New Adwind Campaign targets US Petroleum Industry

netskope.com/blog/new-adwind-campaign-targets-us-petroleum-industry-2

Abhinav Singh October 1, 2019

A new campaign spreading the Adwind RAT has been seen in the wild, specifically targeting the petroleum industry in the US. The samples are relatively new and implement multi-layer obfuscation to try to evade detection. We found multiple RAT samples hosted on the serving domain and spread across multiple directories, all hosted within the last month. We have previously reported the use of this RAT targeting the retail and hospitality industry.

The overall functionality of the RAT has remained the same as our previous post: It achieves persistence through registry modifications, performs process injection to stay under the radar, terminates security services (*e.g.,* firewall, AV), and steals sensitive data. The major change is in the obfuscation technique, wherein multiple embedded JAR archives are used before unpacking the actual payload. Netskope Threat Protection detects the malware as ByteCode-JAVA.Trojan.Kryptik and Gen:Variant.Application.Agentus.1. This blog post provides an analysis of the new campaign and the new obfuscation techniques.

Responsible Disclosure

The URLs hosting the Adwind RAT were reported to Westnet on September 9th, 2019.

Analysis Details

We discovered the new campaign serving the Adwind RAT JAR payload from "members[.]westnet[.]com[.]au/~joeven/". Westnet is an Australian ISP. The attacker is either a Westnet user or has compromised the account of one or more Westnet users. The same RAT is being hosted by multiple other Westnet users. Some of the recent uploads have multiple file extensions (*.png.jar.jar) to hide the actual file-type visibility from the target user. We have listed some of the current upload directories in the Indicators of compromise section. At the time of writing, the links were still active.

When the victim executes the payload, there are multiple levels of JAR extractions that occur. Figure 1 below summarizes the execution stages at a high level.

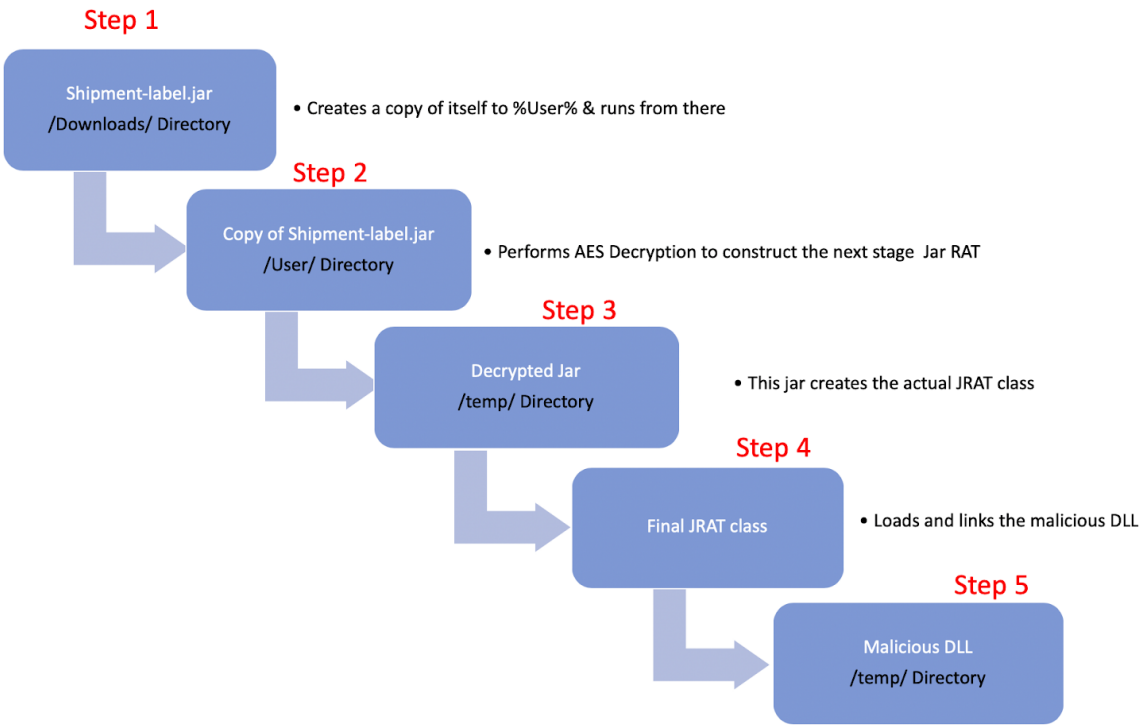

Figure 1: Process execution stages involved in Adwind's infection chain

Step 1

The dropped JAR payload executes and creates the parent java process and copies itself into the %User% directory. Once the copy is created, the java thread performs the following three actions:

- Executes the copy
- Creates a registry entry in *HKCU/CurrentVersion/Run* to maintain persistence.
- Creates WMI scripts in %temp% and launches them. These scripts, shown in Figure 2, disable firewall and antivirus services.

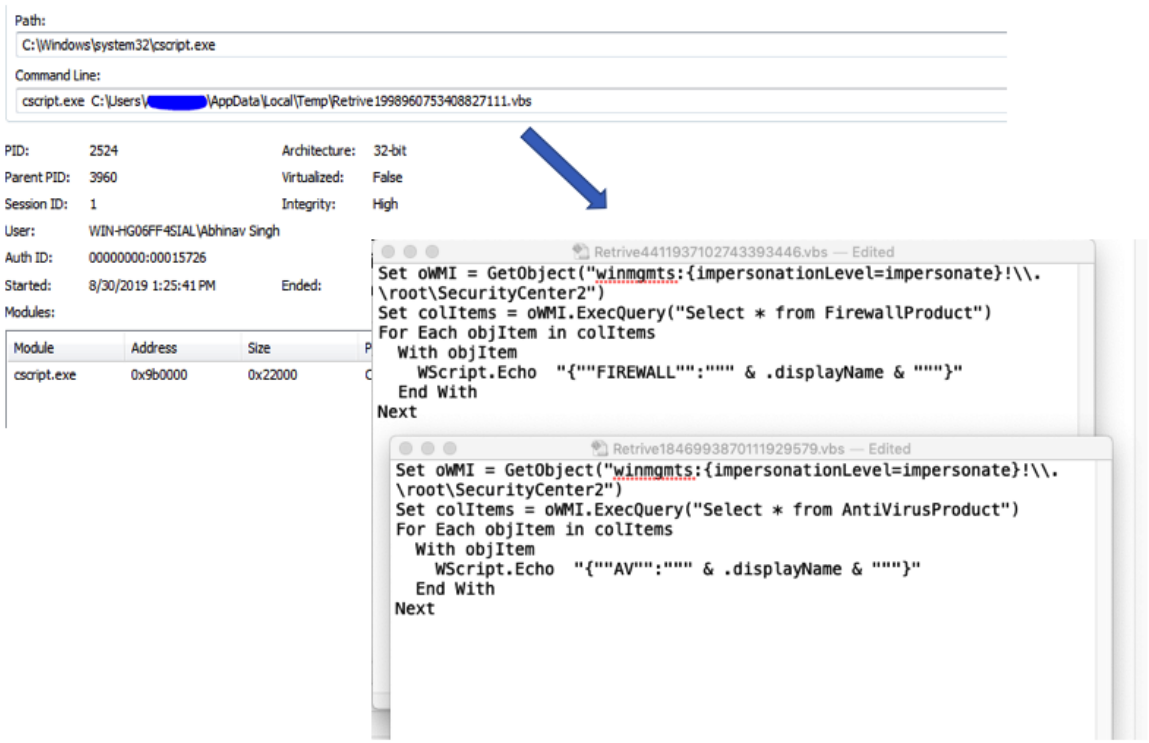

Figure 2: WMI scripts created by the first stage JAR payload.

Step 2

The new JAR dropped in Step 1:

- Performs AES decryption routine on an embedded object to construct the Step 3 JAR
- Writes the Step 3 JAR in the %temp% directory and executes it as a new java thread.

Figure 3 below shows the decompiled class files implementing the decryption routine on an object named “_”.

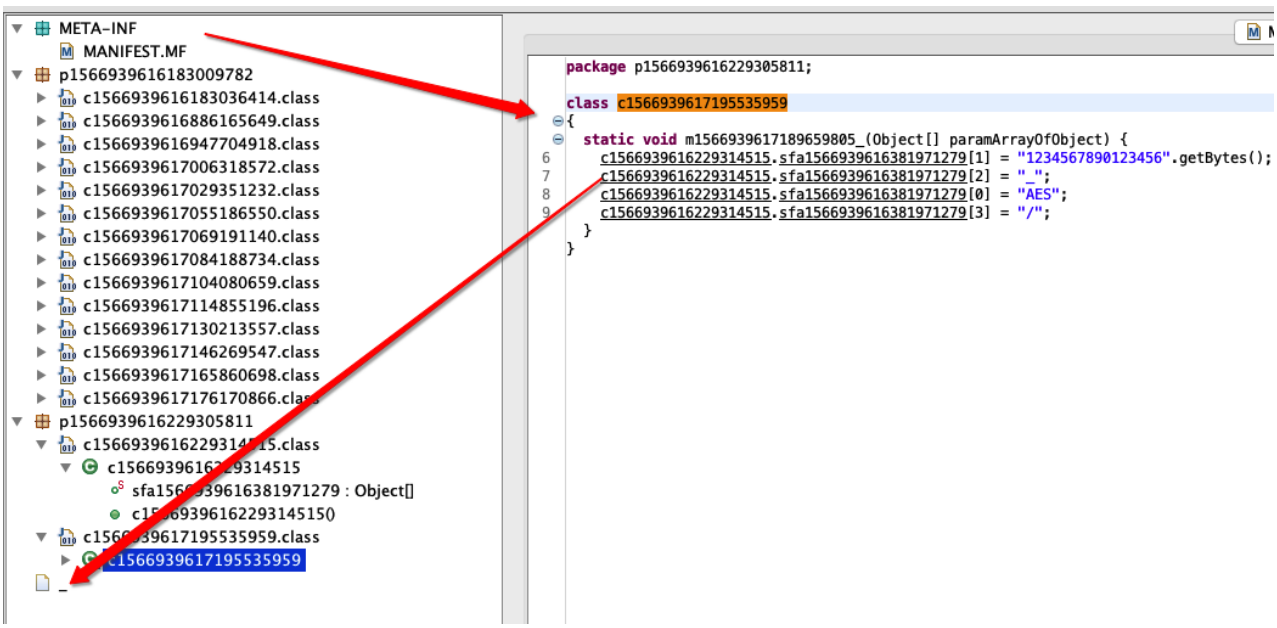

Figure 3: Embedded object which decrypts to JAR file with the JRAT class

Step 3

The Step 3 JAR loads the JRAT class.

Step 4

This JRAT class is responsible for loading and linking the DLL which contains the major RAT functionality. It then tries connecting to its command and control server at 185[.]205[.]210[.]48. The JRAT class contains multiple levels of obfuscations within itself in order to hide its features and functionality.

When we last blogged about it, the RAT was cross-platform and supported Windows, Linux, and Mac. Figure 4 below shows the OS check implemented by JRAT, indicating that the cross-platform support hasn't changed.

```
            }
            if (Server.settings.has("WINDOWS")) {
                RunFile.makeWindowsScript(f);
                return true;
            }
            if (Server.settings.has("LINUX") || Server.settings.has("MAC")) {
                if (RunFile.executeGeneric(f)) {
                    return true;
                }
                Runtime.getRuntime().exec(new String[]{RunFile.getRunnerLinux(), f.getAbsolutePath()});
                return true;
            }
            RunFile.executeGeneric(f);
            return true;
        }
```

Figure 4: JRAT class checking for OS environment

The core functionalities of the RAT is shown in Figure 4 below. Some of the highlight features include:

- Capturing webcam images
- Scanning the hard-drive for files based on extensions defined in RAT's config.
- Spinning up multiple process threads and performing injection into known legitimate windows processes.
- Monitoring system status.
- Encrypting and exfiltrating the data to its command and control server.

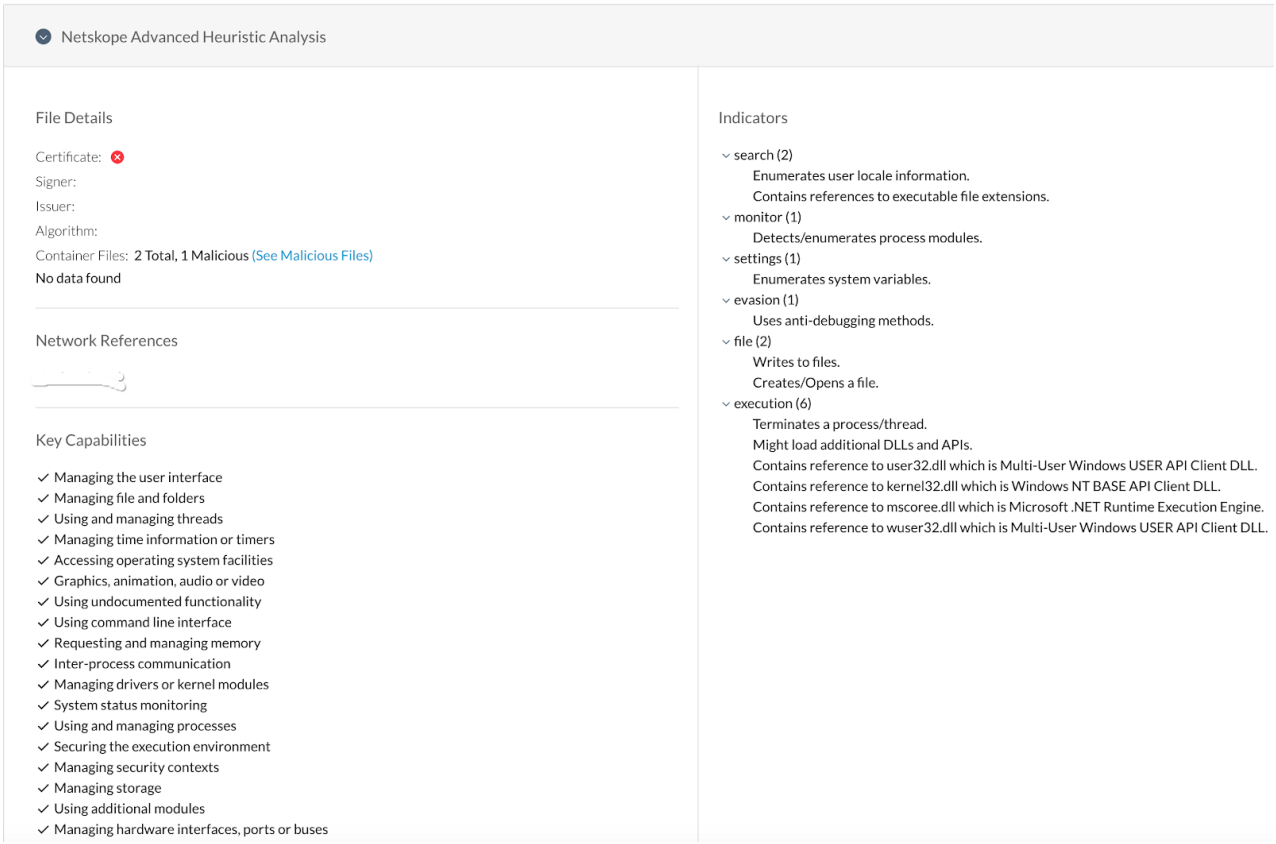

Figure 5: Netskope Advanced Heuristic dashboard listing key features of the RAT.

Conclusion

The Adwind RAT is a well-known malware family that has actively been used in multiple campaigns over the last couple of years. The samples we analyzed showed that the VirusTotal detection ratio for the top-level JAR was 5/56 while that of the final decrypted JAR was 49/58. These detection ratios indicate that attackers have largely been successful in developing new, innovative obfuscation techniques to evade detection.

Indicators Of Compromise

IOC	Type	Description
3bdfd33017806b85949b6faa7d4b98e4	Hash	WMI script created by Malware
a32c109297ed1ca155598cd295c26611	Hash	WMI script created by Malware
a9175094b275a0aaed30604f7dceeb14	Hash	First level JAR payload
781fb531354d6f291f1ccab48da6d39f	Hash	Decrypted JAR file
0b7b52302c8c5df59d960dd97e3abdaf	Hash	DLL file created by the JAR
185.205.210.48	IP	Command and Control IP
huup://members[.]westnet[.]com[.]au/~philchief/	URL	Pages serving the malicious JAR payload
huup://members[.]westnet[.]com[.]au/~lionsnortham/	URL	Pages serving the malicious JAR payload

huup://members[.]westnet[.]com[.]au/~mcleodart/	URL	Pages serving the malicious JAR payload
huup://members[.]westnet[.]com[.]au/~jbush/	URL	Pages serving the malicious JAR payload
huup://members[.]westnet[.]com[.]au/~joeven/	URL	Pages serving the malicious JAR payload
huup://members[.]westnet[.]com[.]au/~howrahnurs-ery_nbn/	URL	Pages serving the malicious JAR payload