2019/06/04 Advisory: Windigo attacks

security.web.cern.ch/security/advisories/windigo/windigo.shtml

- 🇫🇷
- Home
- Computing Rules
- Recommendations
- Training
- Services
- Reports & Presentations

This page covers ongoing attacks and may be updated (latest: 2019-07-17).

Introduction

Windigo has been a long-standing adversary of the Research & Education community. The first known attack dates back to 2011, when the Ebury malware was discovered during the Linux Foundation attack in 2011. A technical paper describing the attacks in-depth was published in 2014 by ESET: "Operation Windigo". After the arrest and conviction of one of the key attackers, our community observed a sharp drop in the number of infections. Sporadic Ebury infections continued to be observed in other sectors and led to another update from ESET: Windigo Still not Windigone: An Ebury Update. A lot could be learnt from the attacks to better protect our community, for example to detect OpenSSH backdoors.

Unfortunately, dozens of Linux servers were discovered compromised again in the Research & Education community in May 2019, all running new and improved versions of Ebury.

Ebury malware

The Ebury malware samples detected in May 2019 appear to be evolutions of the malicious code from 2014 and 2017. The technical documents referenced in the introduction remain largely valid. In a nutshell, the Ebury malware replaces legitimate dynamic libraries (e.g. libkeyutils.so) in the affected system to capture SSH passwords and SSH keys. When executed:

- The infected dynamic library will load a secondary malicious dynamic library (e.g. libstz.so), containing the actual malicious code;
- Malicious hooks and a specific OpenSSH server configuration are pushed within the running OpenSSH server binary to accomodate the capture of credentials and backdoor functionality;
- A malicious process is forked and dedicated to the extraction of credentials;

- The two malicious processes communicate via an abstract unix socket.

Ebury backdoor

Ebury uses a complex backdoor mechanism described in the technical documents referenced in the introduction. In the samples collected in May 2019, the authentication part of the backdoor was updated and reinforced. Nevertheless, the attacker seem to continue to connect daily to infected hosts and exfiltrate the credentials using known techniques.

Indicators of compromises

Network: Backdoor connections

The attacker is known to use direct incoming SSH connections to the victims OpenSSH server. The backdoor connection will be regular SSH connections, with two notable exceptions:

- The SSH client string will be different: SSH-2.0-XXXXXXXXXXXXXXX, where XXXXXXXXXXXXXXX will either be a Hex number (Ebury < 1.7.0) or Base64 string (Ebury >= 1.7.0);
- The string is an encrypted message for the backdoor. It can be decoded based on the incoming IP address used by the attacker;
- Logging all incoming SSH client versions using network monitoring tools is very helpful.

Example malicious strings from incoming IP address 80.82.67.21: SSH-2.0-1f25412f1c4d340d173f003a35150d5734111a5562471c.

Backdoor connections (incoming SSH) may be observed form the following IP addresses:

- 94.140.120.163
- 49.50.70.223
- 80.82.67.21
- 125.160.17.32

It is essential to monitor the SSH client version string.

Network: Password exfiltration

The SSH client string is decrypted by the client to reveal the credentials exfiltration IP address that needs to be used by the malware. **Both the SSH client string and the incoming IP address are needed to decrypt the malicious command and exfiltration IP address**.

The exfiltration IP address is used by the malware to exfiltrate captured credentials as a outgoing DNS request (UDP:53). The following IP address has been identified as the exfiltration (outgoing UDP:53) host: 91.236.116.62

Additional exfiltration methods are described in the technical documents referenced in the introduction. In particular, the malware also relies on a DGA, and the attacker seems to have prepared the following domains:

- larfj7g1vaz3y.net / 78.140.134.7.
- op3f1libgh.biz / 193.0.179.76.

These domains are actively maintained by the attacker. However, no known affected host has been observed using this exfiltration path in the current attack.

Host-based indicators

The easiest way to inspect the OpenSSH server for signs of infection is to verify the dynamic libraries in use. In particular, libkeyutils.so.* will typically have a single link to libc.so. Note that the malware uses different library filenames (e.g. libsbz.so, libstz.so, etc.). For example, on an infected system:

```
# objdump -x /lib64/libkeyutils.so.1|grep NEEDED
NEEDED libc.so.6
NEEDED libstz.so
```

This indicates that libkeyutils.so.1 has been infected and will call Ebury from libstz.so. This contrasts with a clean system:

```
# objdump -x /lib64/libkeyutils.so.1 |grep NEEDED
NEEDED libc.so.6
```

Due to code errors in the malware, it is also common to discover segfault messages in the kernel logs, for example:

```
Jun 04 09:01:03 hostname kernel: sshd[12345]: segfault at 7fcf794b8000 ip
00007fcf792a50b8 sp 00007fffb6b8e0e0 error =
```

The malicious abstract unix socket may also be identified by running the following command:

```
# lsof |grep "@/run"
```

Note: as it is an abstract socket, it is not possible to find it on the filesystem.

Responding to the attack

The attack is relying on stolen SSH credentials in our highly interconnected community. Therefore the initial infection vector is often a valid SSH login from a valid user from a known location.

As a result, collaboration between the affected organizations is absolutely crucial. Responding to this attack extensively relies on victims reporting attacks and contributing new malicious samples, and sharing back with the community. If you are affected by this attack, you can:

- Ask directly for advice or support to cert@cern.ch;
- Share **with or without attribution** malicious samples and indicators of compromise with cert@cern.ch.