Revamped jRAT Uses New Anti-Parsing Techniques

By About the Author

We have recently observed a newer version of the cross-platform jRAT (Trojan.Maljava) remote access Trojan (RAT) in the wild. This version uses new techniques to evade parsing and detection, as well as to prevent itself from being reverse-engineered, by prepending corrupt MZ files before the malicious JAR file.

We first spotted this version of jRAT in early November 2017. In April 2018, we noticed its number increased more than 300 percent to 1,071 from 333 in March. There could be two reasons why we have not seen huge hits for this version: 1) It wants to remain stealthy and difficult to detect, and used only for targeted attacks; and 2) It may not be widely adopted yet among attackers. While the volumes of these attacks are on the lower side, this jRAT has shown that it is quite capable and can go undetected with minimum presence and anti-parsing methods. The malware mainly targets the financial sector, but we've also seen infections in the service, communications, hospitality, government, and energy sectors.

> The malware mainly targets the financial sector, but we've also seen infections in the service, communications, hospitality, governments, and energy sectors.

Finance-themed spam emails

The infection chain begins with spam emails, which are specially crafted using social engineering techniques to entice victims into opening the attachment. We've seen several themes for emails distributing this version of jRAT, including:

- Proof of payment
- Transfer Details Confirmation
- Transfer Error
- Invoice
- Advance payment Transfer slip and bank account details
- Payment Advice
- Wire instruction
- Credit Advice
- Monthly Report format


Figure 1. Sample of finance-themed spam email

Figure 1. Sample of finance-themed spam email

The emails contain a JAR file attachment. This file comes with a surprise MZ header, as well as two corrupt MZ files prepended before the JAR file.


Figure 2. The JAR file attachment comes with a surprise MZ header and two corrupt MZ files prepended

Figure 2. The JAR file attachment comes with a surprise MZ header and two corrupt MZ files prepended

This thwarts not only MZ parsers, but Java parsers as well. These files do not contain \x00 bytes, which indicates the intent. The MZ files cannot be parsed due to a broken PE structure; the files appear to be full MZ but apparently are used only for evading parsers. This may be considered a defense layer to protect the JAR file from being reverse-engineered. Surprisingly, Java is still able to load and execute this JAR file as weaker zip parsing implementations rely on end of central directory record and parses the content to locate and execute main class.


Figure 3. Corrupt MZ file with 0x00 bytes replaced with 0x20

Figure 3. Corrupt MZ file with 0x00 bytes replaced with 0x20

This file can be recognized as jRAT by looking at the class names.


Figure 4. The wrapper JAR structure

Figure 4. The wrapper JAR structure

The wrapper JAR file drops a secondary JAR file and copies it to a *%Temp%* location. The payload JAR file can be extracted using AES decryption. The first 16 bytes in the file “k” seen in Figure 4 contains the key and the file “e” is the encrypted Java payload.

The JAR runs every time Windows starts, and starts executing and connecting to its command and control (C&C) server at *84.[REMOVED].132.145*. It uses a WMIC interface to identify antivirus products installed on the compromised computer and firewall details.

wmic /node:localhost /namespace:\\root\SecurityCenter2 path AntiVirusProduct get /format:list

The configuration file and key file are visible, but the former is AES-encrypted. The JAR file contains various classes for platform-specific implementations for capturing screenshots, playing audio, downloading and executing files, I/O to and from files, logging keystrokes, among others.


Figure 5. jRAT's configuration file, config.dat, can be decrypted using the AES key in key.dat

Figure 5. jRAT's configuration file, config.dat, can be decrypted using the AES key in key.dat

Capabilities and target platforms

This new version of jRAT has the following capabilities:

- Log keystrokes
- Take screenshots
- Play an audio message
- Access the webcam
- Access the file system to read, write, or delete files
- Download and execute files

With these capabilities, the malware can violate victims' privacy and capture and exfiltrate confidential information from target organizations.

It's also potentially capable of running on the following platforms: FreeBSD, OpenBSD, OSX, Solaris, Linux, Windows, and Android.


Figure 6. C&C addresses and configurations for alternative operating systems

Figure 6. C&C addresses and configurations for alternative operating systems

Protection

Symantec and Norton products detect this threat as the following:

- Trojan.Maljava

Symantec Email Security.cloud technology blocks attacks such as this using advanced heuristics.

Mitigation

Symantec advises users to be careful while opening emails about monetary transactions containing JAR attachments.

- Emphasize multiple, overlapping, and mutually supportive defensive systems to guard against single point failures in any specific technology or protection method. This includes deployment of endpoint, email, and web gateway protection technologies as well as firewalls and vulnerability assessment solutions. Always keep these security solutions up-to-date with the latest protection capabilities.
- Employ two-factor authentication (such as Symantec VIP) to provide an additional layer of security and prevent any stolen or cracked credentials from being used by attackers.
- Educate employees and urge them to exercise caution around emails from unfamiliar sources and around opening attachments that haven't been solicited.
- Require everyone in your organization to have long, complex passwords that are changed frequently. Encourage users to avoid reusing the same passwords on multiple websites, and sharing passwords with others should be forbidden.


Revamped jRAT Uses New Anti-Parsing Techniques


Rohit Sharma

Rohit Sharma

Senior Threat Analysis Engineer
