How to Secure Apache on FreeBSD against Range header DoS vulnerability (affecting Apache 1.3/2.x)

Author : admin

Recently has become publicly known for the serious hole found in all Apache webserver versions 1.3.x and 2.0.x and 2.2.x. The info is to be found inside the security **CVE-2011-3192** - *https://issues.apache.org/bugzilla/show_bug.cgi?id=51714*

Apache remote denial of service is already publicly cirtuculating, since about a week and is probably to be used even more heavily in the 3 months to come. The exploit can be obtained from **exploit-db.com** a mirror copy of #Apache httpd Remote Denial of Service (memory exhaustion) is for download here

The DoS script is known in the wild under the name **killapache.pl**
killapache.pl PoC depends on **perl ForkManager** and thus in order to be properly run on FreeBSD, its necessery to install **p5-Parallel-ForkManager bsd port** :

```
freebsd# cd /usr/ports/devel/p5-Parallel-ForkManager
freebsd# make install && make install clean
...
```

Here is an example of the exploit running against an Apache webserver host.

```
freebsd# perl httpd_dos.pl www.targethost.com 50
host seems vuln
ATTACKING www.targethost.com [using 50 forks]
:pPpPpppPpPPppPpppPp
ATTACKING www.targethost.com [using 50 forks]
:pPpPpppPpPPppPpppPp
...
```

In about 30 seconds to 1 minute time the DoS attack with only 50 simultaneous connections is capable of overloading any vulnerable Apache server.

It causes the webserver to consume all the machine memory and memory swap and consequently makes the server to crash in most cases.
During the **Denial of Service** attack is in action access the websites hosted on the webserver becomes either hell slow or completely absent.

The **DoS** attack is quite a shock as it is based on an Apache range problem which started in year 2007.

Today, Debian has issued a new versions of Apache deb package for Debian 5 Lenny and Debian 6, the new packages are said to have fixed the issue.

I assume that Ubuntu and most of the rest Debian distrubtions will have the apache's range header DoS patched versions either today or in the coming few days.
Therefore work around the issue on debian based servers can easily be done with the usual **apt-get update && apt-get upgrade**

On other Linux systems as well as FreeBSD there are work arounds pointed out, which can be implemented to close temporary the Apache DoS hole.

1. Limiting large number of range requests

The first suggested solution is to limit the lenght of range header requests Apache can serve. To implement this work raround its necessery to put at the end of **httpd.conf** config:

```
# Drop the Range header when more than 5 ranges.
# CVE-2011-3192
SetEnvIf Range (?:,.*?){5,5} bad-range=1
RequestHeader unset Range env=bad-range
# We always drop Request-Range; as this is a legacy
# dating back to MSIE3 and Netscape 2 and 3.
RequestHeader unset Request-Range
# optional logging.
CustomLog logs/range-CVE-2011-3192.log common env=bad-range
CustomLog logs/range-CVE-2011-3192.log common env=bad-req-range
```

2. Reject Range requests for more than 5 ranges in Range: header

Once again to implement this work around paste in Apache config file:

This DoS solution is not recommended (in my view), as it uses mod_rewrite to implement th efix and might be additionally another open window for DoS attack as mod_rewrite is generally CPU consuming.

```
# Reject request when more than 5 ranges in the Range: header.
# CVE-2011-3192
#
RewriteEngine on
RewriteCond %{HTTP:range} !(bytes=[^,]+(,[^,]+){0,4}$|^$)
# RewriteCond %{HTTP:request-range} !(bytes=[^,]+(?:,[^,]+){0,4}$|^$)
RewriteRule .* - [F]

  # We always drop Request-Range; as this is a legacy
# dating back to MSIE3 and Netscape 2 and 3.
RequestHeader unset Request-Range
```

3. Limit the size of Range request fields to few hundreds

To do so put in httpd.conf:

```
LimitRequestFieldSize 200
```

4. Dis-allow completely Range headers: via mod_headers Apache module

In **httpd.conf** put:

```
RequestHeader unset Range
RequestHeader unset Request-Range
```

This work around could create problems on some websites, which are made in a way that the Request-Range is used.

5. Deploy a tiny Apache module to count the number of Range Requests and drop connections in case of high number of Range: requests

This solution in my view is the best one, I've tested it and I can confirm on FreeBSD works like a charm. To secure FreeBSD host Apache, against the Range Request: DoS using **mod_rangecnt**, one can literally follow the methodology explained in mod_rangecnt.c header:

```
freebsd# wget http://people.apache.org/~dirkx/mod_rangecnt.c
..
# compile the mod_rangecnt module
freebsd# /usr/local/sbin/apxs -c mod_rangecnt.c
...
# install mod_rangecnt module to Apache
freebsd# /usr/local/sbin/apxs -i -a mod_rangecnt.la
...
```

Finally to load the newly installed *mod_rangecnt*, Apache restart is required:

freebsd# /usr/local/etc/rc.d/apache2 restart
...

I've tested the module on i386 FreeBSD install, so I can't confirm this steps works fine on 64 bit FreeBSD install, I would be glad if I can hear from someone if mod_rangecnt is properly compiled and installed fine also on 6 bit BSD arch.

Deploying the mod_rangecnt.c Range: Header to prevent against the Apache DoS on **64 bit x86_amd64 CentOS 5.6 Final** is also done without any pitfalls.

[root@centos ~]# uname -a;
Linux centos 2.6.18-194.11.3.el5 #1 SMP Mon Aug 30 16:19:16 EDT 2010 x86_64 x86_64 x86_64 GNU/Linux
[root@centos ~]# /usr/sbin/apxs -c mod_rangecnt.c
...
/usr/lib64/apr-1/build/libtool --silent --mode=link gcc -o mod_rangecnt.la -rpath /usr/lib64/httpd/modules -module -avoid-version mod_rangecnt.lo
[root@centos ~]# /usr/sbin/apxs -i -a mod_rangecnt.la
...
Libraries have been installed in:
/usr/lib64/httpd/modules
...
[root@centos ~]# /etc/init.d/httpd configtest
Syntax OK
[root@centos ~]# /etc/init.d/httpd restart
Stopping httpd: [OK]
Starting httpd: [OK]

After applying the *mod_rangecnt* patch if all is fine the **memory exhaustion perl DoS script**'s output should be like so:

freebsd# perl httpd_dos.pl www.patched-apache-host.com 50
Host does not seem vulnerable

All of the above pointed work-arounds are only a temporary solution to these **Grave Apache DoS byterange vulnerability** , a few days after the original vulnerability emerged and some of the up-pointed work arounds were pointed. There was information, that still, there are ways that the vulnerability can be exploited.
Hopefully in the coming few weeks Apache dev team should be ready with rock solid work around to the severe problem.

In 2 years duration these is the second serious Apache Denial of Service vulnerability after before a one and a half year the so called Slowloris Denial of Service attack was capable to DoS most of the Apache installations on the Net.

Slowloris, has never received the publicity of the Range Header DoS as it was not that critical as the mod_range, however this is a good indicator that the code quality of Apache is slowly decreasing and might need a serious security evaluation.