

HS - February 2015

How to secure a web server
~~like a boss~~ “comme un patron”

par Martin Dubé

Tonight's plan

- Security 5 Ws
- Scenario
- Strategy
 - ↗ Approach
 - ↗ Choosing an Operating System
 - ↗ Choosing a web server
 - ↗ Choosing a network topology
 - ↗ Choosing an architecture
- Implementation
- Test security of the solution (penetration test)

About me

- ➔ Daddy
- ➔ Hackfest admin since 2011
 - ↗ Worked hard on Hacking Games
 - ↗ All other kind of tasks
- ➔ Security Analyst at GoSecure
 - ↗ Penetration Testing
 - ↗ Systems Hardening
 - ↗ Firewall Management (Checkpoint & Fortinet)

[Security] What?

→ From OSSTMM

- ↗ “a form of protection where a separation is created between the assets and the threat.”

→ Assets

- ↗ The web application
- ↗ The infrastructure around

→ Threat

- ↗ Outsiders (Those who access the web site)
- ↗ Insiders (Those who manage the web site)

[Security] Who?

→ Everyone.

→ Example:

- ↗ Should the developer put time on security?
- ↗ Should the sysadmin put time on security?

[Security] Where?

- On all available layers
- Example:
 - ↗ Operating System (OS)?
 - ↗ Network?
 - ↗ Application vs Infrastructure
 - ↗ Client side?

[Security] When?

- Until it is considered enough
 - ↗ Depend on risk acceptance
 - ↗ Depend on budgets
- Depends on the criticality of the assets and interest for the threat
 - ↗ An extranet
 - ↗ A public web site
 - ↗ An internal web site
 - ↗ A VPN web portal

[Security] Why?

- Protect assets against Threats

[Security] How?

→ The following shall give some ideas.

Scenario

- We are a system admin
- We need to integrate a web app in the infrastructure
 - ↗ It is extremely insecure (let's say we know it)
 - ↗ It is extremely critical for the company
 - ↗ Do a shit load of things
 - ↗ Must work flawless

Scenario

- The need is clear. We know that:
 - ↗ The app is coded in PHP
 - ↗ The app must run shell commands (dafuq?)
 - ↗ The app must access several servers outside the network

Scenario

We won't waste time on:

- ➔ Performance

- Hardcore security sometimes means performance cost. Here we don't care.

- ➔ Monitoring

- This is essential for security but it would make a 5 hours presentation. :)

Strategy

Securing by component

- OS
- Web server
- PHP
- ⇒ ~~Database~~
- ⇒ ~~Application arch.~~
- ⇒ ~~Application source~~

Strategy

For each component

- Ask about their communications (in / out)
- Ask about DIC
 - ↗ Authenticity
 - ↗ Integrity
 - ↗ Disponibility
- Implement relevant security mechanism
 - ↗ Fuck Security by Obscurity.
- Understand what the hell you do.

[Strategy] Choosing a secure OS


- We want an OS that
 - ↗ Offer multiple relevant security mechanism
 - ↗ Is supported by an active community
 - ↗ Quick security fix
 - ↗ Is well documented

[Strategy] Choice: OpenBSD

→ Why?

- ↗ Is secure by default
 - ↗ “Four years without a remote hole in the default install!”
- ↗ Full of security mechanisms
 - ↗ Memory Protection (W^X, ProPolice, strcpy/strcat)
 - ↗ chroots (by default on some packages)
- ↗ Privilege separation, by default (51 user, 39 for low priv.)
- ↗ Powerful randomness (arc4random, libressl)
- ↗ Regular source code audits (6 to 12 members security team)
- ↗ Quick security updates
- ↗ No damn /proc :)

[Strategy] Choice: OpenBSD

DEFAULT SECURITY	
	
feature	OpenBSD
Random Stack Gap	default
W^X (GOT, PLT, ctors, dtors, .rodata, atexit)	default
ASLR (PIE, mmap, malloc)	default
Stack Smashing Protection	default, system wide
StackGhost (sparc64)	default
NULL page mapping	default
strcpy()/strcat()	default
Randomness/arc4random()	default
swap encrypted	default
LibreSSL	default
Privilege separation	default
Securelevel	default 1
systrace	available
Jails	not implemented
Mandatory Access Control framework	not implemented
pf version	latest, default

Thanks to: <http://networkfilter.blogspot.ca/2014/12/security-openbsd-vs-freebsd.html>

[Strategy] Choice: OpenBSD

- Securelevel: 1 (default)
 - ↗ - /dev/mem and /dev/kmem may not be written to raw disk devices of mounted file systems are read-only
 - ↗ - system immutable and append-only file flags may not be removed
 - ↗ - kernel modules may not be loaded or unloaded
 - ↗ - a panic or trap cannot be forced

[Strategy] Choice: OpenBSD

- ➔ Securelevel 2: "Highly secure mode"
 - all effects of securelevel 1
 - raw disk devices are always read-only whether mounted or not
 - `settimeofday(2)` and `clock_settime(2)` may not set the time backwards or close to overflow
 - firewall and NAT rules may not be altered

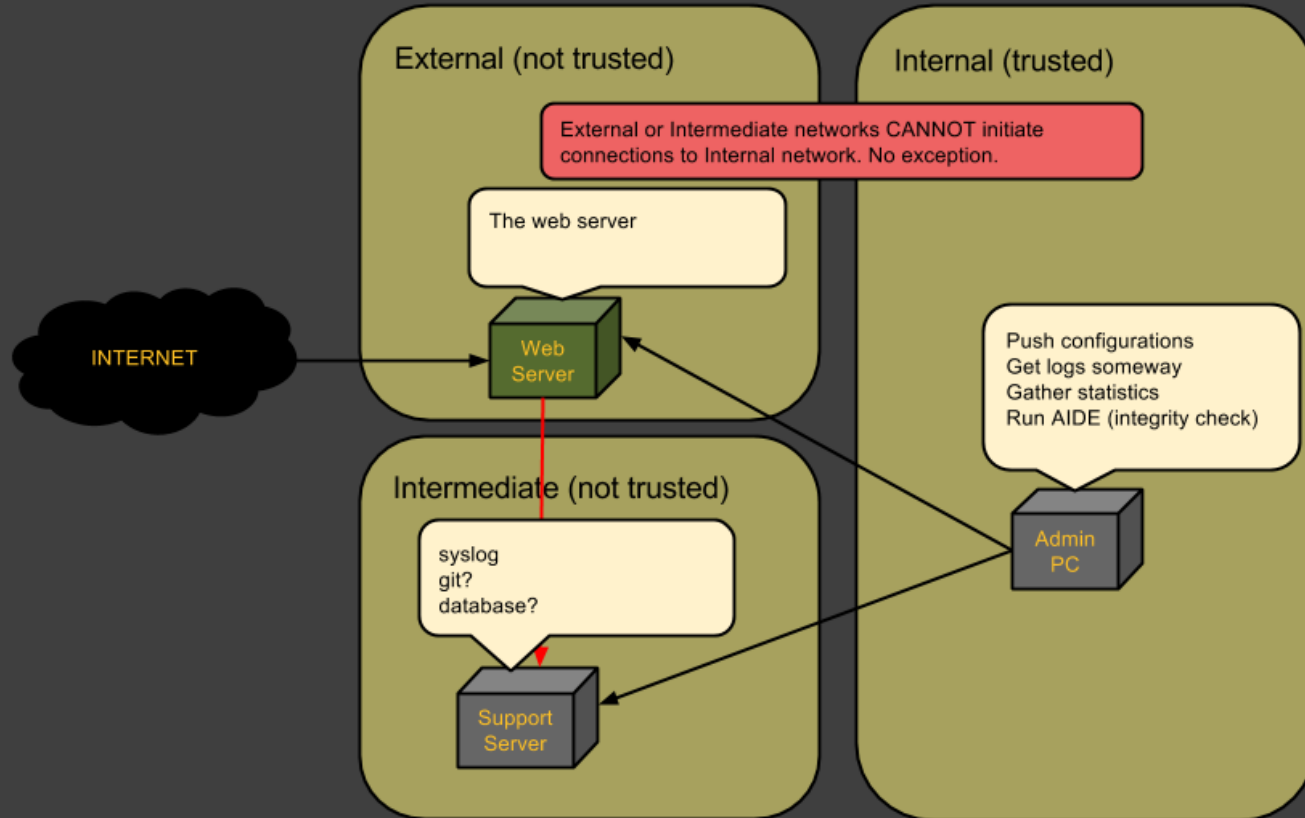
[Strategy] Choosing a web server

- ➔ Nginx vs Apache
- ➔ Both are mature and well documented
 - Nginx pros
 - Chrooted by default
 - Less vulnerabilities documented ([8 on nginx](#) vs [284 on apache](#) @ cvedetails)
 - Apache pros
 - mod_security
 - .htaccess
- ➔ Understanding how to use it is what matter!

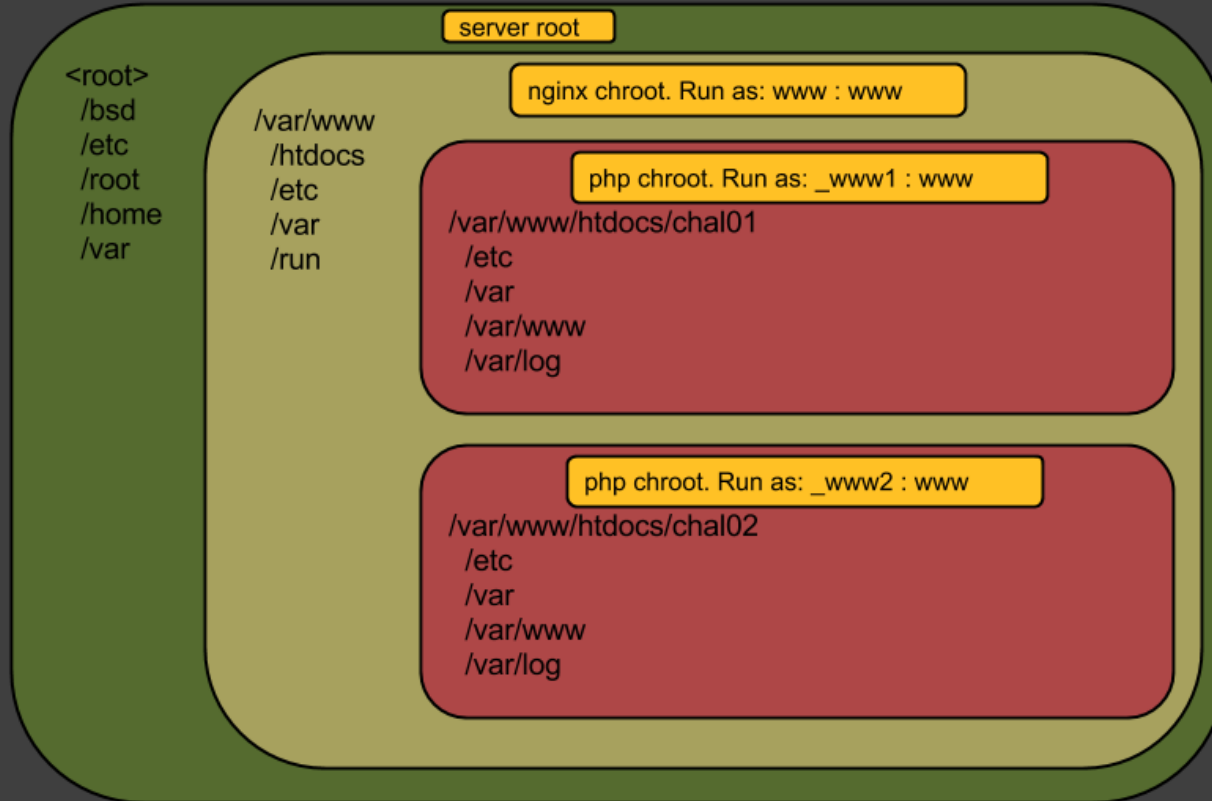
[Strategy] Choice: nginx

- ➔ Why?
 - ↗ For previously enumerated reasons.

[Strategy] Choosing a topology



[Strategy] Choosing an architecture



[Impl.] OS Installation

- Minimal Install. Enable only what's needed.

```
Set name(s)? (or 'abort' or 'done') [done] -game56.tgz
[X] bsd                [X] etc56.tgz          [ ] xbase56.tgz      [ ] xserv56.tgz
[X] bsd.rd             [X] comp56.tgz         [ ] xetc56.tgz
[ ] bsd.mp             [X] man56.tgz           [ ] xshare56.tgz
[X] base56.tgz         [ ] game56.tgz          [ ] xfont56.tgz
```

- During install
 - ↗ An admin user **can** be created
 - ↗ OpenSSH server **can** be installed
 - ↗ root login **can** be restricted on OpenSSH
 - ↗ A ntp server **can** be configured
- After install
 - ↗ Install your favourite tools

[Impl.] OS Hardening

- ➔ Filesystem integrity
 - Install, configure and run AIDE (Advanced Intrusion Detection Environment)

```
[WEB] root@obsdhs:~$ pkg_info -Q aide
aide-0.15.1
[WEB] root@obsdhs:~$ pkg_add aide
quirks-2.9 signed on 2014-07-31T22:37:55Z
aide-0.15.1: ok
[WEB] root@obsdhs:~$ vim /etc/aide.conf
[WEB] root@obsdhs:~$ aide --init
```

AIDE, version 0.15.1

AIDE database at /var/db/aide.db.new initialized.

```
116 /usr/bin R
117 /usr/include R
118 /usr/lib R
119 /usr/libdata R
120 /usr/libexec R
121 /usr/local/bin R
122 /usr/local/etc L+s+sha1
123 /usr/local/lib R
124 /usr/local/libexec R
125 /usr/local/sbin R
126 /usr/local/share R
127 /usr/sbin R
128 /usr/share R
129
130 =/tmp$ L
131
132 # root related
133 # by mdube
134 /root$ R
135
136 # Web related
137 # by mdube
138 /var/www/bin R
139 /var/www/conf R
140 /var/www/htdocs R
141 /var/www/cgi-bin R
...
```

[Impl.] OS Hardening

→ Filesystem integrity

- ↗ Don't forget to download the file at a safe place!
- ↗ Keep understanding what you are doing :)

```
#  
# AIDE 0.10  
#  
# example configuration file  
#  
# IMPORTANT NOTE!!! PLEASE READ  
#  
# This configuration file checks the integrity of the  
# AIDE package.  
#  
# This file is not intended to be used as the primary aide.conf file for  
# your system. This file is intended to be a showcase for different  
# features for aide.conf file.  
#  
# WRITE YOUR OWN CONFIGURATION FILE AND UNDERSTAND WHAT YOU ARE WRITING  
#  
#  
# Default values for the parameters are in comments before the  
# corresponding line.  
#  
_
```

[Impl.] OS Hardening

→ Filesystem integrity

- ↗ To check for changes on the file system:

- ↗ Upload latest baseline at `/var/db/aide.db`

- ↗ `aide --check`

- ↗ To update database with new files

- ↗ `aide --update`

[Impl.] OS Hardening

→ Password Policy

```
[WEB] root@obsdhs:/sbin$ pkg_add passwdqc  
quirks-2.9 signed on 2014-07-31T22:37:55Z  
passwdqc-1.3.0: ok  
--- +passwdqc-1.3.0 -----
```

To enable passphrase quality checking using pwqcheck(1) globally, add the following lines at the end the "default" class in login.conf(5).

```
:passwordcheck=/usr/local/bin/pwqcheck -1:\n:passwordtries=0:
```

[Impl.] OS Hardening

→ Password Policy

➤ Add these lines in /etc/login.conf

```
41 default:\n42     :path=/usr/bin /bin /usr/sbin /sbin /usr/X11R6/bin /usr/local/bin /usr/local/sbin:\n43     :umask=022:\n44     :datasize-max=512M:\n45     :datasize-cur=512M:\n46     :maxproc-max=256:\n47     :maxproc-cur=128:\n48     :openfiles-cur=512:\n49     :stacksize-cur=4M:\n50     :localcipher=blowfish,8:\n51     :ypcipher=old:\n52     :tc=auth-defaults:\n53     :tc=auth-ftp-defaults:\n54     :passwordcheck=/usr/local/bin/pwqcheck -1 config=/etc/passwdqc.conf:\n55     :passwordtries=0:\n56
```

[Impl.] OS Hardening

→ Password Policy

- ↗ Configure policy in `/etc/passwdqc.conf`

```
1 #min=disabled,24,11,8,7
2 min=disabled,50,25,20,16
3 #max=40
4 max=128
5 passphrase=3
6 match=4
7 similar=deny
8 random=47
9 enforce=everyone
10 retry=3
```

[Impl.] OS Hardening

→ Password Policy

↗ Test the policy

```
[WEB] root@obsdhs:~$ echo "Abcd123!" | pwqcheck -1 config=/etc/passwdqc.conf
Bad passphrase (too short)
[WEB] root@obsdhs:~$ echo "Abcdefghijklm1234567!" | pwqcheck -1 config=/etc/passwdqc.conf
Bad passphrase (based on a common sequence of characters and not a passphrase)
[WEB] root@obsdhs:~$ echo "Abcde$fghijklm123!4567!" | pwqcheck -1 config=/etc/passwdqc.conf
Bad passphrase (too short)
[WEB] root@obsdhs:~$ echo "Abcde$fghijklM123!4567!" | pwqcheck -1 config=/etc/passwdqc.conf
Bad passphrase (too short)
[WEB] root@obsdhs:~$ echo "Ab3cde$fghijklM123!45a67!" | pwqcheck -1 config=/etc/passwdqc.conf
Bad passphrase (too short)
[WEB] root@obsdhs:~$ echo "Ab3cde$fghijklM123!45a67!f" | pwqcheck -1 config=/etc/passwdqc.conf
Bad passphrase (too short)
[WEB] root@obsdhs:~$ echo "$$Ab3c1de$fghijklM123!45a67!f" | pwqcheck -1 config=/etc/passwdqc.conf
OK
```

[Impl.] OS Hardening

→ Server Management: OpenSSH

↗ Server side consideration

```
118 PermitRootLogin no      # Avoid logging in as root
119 PasswordAuthentication no # Force user to login with key
120 X11Forwarding no        # Avoid running GUI apps on the server from SSH
121 AllowTcpForwarding no    # Disable Forwarding
122
123 ChrootDirectory %h        # Chroot user in his home folder. Useful for files
124                          # uploads on a web site.
```

↗ Client side consideration

- ↗ Create a strong key (-t rsa -b 4096)
- ↗ Encrypt it with PKCS#8

[Impl.] OS Hardening

→ Partitions security

↗ Default partitions scheme

```
[WEB] root@obsdhs:~$ mount
/dev/wd0a on / type ffs (local)
/dev/wd0k on /home type ffs (local, nodev, nosuid)
/dev/wd0d on /tmp type ffs (local, nodev, nosuid)
/dev/wd0f on /usr type ffs (local, nodev)
/dev/wd0g on /usr/X11R6 type ffs (local, nodev)
/dev/wd0h on /usr/local type ffs (local, nodev)
/dev/wd0j on /usr/obj type ffs (local, nodev, nosuid)
/dev/wd0i on /usr/src type ffs (local, nodev, nosuid)
/dev/wd0e on /var type ffs (local, nodev, nosuid)
```

↗ Other interesting options: noexec, rdonly

[Impl.] OS Hardening

→ Logs export

↗ Distributed setup

↗ Authenticate and Encrypt communication with log server

↗ <http://cromwell-intl.com/cybersecurity/syslog-tls-cloud.html>

↗ Local setup

↗ Just send logs on a remote server

↗ Use a dedicated network

[Impl.] OS Hardening

→ Logs export

↗ Config on client

↗ /etc/syslog.conf

```
41 # Logs to send to obsdlogs
42 *. * @obsdlogs
```

↗ /etc/hosts

```
10 127.0.0.1 localhost
11 ::1 localhost
12 192.168.56.3 obsdlogs
```

↗ Config on server

↗ /etc/syslog.conf

```
42 +obsdhs
43 **.* /var/log/obsdhs.log
44 *.notice;local7,auth,authpriv,cron,ftp,kern,lpr,mail,user.none /var/log/obsdhs/messages
45 kern.debug;syslog,user.info /var/log/obsdhs/messages
46 auth.info /var/log/obsdhs/authlog
47 authpriv.debug /var/log/obsdhs/secure
48 cron.info /var/log/obsdhs/cron
49 daemon.info /var/log/obsdhs/daemon
50 ftp.info /var/log/obsdhs/xferlog
51 lpr.debug /var/log/obsdhs/lpd-errs
52 mail.info /var/log/obsdhs/maillog
```

↗ /etc/hosts

```
10 127.0.0.1 localhost
11 ::1 localhost
12 192.168.56.2 obsdhs
```

↗ /etc/rc.conf.local (Note: -u flag is considered

```
insecure.)
1 ntpd_flags=
2 syslogd_flags="-u"
```

[Impl.] OS Hardening

→ Firewall rules

```
49 # Block everything by default
50 block log all
51 block in quick from <abusive_ips>
52
53 # In: Web Access from management
54 pass in quick on $mgmt_int inet proto tcp from $mgmt_net to port $web_ports
55
56 # In: Public web access. Throttle web connections per second
57 # Max number of connections per source: 100
58 # Rate limit the number of connections to 15 in 5 second
59 pass in on $ext_int proto tcp to ($ext_int) port $web_ports flags S/SA keep state (max-src-conn 100, max-src-conn-
rate 15/5, overload <abusive_ips> flush)
60
61 # In: SSH Access
62 pass in quick on $mgmt_int inet proto tcp from $mgmt_net to port ssh
63
64 # Out: Syslog push
65 pass out quick on $mgmt_int inet proto udp from ($mgmt_int) to $log_host port 514
66
67 # Out: Repo access for packages download to openbsd.cs.toronto.edu
68 pass out quick on $ext_int inet proto tcp from ($ext_int) to $repo_hosts port $web_ports
```

[Impl.] OS Hardening

- If SSH must be publicly accessible
 - ↗ Implement fail2ban with PF
 - ↗ <http://www.bsdguides.org/2012/fail2ban-with-pf-on-openbsd-5-2/>

[Impl.] Web server Installation

→ Installation

```
[WEB] root@obsdhs:~$ pkg_add nginx-1.5.7p3
quirks-2.9 signed on 2014-07-31T22:37:55Z
nginx-1.5.7p3: ok
The following new rcscripts were installed: /etc/rc.d/nginx
See rc.d(8) for details.
Look in /usr/local/share/doc/pkg-readmes for extra documentation.
```

→ Remove “nodev” flag from /var

```
[WEB] root@obsdhs:~$ vim /etc/fstab
[WEB] root@obsdhs:~$ reboot
```

```
1 58093c6b0d750e4b.b none swap sw
2 58093c6b0d750e4b.a / ffs rw 1 1
3 58093c6b0d750e4b.k /home ffs rw,nodev,nosuid 1 2
4 58093c6b0d750e4b.d /tmp ffs rw,nodev,nosuid 1 2
5 58093c6b0d750e4b.f /usr ffs rw,nodev 1 2
6 58093c6b0d750e4b.g /usr/X11R6 ffs rw,nodev 1 2
7 58093c6b0d750e4b.h /usr/local ffs rw,nodev 1 2
8 58093c6b0d750e4b.j /usr/obj ffs rw,nodev,nosuid 1 2
9 58093c6b0d750e4b.i /usr/src ffs rw,nodev,nosuid 1 2
10 58093c6b0d750e4b.e /var ffs rw,nosuid 1 2
```

[Impl.] Web server Hardening

→ Remove default web site

↗ /etc/nginx/nginx.conf

↗ Delete: section http -> server

↗ Add: include conf.d/*.conf;

→ Configure logging

```
7 error_log      logs/error.log notice;  
8 error_log      syslog:server=unix:/dev/log,facility=local7,tag=nginx,severity=error notice;  
21 access_log     logs/access.log combined;  
22 access_log     syslog:server=unix:/dev/log,facility=local7,tag=nginx,severity=info combined;
```

→ Remove nginx version in errors

```
23 server_tokens off;          # Disable emitting nginx version
```

[Impl.] Web server Hardening

→ Set limits

```
10 worker_processes      1;          # Nb of CPU to use
11 worker_rlimit_nofile  1024;       # worker max number of opened files
12 events {
13     worker_connections 50;         # Max number of simultaneous connections by worker
14 }
```

→ Set more limits

```
# Size Limits & Buffer Overflows
client_body_buffer_size    1K;       # Default 8K or 16K
client_header_buffer_size  1k;       # Increase if large cookies
client_max_body_size       1k;       # Check content-length. if exceed: 413 Request Entity Too Large
large_client_header_buffers 2 1k;    # Max number and size of buffers.
```


[Impl.] Web server Hardening

→ Set timeouts

```
# Timeouts
client_body_timeout 10; # Default: 60. Set the read timeout for the request body
                        # Send a "Request timeout" (408) if exceeded
client_header_timeout 10; # Default: 60. Set the read timeout for the request body
                        # Send a "Request timeout" (408) if exceeded
keepalive_timeout 5 5; # Default: 75. For this time, a keep-alive client connection
                        # will stay open on the server side
send_timeout 10; # Default: 60. Sets a timeout for transmitting a response to the
                  # client
```

[Impl.] Web server Hardening

→ Set max simultaneous connections

```
# Control simultaneous connections
limit_conn_zone $binary_remote_addr zone=conn:5m;
# Then use "limit_conn conn <NUMBER OF CONN>;" in server section
```

→ Set max concurrent connections

```
# Limit number of request
limit_req_zone $binary_remote_addr zone=req:10m rate=1r/s;
# Then use "limit_req zone=req burst=10 nodelay;" in server section
```

[Impl.] Web server Hardening

- Limit access to our domain(s)

```
# Allow access to domain names only
if ($host !~ ^(obsdhs|obsdhs.hf)$ ) {
    return 444;
}
```

- Limit Request Methods

```
# Only allow these request methods
if ($request_method !~ ^(GET|HEAD|POST)$ ) {
    return 404;
}
```

[Impl.] Web server Hardening

→ Block Referrer (Spam)

```
# Deny certain Referers
if ( $http_referer ~* (babes|forsale|girl|jewelry|love|nudit|organic|poker|porn|sex|teen) ){
    # return 404;
    return 403;
}
```

→ Block Image Hotlinking

```
# Stop deep linking or hot linking
location /images/ {
    valid_referers none blocked www.example1.com www.example2.com;
    if ($invalid_referer) {
        return 403;
    }
}
```

[Impl.] Web server Hardening

- Limit access to some folder/files by password

```
# Protect the admin section
# Set password with: htpasswd -c /var/www/conf/.htpasswd mdube
location ~ /admin/*|wp-admin* {
    auth_basic "Restricted";
    auth_basic_user_file /var/www/conf/.htpasswd;
}
```

- Limit access to some folder/files by IP

```
# Turn on stats
# Allow only from 1 IP.
location /status {
    stub_status on;
    access_log off;
    allow 192.168.56.1/32;
    deny all;
}
```

[Impl.] Web server Hardening

→ TLS

```
69 # Just an example of secure TLS implementation
70 server {
71     listen      443;
72
73     # Enable HSTS
74     add_header Strict-Transport-Security "max-age=2678400; includeSubdomains;";
75
76     ssl          on;
77     ssl_certificate      /etc/ssl/srv.https.scoreboard.crt;
78     ssl_certificate_key  /etc/ssl/srv.https.scoreboard.key;
79
80     ssl_session_timeout  5m;
81     ssl_session_cache    shared:SSL:10m;
82
83     ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
84
85     # Enable Perfect Forward Secrecy (PFS)
86     ssl_ciphers "HIGH:!aNULL:!MD5 or HIGH:!aNULL:!MD5:!3DES";
87
88     #ssl_prefer_server_ciphers on;
89
90 }
```

[Impl.] Web server Hardening

→ Secure logs

```
[WEB] root@obsdhs:/var/log/nginx$ ls -l
total 560
-rw-r--r--  1 root  wheel  182382 Feb 25 23:22 access.log
-rw-r--r--  1 root  wheel  101471 Feb 25 23:22 error.log
[WEB] root@obsdhs:/var/log/nginx$ chflags sappnd *
[WEB] root@obsdhs:/var/log/nginx$ echo "" > access.log
ksh: cannot create access.log: Operation not permitted
```

Remember what does Securelevel=1 ?

```
[WEB] root@obsdhs:/var/log/nginx$ chflags nosappnd *
chflags: access.log: Operation not permitted
chflags: error.log: Operation not permitted
```

[Impl.] PHP Installation

```
[WEB] root@obsdhs:~$ pkg_add php-fpm-5.5.14
quirks-2.9 signed on 2014-07-31T22:37:55Z
|No change in quirks-2.9Ambiguous: choose dependency for php-fpm-5.5.14:
  a      0: php-5.5.14p0
        1: php-5.5.14p0-ap2
Your choice: 0
php-fpm-5.5.14:libxml-2.9.1p1: ok
php-fpm-5.5.14:femail-0.98: ok
php-fpm-5.5.14:femail-chroot-0.98p2: ok
php-fpm-5.5.14:php-5.5.14p0: ok
php-fpm-5.5.14: ok
The following new rcscripts were installed: /etc/rc.d/php_fpm
See rc.d(8) for details.
Look in /usr/local/share/doc/pkg-readmes for extra documentation.
--- +php-5.5.14p0 -----
To enable the php-5.5 module please create a symbolic link from
/var/www/conf/modules.sample/php-5.5.conf to
/var/www/conf/modules/php.conf. As root:

    ln -sf /var/www/conf/modules.sample/php-5.5.conf /var/www/conf/modules/php.conf

The recommended php configuration has been installed to:
/etc/php-5.5.ini.
```


[Impl.] PHP Hardening

→ Remove default pool

- ↗ `/etc/php-fpm.conf`

- ↗ Comment section `[www]`

- ↗ Add: `include=/etc/fpm.d/*.conf`

→ Create and harden one php file per apps

- ↗ `php_value` vs `php_admin_value`

- ↗ The application cannot change its `php.ini` parameters with `php_admin_value` with `ini_set()`

[Impl.] PHP Hardening

→ Harden unix socket security

```
1 [chal01]
2 listen = /var/www/htdocs/chal01/var/run/php-fpm.sock
3 listen.owner = _www1 ; Unix socket owner
4 listen.group = www ; Unix socket group
5 listen.mode = 0660 ; Unix socket permissions
6 listen.backlog = -1
```

[Impl.] PHP Hardening

→ Harden process security

```
8 user = _www1 ; Process uid
9 group = www ; Process gid
10 chroot = /var/www/htdocs/chal01 ; Process chroot
11 catch_workers_output = yes ; Redirect worker stdout and stderr into main error log
..
```

→ Harden memory management

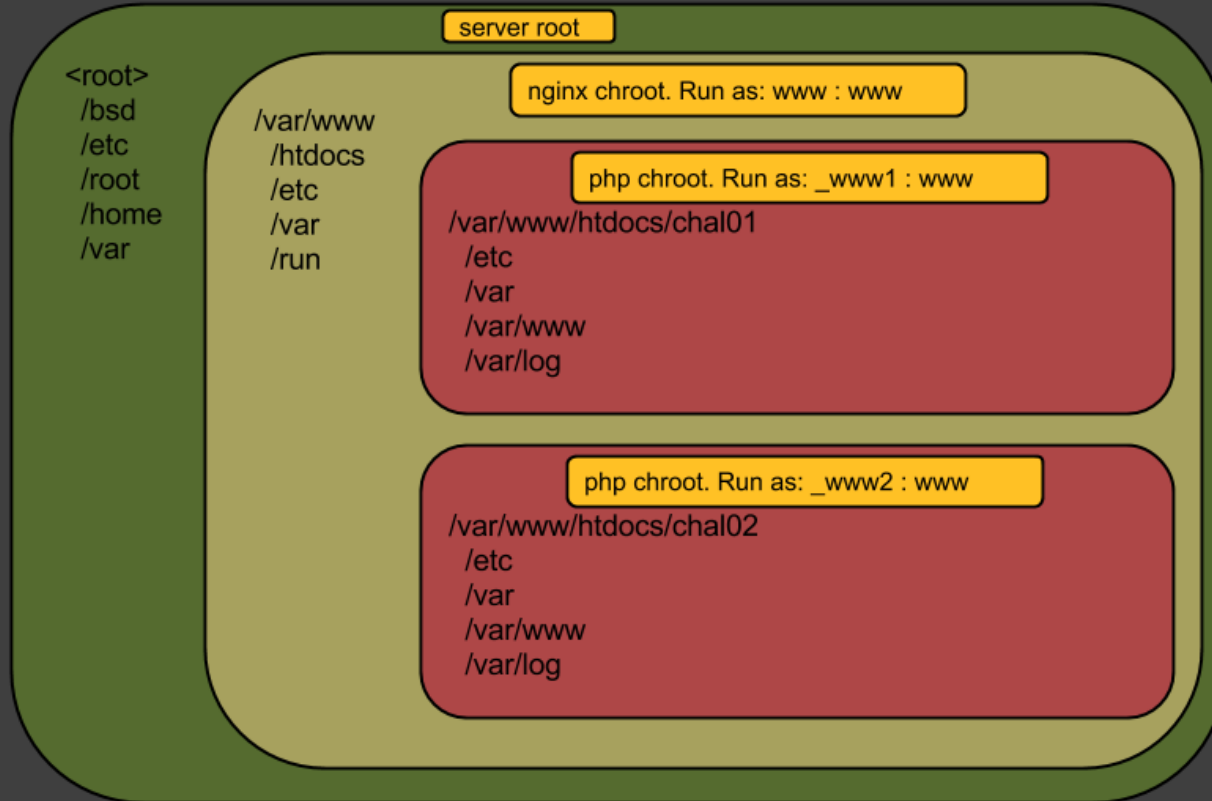
```
13 pm = dynamic ; Choose how the process manager will control child
14 pm.max_children = 10
15 pm.start_servers = 2
16 pm.min_spare_servers = 2
17 pm.max_spare_servers = 8
18 pm.status_path = /status
```

[Impl.] PHP Hardening

→ PHP ini file

```
36 php_admin_value[error_log] = /var/log/php-chal01-error.log
37 php_admin_value[disable_functions] = dl,exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,
38     parse_ini_file,show_source,include,include_once,require,require_once,file_get_contents,
39     readfile,fopen,fread,fwrite,fsockopen,socket_create,stream_socket_client,stream_socket_server
40 php_admin_value[max_execution_time] = 5 ; Maximum execution time of each script, in seconds
41 php_admin_value[max_input_time] = 60 ; Maximum amount of time each script may spend parsing request data
42 php_admin_value[memory_limit] = 8M ; Maximum amount of memory a script may consume (8MB)
43 php_admin_value[post_max_size] = 1M ; Maximum size of POST data that PHP will accept.
44 php_admin_value[file_uploads] = Off ; Whether to allow HTTP file uploads.
45 php_admin_value[upload_max_filesize] = 1M ; Maximum allowed size for uploaded files.
46 php_admin_value[display_errors] = Off ; Do not expose PHP error messages to external users
47 php_admin_value[safe_mode] = On ; Turn on safe mode
48 php_admin_value[safe_mode_exec_dir] = php-required-executables-path ; Only allow access to executables in isolated directory
49 php_admin_value[safe_mode_allowed_env_vars] = PHP_ ; Limit external access to PHP environment
50 php_admin_value[expose_php] = Off ; Restrict PHP information leakage
51 php_admin_value[log_errors] = On ; Log all errors
52 php_admin_value[register_globals] = Off ; Do not register globals for input data
53 php_admin_value[post_max_size] = 1K ; Minimize allowable PHP post size
54 php_admin_value[cgi.force_redirect] = 0 ; Ensure PHP redirects appropriately
55 php_admin_value[sql.safe_mode] = On ; Enable SQL safe mode
56 php_admin_value[allow_url_fopen] = Off ; Avoid Opening remote files
57 php_admin_value[allow_url_include] = Off ; Avoid Opening remote files
58 php_admin_value[include_path] = . ; Smallest path possible
```

[Impl.] Application Integration



[Impl.] File/Folder Security?

→ Whereas **php** run as **_www1:www** and **nginx** run as **www:www**

➤ Is this acceptable?

⚡ no

```
drwxrwxrwx  2 _www1  www      512 Feb 24 20:43 .
-rw-rw-rw-  1 _www1  www    153307 Feb 24 20:15 index.php
-rw-rw-rw-  1 _www1  www       20 Feb 22 11:10 phpinfo.php
```

➤ Is this enough?

⚡ no

```
drwxr-x---  2 _www1  www      512 Feb 24 20:43 .
-rw-r----- 1 _www1  www    153307 Feb 24 20:15 index.php
-rw-r----- 1 _www1  www       20 Feb 22 11:10 phpinfo.php
```

➤ Is this secure?

⚡ yes

```
drwxr-x---  2 root  www      512 Feb 24 20:43 .
-rw-r----- 1 root  www    153307 Feb 24 20:15 index.php
-rw-r----- 1 root  www       20 Feb 22 11:10 phpinfo.php
```

[Impl.] More Hardening

→ User / Groups / Files security

➤ Make /tmp be writable but not readable

➤ `chmod 730 tmp`

➤ Verify that php user (`_www1`) run as gid 67 (www)

```
[WEB] root@obsdhs:/var/www/htdocs$ ps -U _www1 -aux -o gid
```

USER	PID	%CPU	%MEM	VSZ	RSS	TT	STAT	STARTED	TIME	COMMAND	GID
_www1	25859	0.0	0.6	10064	6148	??	S	7:27PM	0:00.14	php-fpm-5.5: poo	67
_www1	16018	0.0	0.6	10064	6144	??	S	7:27PM	0:00.10	php-fpm-5.5: poo	67
_www1	13068	0.0	0.6	10064	6132	??	S	7:28PM	0:00.11	php-fpm-5.5: poo	67

[Impl.] Still need more?

→ systrace demo

```
$ systrace -i -A -d /etc/systrace -E /var/log/systrace.log /bin/ksh
```

```
$ ls -l
```

Step 1: Create policy

```
total 3580
-r-xrwx--- 1 0 67 111824 Feb 26 02:03 cat
-r-xrwx--- 1 0 67 11352 Feb 26 02:09 id
-r-xrwx--- 1 0 67 443600 Feb 26 01:53 ksh
-r-xrwx--- 1 0 67 246992 Feb 26 02:03 ls
-r-xrwx--- 1 0 67 443600 Feb 26 01:53 sh
-r-xrwx--- 1 0 67 8544 Feb 26 01:55 sh_systrace
-r-xrwx--- 1 0 67 8544 Feb 26 01:58 sh_systraceA
-r-xrwx--- 1 0 67 427216 Feb 26 01:53 systrace
```

```
$ ^D
```

```
$ systrace -i -a -d /etc/systrace -E /var/log/systrace.log /bin/ksh
```

```
$ ls
```

Step 2: Apply policy

```
/bin/ksh: ls: Operation not permitted
```

```
$ ls -l
```

Step 3: Test policy

```
total 3580
-r-xrwx--- 1 0 67 111824 Feb 26 02:03 cat
-r-xrwx--- 1 0 67 11352 Feb 26 02:09 id
-r-xrwx--- 1 0 67 443600 Feb 26 01:53 ksh
-r-xrwx--- 1 0 67 246992 Feb 26 02:03 ls
-r-xrwx--- 1 0 67 443600 Feb 26 01:53 sh
-r-xrwx--- 1 0 67 8544 Feb 26 01:55 sh_systrace
-r-xrwx--- 1 0 67 8544 Feb 26 01:58 sh_systraceA
-r-xrwx--- 1 0 67 427216 Feb 26 01:53 systrace
```

"ls" : Fail

"ls -l" : Success

[Impl.] Still need more?

→ systrace logs!

```
systrace: deny user: unknown(2001), prog: /bin/ksh, pid: 14714(0)[26867], policy: /bin/ksh, filters: 52, syscall: native-execve(59), filename: /bin/ls, argv: ls
```

Pentest Time!

- Challenge #1
 - ↗ Find the flag file and read its content
- Challenge #2
 - ↗ Exploit an eval()
- URL: `http://192.168.1.103/`

Thanks!

References

- <http://www.tldp.org/HOWTO/Chroot-BIND-HOWTO-2.html>
- <http://www.cyberciti.biz/tips/linux-unix-bsd-nginx-webserver-security.html>
- <http://cromwell-intl.com/cybersecurity/syslog-tls-cloud.html>
- <http://networkfilter.blogspot.ca/2014/12/security-openbsd-vs-freebsd.html>
- <http://www.cyberciti.biz/tips/php-security-best-practices-tutorial.html>
- <http://martin.kleppmann.com/2013/05/24/improving-security-of-ssh-private-keys.html>
- <http://www.openbsd.org/faq/pf/config.html>