Security Considerations

Security maintenance is important for the longevity of a Hub. This manual will discuss security suggestions our experts believe recommended reads are:

- Operating System Hardening Guide
- Hardening the Content Management System

Operating System Hardening Guide

Host Configuration

HUBzero software needs to be installed on a secured base: hardened OS and services. The following is a description of most of what we do at Purdue to harden the HUBs we manage.

- Software updates. Debian 6 will have security updates until May 2014. Look for patches and apply them daily:

  ```
  aptitude update;  aptitude upgrade
  ```

  Check for failed or incomplete package updates in a cron job running daily. We once have had a compromise due to a failed package security update which would have been detected by this check. Since then we've run it every day:

  ```
  dpkg --audit
  ```

  Check for any deviations from packages in a cron job running at least weekly:

  ```
  debsums | grep FAILED
  ```

  Make sure that the HUBzero repository is configured as a source of updates. Our updates also modify and fix issues with Joomla! itself. We provide support for Joomla! 1.5 through this repository. Even though it was EOL by the original authors, we effectively forked it and are maintaining it until the next HUBzero release.
- Configure Apache with the suhosin application firewall (package php5-suhosin), or mod-security. An application firewall is to PHP and the HUBzero CMS what a network firewall is to an operating system. They can block attacks targetting an existing code vulnerability, even if only the attacker is aware of the vulnerability. They can also limit or mitigate abuse. Suhosin is known to have protected against so-called "0-day" attacks. In addition, log messages from suhosin can be used as a basis for a fail2ban jail (see fail2ban paragraph)
- Configure Apache with mod-spamhaus (package libapache2-mod-spamhaus). The Spamhaus organization is very effective in finding IP addresses used for spamming. Blocking those from submitting content to the web site, while still allowing affected users to browse, will decrease spam questions and other forms of spam. In addition, it alerts users effectively that their machine is likely compromised. This is the message we show when someone is blocked:

  ```
   Access Denied! Your address is blacklisted. It could be because
  your computer is infected and participates in a spam botnet. If y
  ou're using a shared access point (e.g., wireless), it's possible
   that the IP address of that access point has been banned because
   someone else's computer is infected.  You can find the IP addres
  s of that device by going to http://whatismyipaddress.com/ .  You
  ```

```
 can find the reason for blacklisting the address by going to htt
p://www.spamhaus.org/lookup/  You will regain full access after c
orrecting the situation and removing the IP address from the blac
klist.
```

Note that commercial users need to obtain a datafeed from Spamhaus.

- Configure Apache to redirect all plain HTTP connections to HTTPS. This will mitigate issues such as pages presenting mixed HTTP and HTTPS content, or accepting plain HTTP connections to areas that require a login.
- Install Dshield's blocklist in the firewall iptables. dshield.org's blacklist is available at:

  ```
  http://dshield.org/block.txt
  ```

  and the signature used to verify its integrity is at:

  ```
  http://dshield.org/block.txt.asc
  ```

- Install spamassassin and tune it.
- Install an antivirus like ClamAV and make sure that HUBzero is configured to use it to scan all uploads on the fly. Use the EICAR test file to verify that it is operating properly:

  ```
  http://www.eicar.org/86-0-Intended-use.html
  ```

  and keep a copy on the web site. Scan the entire web site with ClamAV at least weekly. Make sure the "fresh-clam" daemon is running all the time and restart it as necessary (we do it every day to make sure) to get the most up-to-date malware definitions.
- Install fail2ban and configure it with jails for WebDAV, SSH, Apache errors, Apache suhosin messages, HUBzero CMS logins (in /var/log/hubzero/cmsauth.log), exim4, and spamassassin logs. We permanently ban anyone trying to login as root and some other key accounts. SSH is configured to deny root logins with passwords, but we let them try anyway so we can better ban attackers.
- Run a configuration management software that will automatically detect and report any unauthorized configuration changes. We run something called "Ogre" every hour. The Ogre core engine itself is open-source, but it needs meta-data and file templates to be useful. Those are not open-source at this time.
- Configure file system permissions such that the Apache user (www-data) can't modify the HUBzero and Joomla! code directories, by setting ownership to a different user. We're currently testing and deploying this.
- Run the auditing tool "Lynis" (package "lynis") and obtain a hardening score of at least 72.
- Run "rkhunter" at least every day to detect suspicious changes.
- Add firewall rules to block IPv6 on all interfaces but local, unless you also deploy an IPv6 equivalent for fail2ban and dshield. Due to the large address space of IPv6 and its privacy extension, blocking individual IPs is pointless. Any blocking tool worth using must have the capability to block networks. Blocking networks may also block innocent users, so it is important that the tool scales the size of the block, from individual IPs to

large networks, depending on the number of failures. We do not accept IPv6 traffic due to this connundrum; blocking misbehaving IP addresses is too useful a security measure to consider offering service without that capability. Tools that offer IPv6 address blocking capabilities all seem to be attempting to block individual IPs or fixed network sizes at this time; we believe that they are not useful.
- Consider recommendations from the PHP auditor to change php.ini settings. It's very easy to install, but some recommendations aren't practical and are incompatible with HUBzero functionality. You can get it, and see screenshots, from https://www.idontplaydarts.com/2011/02/hardening-and-securing-php-on-linux/

External Hardening Tools

- Run network-based vulnerability scans periodically. Note that Debian doesn't increase the version numbers in banners when applying security patches. This makes it very difficult to draw any conclusions from a vulnerability scanning engine like Nessus, using default settings. Most results are false positives. However, the scan can be useful to detect new open ports and some configuration issues.
- Make sure that the site is rated "A" by the Qualys SSL Labs server test at:

  `https://www.ssllabs.com/ssltest/index.html`

Hardening the Content Management System

Application Scanning

A web application scanner will look for typical mistakes made in PHP applications: XSS, CSRF and SQL injections, and more. We use AppScan, but many free application scanners are available. You should scan any component you code yourself, or 3rd party component, that is not part of the HUBzero release. Also, if you modify a component in a HUBzero release, you should scan it for vulnerabilities the change may have introduced. If you find a vulnerability in the HUBzero release itself, please file a ticket at hubzero.org!

CMS-Controlled Fail2Ban Jail

Configuration and Details

Objective

To have the CMS handle user login banning in an attempt to deter brute force attacks.

CMS Configuration Page

The following settings are accessible from the CMS administrative backend.

These options are accessible by going to the User Menu Members and Member Options.

User Password Reset Limit

The number of password resets per time period is limited. If the user attempts to reset their password at a threshold deemed by the HUB administrator as excessive, the following message is displayed.

User Failed Login Limit

The number of failed logins per time period is limited. If the user attempts to reset their password at a threshold deemed by the HUB administrator as excessive, the following message is displayed. This means that an individual's account is temporarily blocked until the time period expires.

IPbased Blocked User Limit

When the threshold of blocked user accounts per IP network is met, the CMS will trigger a Fail2Ban rule which will block incoming requests from an IP address for a period of time. This is the last line of defense as blocking an IP address may have unintended consequences such as

blocking a NATed IP address which several valid users are using to access the hub.

Assumptions

This approach assumes that the system administrator has configured a jail and a system user account to execute (with sudo) Fail2Ban via the fail2banclient utility.

On Debian Hosts, Fail2Ban should be version 0.9.5.

0.9.51~nd70+1 from http://neuro.debian.net/debian/ wheezy/main amd64 Packages

Setup & Configuration

There are a number of subsystems which need to be configured for this scheme to work properly.

sudo Configuration

A privileged user which can execute:

(root) NOPASSWD: /usr/bin/fail2banclient set hublogin banip [09.]* (root) NOPASSWD: /usr/bin/fail2banclient set hublogin unbanip [09.]*

This can be accomplished by adding a sudoers rule in /etc/sudoers.d/ that looks like:

wwwdata ALL=(root)NOPASSWD: /usr/bin/fail2banclient set hublogin banip [09.]*

Fail2Ban Configuration

The system administrator should configure Fail2Ban to create jails which the CMS can add offending IP address into. The amount of time that the ban is valid is configured in Fail2Ban.

The CMS will simply add IP addresses to the Fail2Ban jail which will trigger the Ban Action as specified in the rule set.

The following configuration are more of an example than anything. A seasoned system administrator will have crafted better rules.

[Sample] Jail Configuration

#

JAILS

/etc/fail2ban/jail.local

#

[hublogin] enabled = true

port = http,https filter = hublogin

logpath = /var/log/messages banaction = hubloginfailure bantime = 600

findtime = 1

maxretry = 1

[Sample] Filter Configuration

/etc/fail2ban/filter.d/hublogin.conf

```
# Fail2Ban configuration file

#

[Definition]




# Option:  failregex

# Notes.:  Regexp to catch known spambots and software alike. Please verify

#          that it is your intent to block IPs which were driven by

#          abovementioned bots.

# Values:  TEXT

#


#We choose something that will never happen

# Since the CMS will control IP’s placed in the jails failregex =
^<HOST>thisfilterwillneverbefound


# Option:  ignoreregex

# Notes.:  regex to ignore. If this regex matches, the line is ignored.

# Values:  TEXT

#

ignoreregex =
```

[Sample] Action Configuration

Fail2Ban configuration file

cat /etc/fail2ban/action.d/hubloginfailure.conf [INCLUDES]

before = iptablescommon.conf

[Definition]

Option: actionstart

Notes.: command executed once at the start of Fail2Ban.

Values: CMD

#

actionstart = iptables N fail2banhublogin

iptables A INPUT j DROP

iptables I INPUT p tcp j fail2banhublogin

Option: actionstop

Notes.: command executed once at the end of Fail2Ban

Values: CMD

#

actionstop = iptables D fail2banhublogin p tcp j fail2banhublogin iptables F fail2banhublogin

iptables X fail2banhublogin

Option: actioncheck

Notes.: command executed once before each actionban command

Values: CMD

#

actioncheck = iptables n L fail2banhublogin | grep q fail2banhublogin

Option: actionban

Notes.: command executed when banning an IP. Take care that the

command is executed with Fail2Ban user rights.

Tags: <ip> IP address

<failures> number of failures

<time> unix timestamp of the ban time

Values: CMD

#

actionban = iptables I fail2banhublogin p tcp dport 443 s <ip> j DROP

iptables I fail2banhublogin p tcp dport

80 s <ip> j DROP

Option: actionunban

Notes.: command executed when unbanning an IP. Take care that the

command is executed with Fail2Ban user rights.

Tags: <ip> IP address

<failures> number of failures

<time> unix timestamp of the ban time

Values: CMD

#

actionunban = iptables D fail2banhublogin p tcp dport 443 s <ip> j DROP iptables D fail2ban-hublogin p tcp dport 80 s <ip> j DROP

[Init]

Defaut name of the chain

#

name = DEFAULT

Option: protocol

Notes.: internally used by config reader for interpolations.

Values: [tcp | udp | icmp | all] Default: tcp

#

protocol = tcp

Option: chain

Notes specifies the iptables chain to which the fail2ban rules should be

added

Values: STRING Default: INPUT chain = INPUT

[Sample] Banned IP address results root@example:/var/www/example# fail2banclient status hublogin Status for the jail: hublogin

| Filter

| | Currently failed: 0

| | Total failed: 4

| ` File list: /var/log/messages

` Actions

| Currently banned: 1

| Total banned: 4

` Banned IP list: 192.168.226.1

root@example:/var/www/example# iptables L Chain INPUT (policy DROP)

target prot opt source destination fail2banhublogin tcp anywhere anywhere ACCEPT all anywhere anywhere

ACCEPT all anywhere anywhere state RELATED,ESTABLISHED

ACCEPT	tcp		anywhere	anywhere	tcp	dpt:ssh
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:smtp
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:mysql
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:ldap
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:http
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:https
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:httpalt
ACCEPT	tcp		anywhere	anywhere	tcp	dpts:830:831
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:http
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:https
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:httpalt
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:1170

ACCEPT	icmp		anywhere	anywhere		
DROP	all		anywhere	anywhere		

Chain FORWARD (policy DROP)

target ACCEPT ACCEPT	prot all all	opt	source 10.0.0.0/8 anywhere	destination anywhere anywhere	ctstate	
RELATED,ESTABLISHED,DNAT						
ACCEPT	tcp		anywhere	anywhere	tcp	dpts:830:831
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:http
ACCEPT	tcp		anywhere	anywhere	tcp	dpt:https
ACCEPT	udp		anywhere	anywhere	udp	dpt:domain

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

Chain fail2banhublogin (1 references)					
target	prot opt source		destination		
DROP	tcp	container.localhost	anywhere	tcp	dpt:http
DROP	tcp	container.localhost	anywhere	tcp	dpt:https

root@example:/var/www/example#

User Impact

When a large number of people intend on using the CMS, it may be wise to temporarily disable this feature (e.g. conference, class activity, etc). In the past, many conference goers have mistyped their password in a short period of time creating a false positive for normal Fail2Ban operation. This risk is mitigated by the fact that the number of blocked users is observed before triggering Fail2Ban.