Distributed Apache Killer

David Holmes, 2012-12-03

Remember the Apache Killer vulnerability from last year? It's an ugly little denial of service attack that sends chews up Apache's CPU cycles by requesting a series of nonsensical ranges of content.

```
HEAD / HTTP/1.1
```

```
Host:xxxx
```

```
Range:bytes=0-,5-1,5-2,5-3,…
```

To handle ranged requests, Apache distributes each range to a different worker process, significantly increasing the overall CPU load at the server. Lori MacVittie wrote an excellent blog detailing three different ways that F5 Networks' technology mitigates the problem.

Earlier this week, a Russian botnet *Armageddon* was seen using the "Apache Killer" range technique, so we can now consider this vulnerability *distributed* and it becomes an **application-layer DDoS attack**.

Apache developers had done some mitigation, but it was not necessarily complete. They've made some performance improvements but likely the root issue still exists in some form: there will always be pathways where an attacker can issue a small request that leads to large resource consumption (either CPU or bandwidth). The performance improvements may now be ***obsoleted*** by the fact that Apache Killer is distributed. Don't necessarily look to Apache for mitigation beyond disabling range processing on all your servers.

Also note that the Armageddon tool also incorporates other sophisticated application layer attacks, such as requesting well-known URLs that lead to heavy database queries.

Mitigation

Going back to Lori's original article; all three mitigation strategies using F5 technology are still valid now that Apache Killer is distributed.

1 HEADER SANITIZATION

First, you can modify the HTTP profile to simply remove the Range header. HTTP header removal – and replacement – is a common means of manipulating request and response headers as a means to "fix" broken applications, clients, or enable other functionality.

The beauty of this solution is that it's just a couple of clicks in the BIG-IP GUI. If you need a quick solution that doesn't involve editing httpd.conf on 200 web servers, there you go.

2 HEADER VALUE SCRUBBING

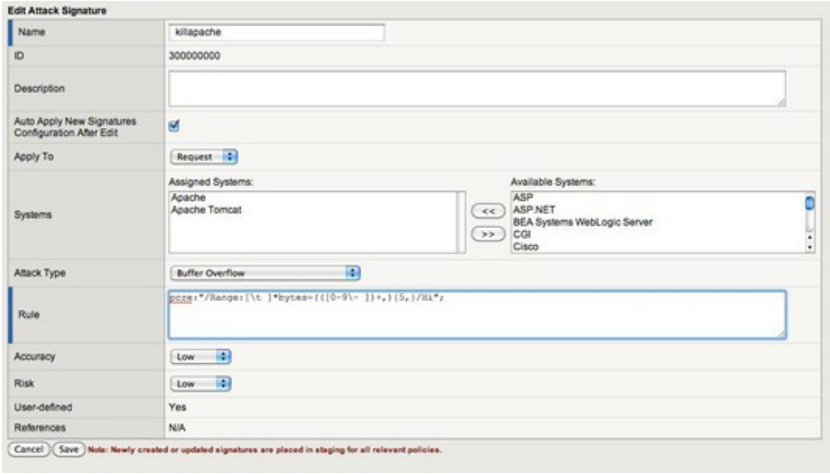
You can also use an iRule to scrub the headers. By inspecting and thus detecting large numbers of ranges in the RANGE header, you can subsequently handle the request based on your specific needs.

If you need customization (perhaps allowing a few range requests but denying more than five based on your own policy), this is your best bet. See Lori's original post for the 6-line iRule.

3 BIG-IP ASM ATTACK SIGNATURE

Use a BIG-IP Application Security Manager (ASM) attack signature to detect and act upon an attack using this technique.

Interesting, the Armageddon bot appears to incorporate techniques to foil the common JavaScript redirect counter-measure. When using the ASM attack signature be safe and just drop the connection.



Here are the different solutions that can be implemented by three different teams.

1. The HTTP Header Sanitization can be done by your network team.
2. The iRule could be implemented by your iRule team.
3. The ASM attack signature can be deployed by your security team.

The beauty of the *toolkit-aspect* of TMOS is that there are so many ways that you can manipulate data center traffic from a single point of control.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com