# VCURMS: A Simple and Functional Weapon

fortinet.com/blog/threat-research/vcurms-a-simple-and-functional-weapon

March 12, 2024

Article Contents

By Yurren Wan | March 12, 2024
**Affected platforms:** All platforms with Java installed
**Impacted parties:** Any organization
**Impact:** Attackers gain control of the infected systems
**Severity level:** High

Recently, FortiGuard Labs uncovered a phishing campaign that entices users to download a malicious Java downloader with the intention of spreading new VCURMS and STRRAT remote access trojans (RAT). The attackers stored malware on public services like Amazon Web Services (AWS) and GitHub, employing a commercial protector to avoid detection of the malware. The attacker attempts to use email as its command and control throughout the attack campaign. The receiving endpoint utilizes Proton Mail, which offers email services that include privacy protection. Figure 1 shows the attack chain.

Figure 1: Attack flow

This blog describes how the malware is delivered and specifically examines the unusual VCURMS RAT that is involved in this campaign.

## Initial Access

The phishing email shown in Figure 2 is part of this attack campaign. It targets staff members, implying that a payment is underway and encourages them to click a button to verify payment information. Upon clicking the button, a harmful JAR file hosted on AWS is downloaded to the victim's computer.

Figure 2: The phishing e-mail

## Payment-Advice.jar

The downloaded files resemble typical phishing attachments with spoofed names intended to lure people into opening them. When you look at the file with a JAR decompiler, many strings are obfuscated and one of the class names “DownloadAndExecuteJarFiles.class” clearly indicates the intention of this program, as shown in Figure 3. The program aims to download two JAR files to the attacker-provided path and executes them.

Figure 3: Code to download and execute Jar Files

As shown in Figure 4, a class employed by the obfuscator is labeled "sense loader" in the debug data. The obfuscator selects the appropriate native loader module from the resources based on the current operating system during the execution process.

Figure 4: A class employed by the obfuscator

After a specific date, running the malware causes a notification to appear regarding the expiration of the trial for protected tools as shown in Figure 5.

Figure 5: Expiration of the trial for Virbox Protector

Additionally, the code generated by the obfuscator closely resembles the code produced by a legitimate obfuscation tool known as "Sense Shield Virbox Protector" as shown in Figure 6.

Figure 6: Virbox Protector GUI

The rest of the execution flow of the JAR file downloads two additional JAR files and runs them separately.

## Windows.jar

We observed an unusual RAT that communicates with its command and control through email. During the initialization step, the program replicates itself into the Startup folder to ensure that it runs automatically when Windows starts. It then alerts the attacker that the victim is online and establishes a schedule to periodically check the mailbox as shown in Figure 7.

Figure 7: The main function of VCURMS RAT
The attacker identifies the victim using the computer name and Volume ID. When the malware needs to verify the command provided by the attacker, it first examines whether the subject of the email contains identifying information and then proceeds to check the command within the body of the email.

Figure 8: Command of VCURMS RAT
The keylogger and password recovery malware are also hosted on AWS and disguised with a .jpg extension. They are downloaded using a PowerShell command.

Figure 9: Download components using a PowerShell command
In addition to installing keyloggers and password recovery malware, the command provides various customizable features such as the ability to execute shell commands and upload and download files as shown in Table 1.

| Command | Details |
|---|---|
| **get information** | Retrieve system details such as the operating system version, memory capacity, computer name, volume ID, username, country, and the files in the Desktop and Documents folders. |
| **shell** | Obtain the command and execute it through **cmd.exe /c;** the result is sent back to the attacker via email. |
| **recovery** | Download a **recovery** JAR file with a .jpg extension and execute it. |
| **start keylogger** | Download a **keylogger** JAR file with a .jpg extension and execute it. |
| **get keylogger** | Attach the keylogger data and send it as an attachment. |
| **upload** | Compress the file at the specified location and then send it as an attachment. |

| | |
|---|---|
| **download** | Retrieve the attachment and only allow file extensions that are in .jpg format to be accepted. |
| **search** | Look for file names containing keywords specified by the attacker. |

Table 1: Commands

## Malware Protected with Commercial Obfuscator

The most downloaded malware in this campaign are obfuscated using the Branchlock obfuscator. Information about this obfuscator is located at the end of the JAR file as shown in Figure 10.

Figure 10: The obfuscator at the end of the file “stl2.jpg”
The Narumii/Deobfuscator plays a crucial role in partially supporting the deobfuscation of a program obfuscated with Branchlock.

Figure 11: Deobfuscation with Narumii

## Infostealer - Stl2.jpg

When the command "recovery" is received, the program is downloaded and deployed into the %USERPROFILE%\AppData\cookie directory with the name st.jar. The primary purpose of the program is to steal information, particularly system information, popular browsers, and apps.

- Apps: Discord and Steam
- Browsers: Brave, Chrome, Edge, Firefox, Opera, OperaGX, Vivaldi, and Yandex
- System information: Network information, computer information, hardware information, process lists and screenshots.

The program gathers account information from apps and collects cookies, autofill data, browsing history, and passwords from browsers. The data collected is stored in the directory located at %USERPROFILE%/<username>.

Figure 12: The file structure of stl2.jpg and the collected data
Despite the component similarities to Rude Stealer, a Java-based infostealer, this program adopts the name VCURMS. We also have observed a distinction in the method of transmitting the pilfered data. The attacker follows the same path as the main program and sends the stolen information through the same email address.

Figure 13: Code extracted from the SendFile module

## Keylogger - Kl.jpg

The downloaded keylogger will ultimately be stored in %USERPROFILE%\AppData\cookie\klog.jar. This file is responsible for recording keystrokes. Additional actions such as sending logs back to the attacker requires the main JAR file "windows.jar" to execute the functions.

## STRRAT

STRRAT is a RAT built using Java, which has a wide range of capabilities, such as serving as a keylogger and extracting credentials from browsers and applications.

By the end of 2023, it was discovered that STRRAT utilizes two string obfuscation techniques, namely "Zelix KlassMaster (ZKM)" and "Allatori" to avoid detection. However, the STRRAT observed in this attack campaign follows the same convoluted process. It uses the Allatori Java obfuscator and includes the Branchlock obfuscator, which makes analysis more difficult.

Figure 14: The Branchlock obfuscator the end of the file “explorer.jar” STRRAT

Figure 15: A splash screen is displayed when trying to run explorer.jar
The configuration file still remains in the resource. By decoding it using Base64 and decrypting it with the AES Algorithm using the passphrase "strigoi," we can obtain information about the command and control server and ID “Khonsari.”

Figure 16: The decrypted configuration file

## Conclusion

This comprehensive attack operation deploys several malicious programs simultaneously on a victim’s system. It deploys a well-known STRRAT and a new VCURMS based on Java. Even though the VCURMS RAT primarily handles command and control communication, it also includes a modified version of a Rude Stealer and a keylogger in its second phase to gather sensitive data from the victim's system. We discovered that the threat actor was using multiple obfuscation techniques to avoid detection and attempting to use email for communicating with the command and control server.

### Fortinet Protections

The malware described in this report are detected and blocked by FortiGuard Antivirus as:

Java/Agent.A881!tr
Java/Agent.X!tr.spy
Java/Agent.A249!tr

Java/Agent.6057!tr
Java/Agent.E730!tr

FortiGate, FortiMail, FortiClient, and FortiEDR support the FortiGuard Antivirus Service. The FortiGuard antivirus engine is part of each of those solutions. As a result, customers who have these products with up-to-date protections are protected.

The FortiGuard CDR (content disarm and reconstruction) service can disarm the malicious macros within the document.

We also suggest that organizations take the free Fortinet Fortinet Certified Fundamentals (FCF) cybersecurity training. The training is designed to help users learn about today's threat landscape and introduces basic cybersecurity concepts and technology.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block malware attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

If you believe this or any other cybersecurity threat has impacted your organization, please contact the Global FortiGuard Incident Response Team.

## IOCs

### E-mails

copier@ferrellengineering[.]com
sacriliage@proton[.]me

### Domains

bankofindustry[.]s3[.]us-east-2[.]amazonaws[.]com
riseappbucket[.]s3[.]ap-southeast-1[.]amazonaws[.]com
oforntа[.]ddns[.]net
jbfrost[.]live
backinghof[.]ddns[.]net

### Files

97e67ac77d80d26af4897acff2a3f6075e0efe7997a67d8194e799006ed5efc9
8d72ca85103f44742d04ebca02bff65788fe6b9fc6f5a411c707580d42bbd249
588d6f6feefa6273c87a3f8a15e2089ee3a063d19e6a472ffc0249298a72392d
8aa99504d78e88a40d33a5f923caf7f2ca9578031d004b83688aafdf13b3b59f
c0d0dee9b8345da3c6cf3e1c3ce5b5b6e8c9e4002358517df1e3cd04c0f0b3d1