GNU/Linux Administration - Feature #343

Configuring Apache To Use Perfect Forward Secrecy

02/26/2014 01:08 PM - Daniel Curtis

Status:	Closed	**Start date:**	02/26/2014
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Web Server	**Estimated time:**	1.00 hour
Target version:		**Spent time:**	1.00 hour

Description

This is a recipe for Apache SSL configuration that achieves perfect forward secrecy while avoiding other pitfalls such as the BEAST attack.

- First, SSLv2 is vulnerable, so disable it. On my Ubuntu box this was already done in ssl.conf:

```
#! enable only secure protocols: TLSv1, but not SSLv2 and SSLv3
SSLProtocol all -SSLv2
```

- Second, tell the browser to pay attention to the order ciphers are specified in:

```
SSLHonorCipherOrder On
```

- Next, compose the cipher list. The BEAST attack against how SSLv3 and TLSv1.0 do cipher block chaining makes most of the otherwise good ciphers (e.g. AES) vulnerable, leaving only the weaker RC4 as a viable option for those protocols. That is easier said than done, since Apache doesn't allow conditional cipher list control based on protocol, and one can't simply disable those protocols because browser support for TLS v1.1 and higher is still weak. As a proxy for checking the protocol version I therefore I resort to preferring ciphers that were only introduced after TLSv1.0. TLSv1.1 didn't introduce anything new, but TLSv1.2 added new hashing algorithms, AEAD, SHA384, SHA256; prior to that, AES was only available with SHA1 hashing). Thus the first organizational principal of the list is: TLSv1.2 and above, followed by RC4, followed by older protocols.
  Perfect forward secrecy is achieved by using ephemeral Diffie-Hellman ,+EDH+. Ephemeral elliptic-curve Diffie-Hellman , EECDH, is reasonably fast, so I prefer it. Otherwise EDH is slow; consider omitting if you're serving a lot of traffic on limited hardware. Thus the second organizational principal is: use each cipher only in combination with EECDH or plain EDH. (But prefer to relinquish perfect forward secrecy before being vulnerable to BEAST.)
  Finally, for good hygiene, explicitly disable anything using no authentication, !aNULL, no or weak encryption, !eNULL, !EXP, !LOW, or weak hashing, !MD5.
  The cipher list thus is:

```
SSLCipherSuite EECDH+AES:EDH+AES:-SHA1:EECDH+RC4:EDH+RC4:RC4-SHA:EECDH+AES256:EDH+AES256:AES25
6-SHA:!aNULL:!eNULL:!EXP:!LOW:!MD5
```

- The cipher list for apache 2.4+ and support for ECDSA and elliptic curve

```
SSLCipherSuite "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384 EECDH+ECDSA+SHA256 EEC
DH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !M
D5 !EXP !PSK !SRP !DSS"
```

The syntax for the cipher list is the same as for the openssl ciphers command. Of note, the leading "-" in -SHA1 means remove any ciphers with SHA1 hashing that had been previously added, whereas RC4-SHA is just the name of a particular cipher.

Unfortunately, older versions of Apache might not include all of these. E.g. Apache 2.2 on Ubuntu 12.04 LTS lacks EECDH (and there is no EDH RC4 variant). Thus in practice most browsers would use RC4 without perfect forward secrecy (but at least no BEAST vulnerability). The solution is to get a newer version of Apache, either by waiting for Ubuntu 13.10 obtaining it elsewhere. Configuration can be tested easily via SSLLabs.

Update 11-09-2013:

I've found a few alternate recommendations around the web. They put less emphasis on BEAST protection (perhaps wise; BEAST is mostly mitigated client-side now) and more emphasis on perfect forward secrecy. To varying degrees they also have stronger preferences for GCM and greater reluctance to accept RC4.

Personally, I'm going to go with Mozilla OpSec's. Their reasoning is well explained on their page. Of note, they prefer AES128 over AES256. In their words: "[AES128] provides good security, is really fast, and seems to be more resistant to timing attacks."

Noteworthy in Ivan Ristic's and Geoffroy Gramaize's recommendation is that SSLv3 is disabled. I think this mostly just breaks IE6, though some security related differences between SSLv3 and TLS v1.0 are mentioned on Wikipedia.

- Also before I didn't talk about CRIME and BREACH. To protect against CRIME, disable SSL compression. This is included in the examples linked. To protected against BREACH, you need to disable compression at the HTTP level. For Apache 2.4, just do this once globally:

```
<Location />
SetEnvIfExpr "%{HTTPS} == 'on'" no-gzip
</Location>
```

- For older versions of Apache, place this in each VirtualHost where SSLEngine is on:

```
<Location />
SetEnv no-gzip
</Location>
```

History

#1 - 02/26/2014 01:32 PM - Daniel Curtis

- Description updated

#2 - 04/17/2014 07:45 PM - Daniel Curtis

- Status changed from In Progress to Closed

NOTE: Switch to Nginx or a newer version of Apache. Maybe switch to Arch Linux or FreeBSD.

#3 - 02/16/2015 12:01 PM - Daniel Curtis

- Project changed from 40 to GNU/Linux Administration

- Description updated

- Category set to Web Server