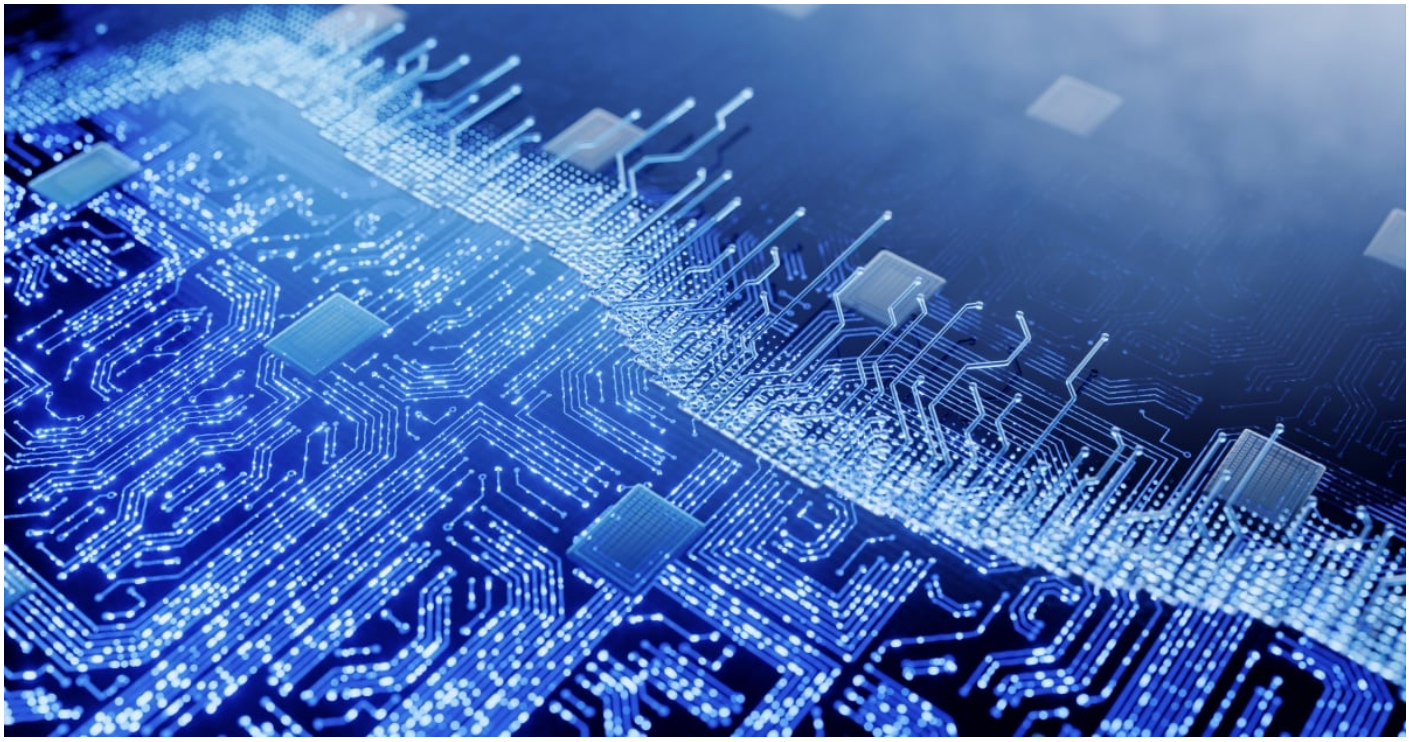
Springtail: New Linux Backdoor Added to Toolkit



Symantec's Threat Hunter Team has uncovered a new Linux backdoor developed by the North Korean Springtail espionage group (aka Kimsuky) that is linked to malware used in a recent campaign against organizations in South Korea.

The backdoor (Linux.Gomir) appears to be a Linux version of the GoBear backdoor, which was used in a recent Springtail campaign that saw the attackers deliver malware via Trojanized software installation packages. Gomir is structurally almost identical to GoBear, with extensive sharing of code between malware variants.

Background

Springtail is a tight-knit espionage group that initially specialized in attacks on public sector organizations in South Korea. The group first came to public attention in 2014, when the South Korean government said it was responsible for an attack on Korea Hydro and Nuclear Power (KHNP). Multiple employees at KHNP were targeted with spear-phishing emails containing exploits that installed disk-wiping malware on their machines. The U.S. government has said that the group is a unit of North Korea's military intelligence organization, the Reconnaissance General Bureau (RGB).

The group was the subject of a U.S. government alert in recent days due to attempts to exploit improperly configured DNS Domain-based Message Authentication, Reporting and Conformance (DMARC) record policies to conceal social engineering attempts. According to a joint advisory issued by the Federal Bureau of Investigation (FBI), the U.S. Department of State, and the National Security Agency (NSA), the

group has been mounting spear phishing campaigns pretending to be journalists, academics, and experts in East Asian affairs “with credible links to North Korean policy circles”.

Trojanized software packages

The campaign, which was first documented by South Korean security firm S2W in February 2024, saw Springtail deliver a new malware family named Troll Stealer using Trojanized software installation packages. Troll Stealer can steal a range of information from infected computers including files, screenshots, browser data, and system information. Written in Go, like many newer Springtail malware families, Troll Stealer contained a large amount of code overlap with earlier Springtail malware.

Troll Stealer’s functionality included the ability to copy the GPKI (Government Public Key Infrastructure) folder on infected computers. GPKI is the public key infrastructure schema for South Korean government personnel and state organizations, suggesting that government agencies were among the targets of the campaign.

S2W reported that the malware was distributed inside installation packages for TrustPKI and NX_PRNMAN, software developed by SGA Solutions. The installation packages were reportedly downloaded from a page that was redirected from a specific website.

South Korean security firm AhnLab subsequently provided further details on the downloads, saying they originated from the website of an association in the construction sector. The website required users to log in and the affected packages were among those users had to install to do so.

Symantec has since discovered that Troll Stealer was also delivered in Trojanized Installation packages for Wizvera VeraPort. It is unclear how these installation packages were delivered during the current campaign. Wizvera VeraPort was previously reported to have been compromised in a North Korea-linked software supply chain attack in 2020.

Troll Stealer and GoBear

Troll Stealer appears to be related to another recently discovered Go-based backdoor named GoBear. Both threats are signed with a legitimate certificate issued to “D2innovation Co.,LTD”. GoBear also contains similar function names to an older Springtail backdoor known as BetaSeed, which was written in C++, suggesting that both threats have a common origin.

AhnLab later explicitly linked the two threats, saying that many of the malicious installers it had analyzed contained both Troll Stealer and either of the GoBear or BetaSeed backdoors, which it referred to as the Endoor malware family.

Several weeks later, GoBear was being distributed by a dropper masquerading as an installer for an app for a Korean transport organization. In this case, the attackers did not Trojanize a legitimate software package but instead disguised the dropper as an installer featuring the organization’s logos. The dropper was signed with what appeared to be a stolen certificate.

Gomir backdoor

Symantec's investigation into the attacks uncovered a Linux version of this malware family (Linux.Gomir) which is structurally almost identical and shares an extensive amount of distinct code with the Windows Go-based backdoor GoBear. Any functionality from GoBear that is operating system-dependent is either missing or reimplemented in Gomir.

When executed, it checks its command line and if contains the string "install" as its only argument, it will attempt to install itself with persistence.

To determine how it installs itself, Gomir checks the effective group ID (as reported by the getegid32() syscall) of its own process. If the process is running as group 0, Gomir assumes that it is running with superuser privileges and attempts to copy itself as the following file:

```
/var/log/syslogd
```

It then attempts to create a systemd service with the name "syslogd" by creating the file:

```
/etc/systemd/system/syslogd.service
```

The file contains:

```
[Unit]

After=network.target

Description=syslogd

[Service]

ExecStart=/bin/sh -c "/var/log/syslogd"

Restart=always

[Install]

WantedBy=multi-user.target
```

Gomir will then enable and start the created service by executing the following sequence of commands:

```
${SHELL} -c systemctl daemon-reload

${SHELL} -c systemctl reenable syslogd

${SHELL} -c systemctl start syslogd
```

It will then delete the original executable and terminate the original process.

If the process is running as any group other than 0, Gomir attempts to configure a crontab to start the backdoor on every reboot. It first creates a helper file (cron.txt) in the current working directory with the following content:

```
@reboot [PATHNAME_OF_THE_EXECUTING_PROCESS]
```

Next, it seems to attempt to list any pre-existing crontab entries by running the following command:

```
/bin/sh -c crontab -l
```

It appends the output to the created helper file.

Gomir then updates the crontab configuration by executing the following command:

```
${SHELL} -c crontab cron.txt
```

Gomir then deletes the helper file before executing itself without any command-line parameters.

Once installed and running, Gomir periodically communicates with its command-and-control (C&C) server by sending HTTP POST requests to: http://216.189.159[.]34/mir/index.php

When pooling for commands to execute, Gomir requests with the following HTTP request body:

```
a[9_RANDOM_ALPHANUMERIC_CHARACTERS]=2&b[9_RANDOM_ALPHANUMERIC_CHARACTERS]=
[INFECTION_ID]1&c[9_RANDOM_ALPHANUMERIC_CHARACTERS]=
```

The INFECTION_ID is generated using the following method:

```
def generate_infection_id(hostname, username):

   hexdigest = hashlib.md5(hostname + username).hexdigest()

    return "g-" + hexdigest[:10]
```

The expected body of the HTTP server response is a string starting with the letter S. Gomir then attempts to decode the remaining characters of the string using the Base64 algorithm. The decoded blob has the following structure:

Offset	Size	Description
0	4 Bytes	Encryption key
4	Remainder of the blob	Encrypted command

Gomir then uses a custom encryption algorithm to decrypt the previously discussed command.

The first two characters of the command identify the operation to execute. Gomir allows the execution of 17 different commands. The commands are almost identical to those supported by the GoBear Windows backdoor:

Table 2. Gomir command operations

Operation	Description
01	Pauses communication with the C&C server for an arbitrary time duration.
02	Executes an arbitrary string as a shell command ("[shell]" "-c" "[arbitrary_string]"). The shell used is specified by the environment variable "SHELL", if present. Otherwise, a fallback shell is configured by operation 10 below.
03	Reports the current working directory.
04	Changes the current working directory and reports the working directory’s new pathname.

Operation	Description
05	Probes arbitrary network endpoints for TCP connectivity.
06	Terminates its own process. This stops the backdoor.
07	Reports the executable pathname of its own process (the backdoor executable).
08	Collects statistics about an arbitrary directory tree and reports: total number of subdirectories, total number of files, total size of files
09	Reports the configuration details of the affected computer: hostname, username, CPU, RAM, network interfaces, listing each interface name, MAC, IP, and IPv6 address
10	Configures a fallback shell to use when executing the shell command in operation 02. Initial configuration value is "/bin/sh".
11	Configures a codepage to use when interpreting output from the shell command in operation 02.
12	Pauses communication with the C&C server until an arbitrary datetime.
13	Responds with the message "Not implemented on Linux!" (hardcoded).
14	Starts a reverse proxy by connecting to an arbitrary control endpoint. The communication with the control endpoint is encrypted using the SSL protocol and uses messages consistent with https://github.com/kost/revsocks.git, where the backdoor acts as a proxy client. This allows the remote attacker to initiate connections to arbitrary endpoints on the victim network.
15	Reports the control endpoints of the reverse proxy.
30	Creates an arbitrary file on the affected computer.
31	Exfiltrates an arbitrary file from the affected computer.

Heavy focus on supply chain attacks

This latest Springtail campaign provides further evidence that software installation packages and updates are now among the most favored infection vectors for North Korean espionage actors. Variations of this tactic include:

- Software supply chain attacks
- Trojanized software installers
- Fake software installers

The most notable example to date is the 3CX supply chain attack, which itself was the result of the earlier X_Trader supply chain attack. Springtail, meanwhile, has focused on Trojanized software installers hosted on third-party sites requiring their installation or masquerading as official apps. The software targeted appears to have been carefully chosen to maximize the chances of infecting its intended South Korean-based targets.

Protection/Mitigation

For the latest protection updates, please visit the Symantec Protection Bulletin.

Indicators of Compromise

If an IOC is malicious and the file is available to us, Symantec Endpoint products will detect and block that file.

30584f13c0a9d0c86562c803de350432d5a0607a06b24481ad4d92cdf7288213 – Linux.Gomir

7bd723b5e4f7b3c645ac04e763dfc913060eaf6e136eecc4ee0653ad2056f3a0 – GoBear Dropper

d7f3ecd8939ae8b170b641448ff12ade2163baad05ca6595547f8794b5ad013b – Troll Stealer

36ea1b317b46c55ed01dd860131a7f6a216de71958520d7d558711e13693c9dc – Troll Stealer

8e45daace21f135b54c515dbd5cf6e0bd28ae2515b9d724ad2d01a4bf10f93bd – Troll Stealer

6c2a8e2bbe4ebf1fb6967a34211281959484032af1d620cbab390e89f739c339 – Troll Stealer

47d084e54d15d5d313f09f5b5fcdea0c9273dcddd9a564e154e222343f697822 – Troll Stealer

8a80b6bd452547650b3e61b2cc301d525de139a740aac9b0da2150ffac986be4 - Troll Stealer

380ec7396cc67cf1134f8e8cda906b67c70aa5c818273b1db758f0757b955d81 – Troll Stealer

ff945b3565f63cef7bb214a93c623688759ee2805a8c574f00237660b1c4d3fd – Troll Stealer

cc7a123d08a3558370a32427c8a5d15a4be98fb1b754349d1e0e48f0f4cb6bfc – Troll Stealer

8898b6b3e2b7551edcceffbef2557b99bdf4d99533411cc90390eeb278d11ac8 – Troll Stealer

ecab00f86a6c3adb5f4d5b16da56e16f8e742adfb82235c505d3976c06c74e20 – Troll Stealer

d05c50067bd88dae4389e96d7e88b589027f75427104fdb46f8608bbcf89edb4 – Troll Stealer

a98c017d1b9a18195411d22b44dbe65d5f4a9e181c81ea2168794950dc4cbd3c – Troll Stealer

831f27eb18caf672d43a5a80590df130b0d3d9e7d08e333b0f710b95f2cde0e0 – Troll Stealer

bc4c1c869a03045e0b594a258ec3801369b0dcabac193e90f0a684900e9a582d – Troll Stealer

5068ead78c226893df638a188fbe7222b99618b7889759e0725d85497f533e98 – Troll Stealer

216.189.159[.]34