# Analysis of Nood RAT Used in Attacks Against Linux (Gh0st RAT's Variant) - ASEC

By ATCP

Published: 2024-02-18 · Archived: 2026-04-02 12:31:18 UTC

AhnLab SEcurity intelligence Center (ASEC) recently discovered that Nood RAT is being used in malware attacks. Nood RAT is a variant of Gh0st RAT that works in Linux. Although the number of Gh0st RAT for Linux is fewer compared to Gh0st RAT for Windows, the cases of Gh0st RAT for Linux are continuously being collected. Nood RAT is categorized as a variant of Gh0st RAT based on the code's similarity with previous codes from Gh0st RAT **[1]**. A builder used in the latest developments was found, and it was dubbed Nood RAT, because the author named it Nood.

Nood RAT has been used in various vulnerability attacks since 2018. Although no specific cases of vulnerability attacks have been found recently, cases are continuously being discovered according to the VirusTotal website. This article highlights malware strains discovered over the last few years and analyzes them along with the builder.

## 1. Overview

Gh0st RAT is a remote control malware developed by the C. Rufus Security Team of China **[2]** (This link is only available in Korean.) Because its source code is open to the public, malware authors have been developing various variants using this code, and the threat actors have been utilizing the codes in their attacks to this day. Although the source code is open to the public, the code is mainly used by threat actors who speak Chinese.

In the past, ASEC posted an article about the case where Gh0st RAT's variant Gh0stCringe RAT was distributed to database servers (MS-SQL and MySQL server) **[3]** and later posted the case where HiddenGh0st—the variant of Gh0st RAT that simultaneously installs a Hidden rootkit—was used in attacks on MS-SQL servers. **[4]**

Although there may be various Linux versions of the malware strains as the source code is open to the public, the Nood RAT variant discussed in this article was first found around 2018. The oldest record of the malware is the case where it was installed via a WebLogic vulnerability (CVE-2017-10271) attack **[5]**, and the case where it was used by the threat actor Rocke to install CoinMiners in their attacks. **[6]** The malware was also used in the Cloud Snooper APT attack campaign in 2020, where the threat actor installed a backdoor malware in AWS (Amazon.com's cloud service) servers and hijacked control of the servers. **[7]**

Nood RAT is developed using the following builder. The compressed file contains a release note, a builder program "NoodMaker.exe", and a "Nood.exe" which is used to control the backdoor. During the creation of NoodMaker, the threat actor can create x86 or x64 binary based on the architecture and choose and use the binary that fits the target system.


Figure 1. Nood RAT builder (A Linux version of Gh0st RAT)

Nood RAT has a feature that changes its name in order to disguise itself as a legitimate program. The threat actor is able to decide the malware's fake process name during the development stage. When the malware is launched for the first time it uses the RC4 algorithm to decrypt the encrypted data. The string decrypted here is the name of the process to be changed. Additionally, the configuration data is also encrypted using the RC4 algorithm, and the RC4 key used in the decryption process is the string "r0st@#$". Note that in Socks proxy and port forwarding communication, the string "VMware#@!Station" is used instead.

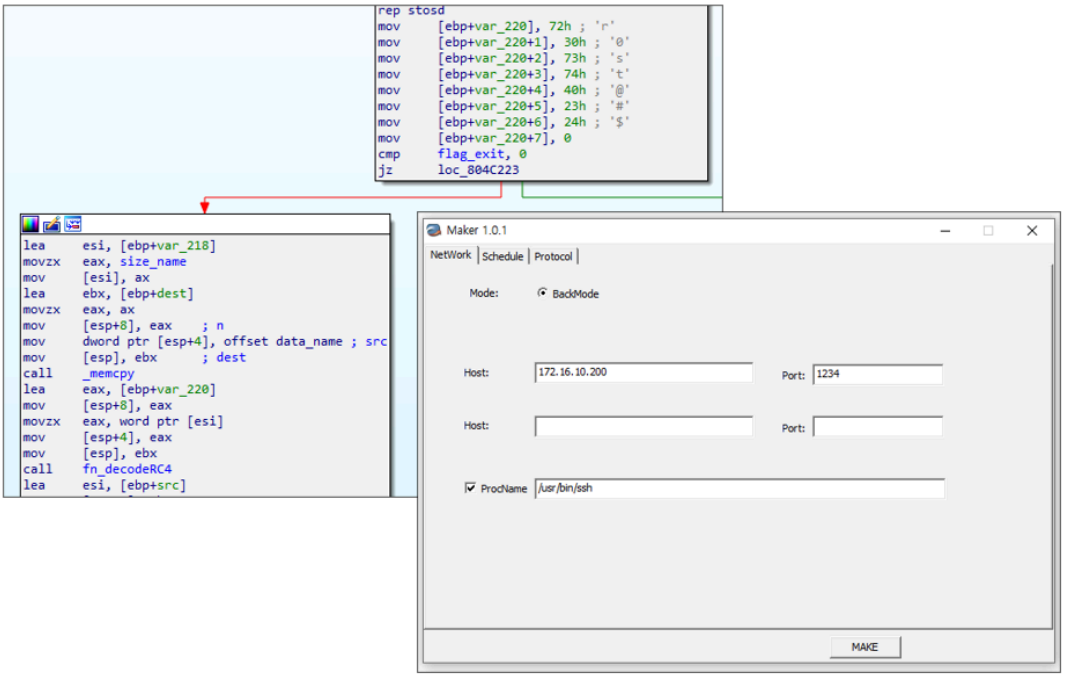


Figure 2. The feature that changes the process name

After changing its process name, said malware copies and pastes itself into the “/tmp/CCCCCCCC” path, runs it, and deletes the copied file “/tmp/CCCCCCCC.” As such, the running malware takes the form of an executed file “/tmp/CCCCCCCC,” but the file does not exist and the malware is shown as a legitimate process with a fake process name.

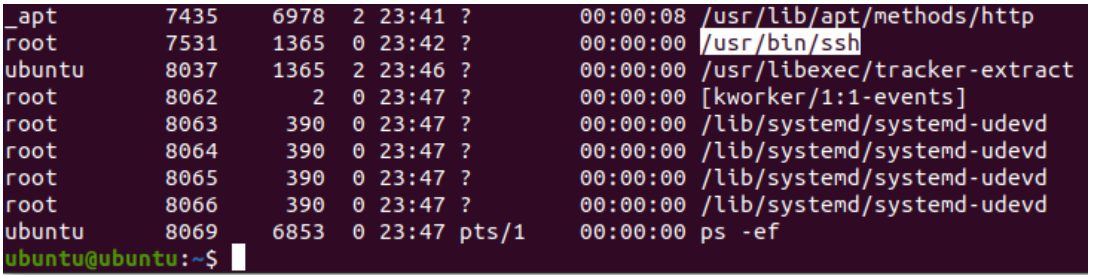


Figure 3. The changed process name

Afterward, the malware decrypts the configuration data which is largely divided into C&C server addresses, date and time of activation, and C&C connection attempt intervals. The threat actor can set the activation date and time at which said malware can communicate with the C&C server and receive commands.

- **Configuration Data Format**:
  “C&C_Server_1″;”C&C_Server_2″|”Mon”;”Tue”;”Wed”;”Thu”;”Fri”;”Sat”;”Sun”;|”Time”;|”Interval”

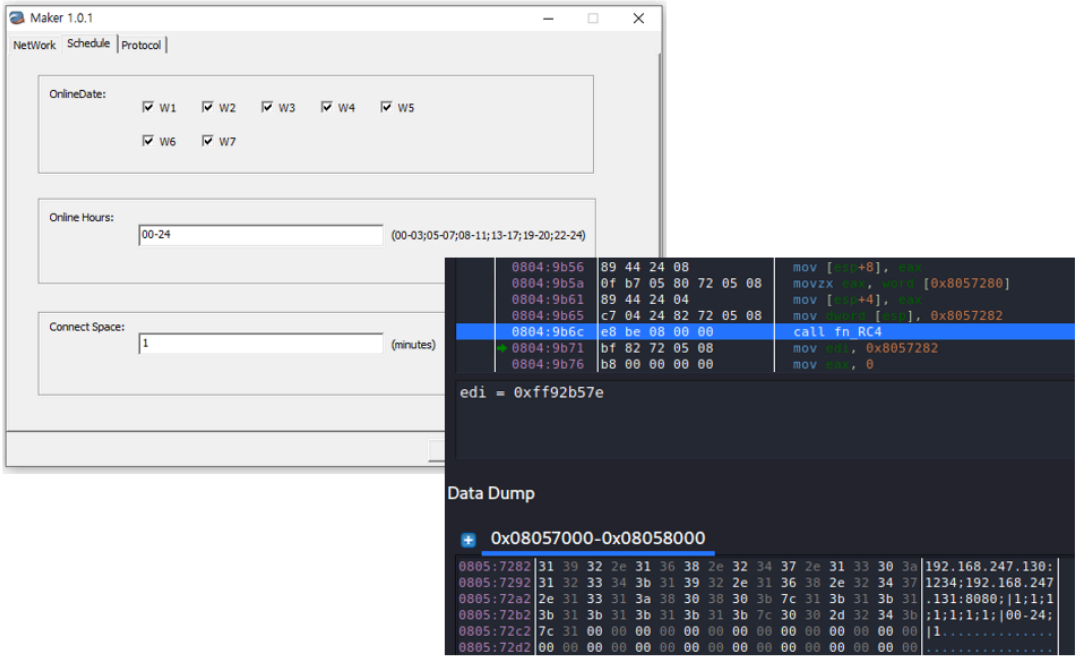


Figure 4. Builder and configuration data

When connecting to the C&C server for the first time, Gh0st RAT obtains basic information about the infected system and sends the data. The sent data is encrypted using the RC4 algorithm, and because the key used in the encryption is created based on the current time, it can bypass network packet-based detection.

| Offset | Size | Data |
|---|---|---|
| 0x0000 | 0x0018 | “Key Type 2” (encrypted with Key Type 1) |
| 0x0018 | 0x0004 | “Key Type 1” |

| Offset | Size | Data |
|---|---|---|
| 0x001C | 0x0208 | Infected system's information (encrypted with Key Type 2) |

Table 1. Data sent to C&C server

The first sent data has a size of 0x18 and consists of two hardcoded 4-byte values and four 4-byte values that are created based on the current time. These values are encrypted using the RC4 algorithm and are sent to the server. The keys used to encrypt these values are created using a key called "Key Type 1."

| Offset | Size | Data |
|---|---|---|
| 0x00 | 0x04 | Created 4-byte key #1 |
| 0x04 | 0x04 | Created 4-byte key #2 |
| 0x08 | 0x04 | Created 4-byte key #3 |
| 0x0C | 0x04 | 0x00009F72 |
| 0x10 | 0x04 | Created 4-byte key #4 |
| 0x14 | 0x04 | 0x000002E9 |

Table 2. Encrypted key data

The C&C server is able to use "Key Type 1" to create an RC4 key to decrypt "Key Type 2," and utilize the RC4 key that was created using "Key Type 2" to decrypt 0x0208-sized data, ultimately obtaining the infected system's information.

| Offset | Type | Data |
|---|---|---|
| 0x0000 | String | Login banner string (the file content of "/etc/issue.net" or "/etc/issue") |
| 0x0100 | Flag | Whether the login banner string data was obtained (0x01 / 0x00) |
| 0x0101 | Flag | Whether the keyword x86_64 exists in the "/proc/version" architecture (0x01 / 0x00) |
| 0x0102 | String | Host name |
| 0x0202 | Flag | Whether the host name was obtained (0x01 / 0x00) |
| 0x0203 | Hex | The hexadecimal value of the IP address |
| 0x0207 | Flag | Whether the IP address was obtained (0x01 / 0x00) |

Table 3. The infected system's information sent to the C&C server

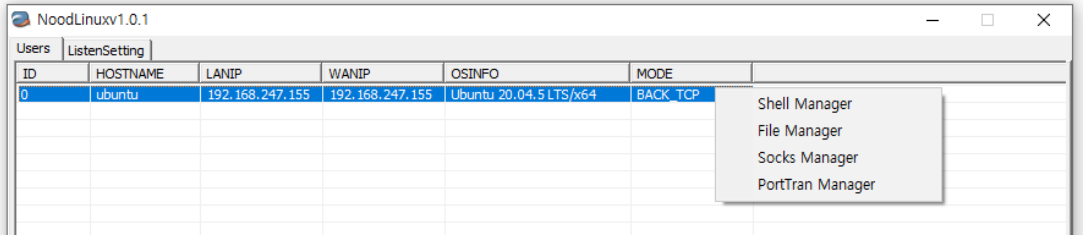


Figure 5. The infected system's information shown on the C&C panel

Nood RAT largely supports four features which are: remote shell & file management, Socks proxy, and port forwarding. Through this, threat actors can run malicious commands on infected systems or steal information using file upload and download features. Additionally, threat actors can use infected systems as proxies or use the systems during the lateral movement phase via the port forwarding feature.

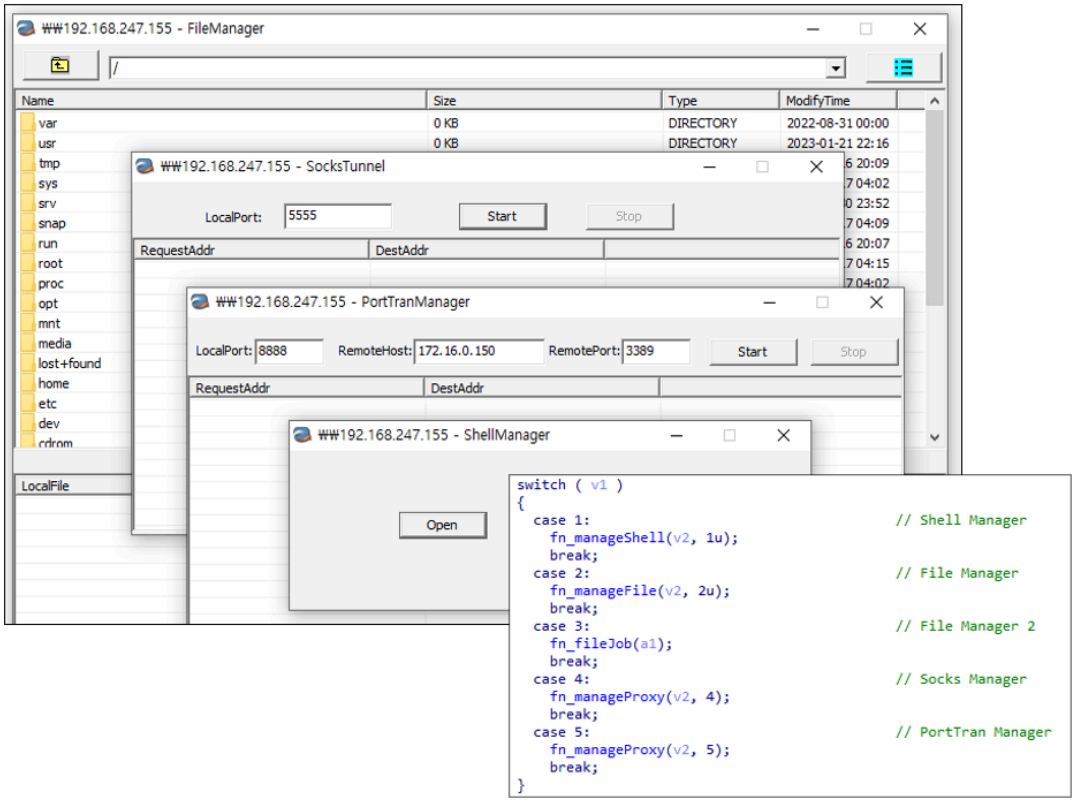


Figure 6. Commands supported by Nood RAT

## 3. Attack Cases

WebLogic vulnerability attacks and Cloud Snooper APT attacks are some of the attacks that used Nood RAT in the past. Nood RAT are still being continuously collected even today, and are also uploaded by the VirusTotal website. Details of attack methods have not yet been uncovered, but it is likely that threat actors are using the malware to control infected systems and steal information from such systems. The following is a table that provides an overview of Nood RATs discovered during the past few years.

| Date of Collection | Country | Name | Disguised Process | Configuration Data |
|---|---|---|---|---|
| 240130 | KR | AliDunYun | /usr/bin/ssh | 43.156.118[.]72:443;43.156.118.72:443;\|1;1;1;1;1;1;1;\|00-24;\|1 |
| 240116 | HK | pki.rar | /usr/bin/ssh | b.niupilao[.]vip:80;\|1;1;1;1;1;1;1;\|00-24;\|1 |
| 231028 | PH | x.uu | [kworker/0:0] | update.kworker[.]net:443;check.snapupdate[.]org:80<br>1;1;1;1;1;1;1;\|00-24;\|1 |
| 231027 | CN | nginx | /usr/bin/ssh | 42.51.40[.]184:56;\|1;1;1;1;1;1;1;\|00-24;\|1 |
| 230907 | RU | MFWzS4YNXpQd | [kworker/2:0] | 13.214.222[.]35:443;\|1;1;1;1;1;1;1;\|00-24;\|1 |

| Date of Collection | Country | Name | Disguised Process | Configuration Data |
|---|---|---|---|---|
| 221013 | HK | hsperf | kworker | cloud.awsxtd[.]com:443;\|1;1;1;1;1;1;1;1;\|00-24;\|3 |
| 220911 | RU | adyagent | /usr/bin/ssh | 43.140.251[.]218:8080;\|1;1;1;1;1;1;1;1;\|00-24;\|1 |
| 220726 | CN | update | /usr/bin/ssh | 101.42.139[.]110:8443;101.42.139[.]110:53;\|1;1;1;1;1;1;1;\|00-24;\|1 |
| 220113 | CN | update | /usr/bin/ssh | 81.68.143[.]132:1234;81.68.143[.]132:8080;\|1;1;1;1;1;1;1;\|00-24;\|1 |
| 211213 | VN | bo | /usr/bin/ssh | bo.appleupcheck[.]com:443;\|1;1;1;1;1;1;1;\|00-24;\|1 |
| 210921 | PK | N/A | /usr/sbin/xfs_srv | 194.36.191[.]75:443;\|1;1;1;1;1;1;1;\|00-24;\|1 |
| 210601 | CN | titan.bin | /usr/bin/ssh | 1.117.165[.]141:53;1.117.165[.]141:53;\|1;1;1;1;1;1;1;\|00-24;\|1 |
| 210403 | CN | N/A | /sbin/auditd | 23.100.88[.]61:53;\|1;1;1;1;1;1;1;\|00-24;\|10 |

Table 4. Nood RAT malware

## 4. Conclusion

Various threat actors have been actively using Gh0st RAT to infect not only Windows systems but also its Linux counterpart—developed based on the publicized source code. Among the variants of Gh0st RAT, a Linux variant called Nood continues to be found and collected across nations.

Nood RAT is a backdoor malware that can receive commands from the C&C server to perform malicious activities such as downloading malicious files, stealing systems' internal files, and executing commands. Although simple in form, it is equipped with the encryption feature to avoid network packet detection and can receive commands from threat actors to carry out multiple malicious activities.

To prevent such security threats, users must check their vulnerable environment configuration or credentials and always update relevant systems to the latest versions. Also, V3 should be updated to the latest version so that malware infection can be prevented.

**File Detection**
– Linux/Agent.86208 (2029.01.08.00)
– Backdoor/Linux.Rekoobe.86144 (2022.06.15.00)
– Backdoor/Linux.Rekoobe.86176 (2022.06.15.00)
– Backdoor/Linux.Rekoobe.83264 (2022.06.15.00)

MD5

035f83018cf96f5e1f6817ccd39fc0b6

0a35e06f53c17ab1c8e18e7e0c0821d8

35743db3dc333245ef5b69100721ced9

4f3afdcfff8f7994b7d3d3fbaa6858b4

75838e5d481da40db2e235a6d5a222ef

Additional IOCs are available on AhnLab TIP.

URL

http[:]//1[.]117[.]165[.]141[:]53/

http[:]//101[.]42[.]139[.]110[:]53/

http[:]//101[.]42[.]139[.]110[:]8443/

http[:]//23[.]100[.]88[.]61[:]53/

http[:]//42[.]51[.]40[.]184[:]56/

Additional IOCs are available on AhnLab TIP.

Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For subscription details, click the banner below.



---

Source: https://asec.ahnlab.com/en/62144/