/tune-my.cnf
- http://www.w3resource.com/mysql/mysql-security.php
- https://www.digitalocean.com/community/tutorials/how-to-secure-mysql-and-mariadb-databases-in-a-linux-vps

Sources

Great Book! a little bit out of date

https://www.feistyduck.com/books/apache-security/

https://blog.sucuri.net/2014/03/understanding-denial-of-service-and-brute-force-attacks-wordpress-joomla-drupal-vbulletin.html

http://xmodulo.com/harden-apache-web-server-mod_security-mod_evasive-centos.html

https://www.owasp.org/index.php/Web_Application_Firewall