The Dark Side of the ForSSHe

A landscape of OpenSSH backdoors

Gaspare Ferraro

ICT Risk Assessment

University of Pisa
Master Degree in Computer Science

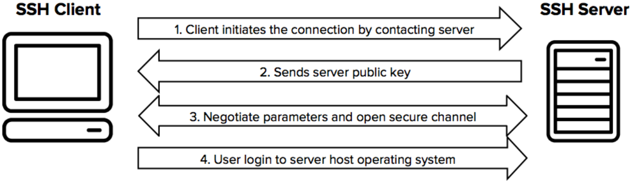
Pisa, July 21, 2019

Part I

Introduction

SSH

Secure **Sh**ell, protocol for secure remote login and other secure network services **over an insecure network**.



Simplified setup flow (source: `ssh.com`)

Developed in 1995 in response to a hacking incident, **today standard protocol** for secure operations.

OpenSSH suite

Suite of secure networking utilities based on SSH protocol.

Coming by default in a large number of operating systems

Utilities:

- SCP, secure copy of files between two different hosts
- SFTP, secure file transfer program
- SSH, secure shell client
- SSHD, ssh server daemon
- keys utilities (SSH-ADD, SSH-AGENT, SSH-KEYGEN, SSH-KEYSCAN)

Operation Windigo

Large and sophisticated operation started in 2011 and discovered after 3 years.

The operation has compromised linux servers in order to **steal SSH credentials**, redirect web traffic and send spam message.

Three different components of the operations:

- **Ebury, OpenSSH backdoor** used to gain full access, steal credentials and keep control of the servers.
- Cdorked, an HTTP backdoor used to redirect traffic and a modified DNS server to resolve arbitrary IP addresses.
- Calfbot, a Perl script used to send spam.

Results:

- highly portable malicious modules were developed in order to cover as many systems as possible.
- 25,000 unique servers compromised.
- 500,000 visitors per day redirected to malicious websites.
- 35,000,000 spam email sent.

Post-operation analysis

Post-operation analysis lead ESET to **extend coverage about OpenSSH backdoors**.

After months of research and data collection, ESET grouped a series of samples in 21 different OpenSSH malware families, **12 of them undocumented** at the time of the paper.

ESET - IT security company

Malware were divided according to common features and cataloged by complexity and time of activity.

Part II

Common features of OpenSSH backdoors

Strings and code obfuscation

Attackers need a way to obfuscate strings and code of backdoor (such as filenames or directories).

XOR cipher: simplest method, encrypt the strings by xor the string with a key.

String stacking: construct strings directly in the stack in order to bypass simple string searched.

```
0040A33B BE 20 D1 43 00          mov     esi, offset aA  ; "a+"
0040A340 BF 70 A7 65 00          mov     edi, offset log_filename ; filename
0040A345 C6 05 24 04 25 00 2F    mov     cs:log_filename, 2Fh ; '/'
0040A34C C6 05 1E 04 25 00 75    mov     cs:log_filename+1, 75h ; 'u'
0040A353 C6 05 18 04 25 00 73    mov     cs:log_filename+2, 73h ; 's'
0040A35A C6 05 12 04 25 00 72    mov     cs:log_filename+3, 72h ; 'r'
0040A361 C6 05 0C 04 25 00 2F    mov     cs:log_filename+4, 2Fh ; '/'
0040A368 C6 05 06 04 25 00 73    mov     cs:log_filename+5, 73h ; 's'
0040A36F C6 05 00 04 25 00 68    mov     cs:log_filename+6, 68h ; 'h'
0040A376 C6 05 FA 03 25 00 61    mov     cs:log_filename+7, 61h ; 'a'
0040A37D C6 05 F4 03 25 00 72    mov     cs:log_filename+8, 72h ; 'r'
0040A384 C6 05 EE 03 25 00 65    mov     cs:log_filename+9, 65h ; 'e'
0040A38B C6 05 E8 03 25 00 2F    mov     cs:log_filename+0Ah, 2Fh ; '/'
0040A392 C6 05 E2 03 25 00 58    mov     cs:log_filename+0Bh, 58h ; 'X'
0040A399 C6 05 DC 03 25 00 31    mov     cs:log_filename+0Ch, 31h ; '1'
0040A3A0 C6 05 D6 03 25 00 31    mov     cs:log_filename+0Dh, 31h ; '1'
0040A3A7 C6 05 D0 03 25 00 2F    mov     cs:log_filename+0Eh, 2Fh ; '/'
0040A3AE C6 05 CA 03 25 00 63    mov     cs:log_filename+0Fh, 63h ; 'c'
0040A3B5 C6 05 C4 03 25 00 6F    mov     cs:log_filename+10h, 6Fh ; 'o'
0040A3BC C6 05 BE 03 25 00 72    mov     cs:log_filename+11h, 72h ; 'r'
0040A3C3 C6 05 B8 03 25 00 65    mov     cs:log_filename+12h, 65h ; 'e'
0040A3CA C6 05 B2 03 25 00 64    mov     cs:log_filename+13h, 64h ; 'd'
0040A3D1 C6 05 AC 03 25 00 75    mov     cs:log_filename+14h, 75h ; 'u'
0040A3D8 C6 05 A6 03 25 00 6D    mov     cs:log_filename+15h, 6Dh ; 'm'
0040A3DF C6 05 A0 03 25 00 70    mov     cs:log_filename+16h, 70h ; 'p'
0040A3E6 C6 05 9A 03 25 00 2E    mov     cs:log_filename+17h, 2Eh ; '.'
0040A3ED C6 05 94 03 25 00 69    mov     cs:log_filename+18h, 69h ; 'i'
0040A3F4 C6 05 8E 03 25 00 6E    mov     cs:log_filename+19h, 6Eh ; 'n'
```

String stacking in a binary

Credential stealing

Various methods to steal users credential on both sides.

Client

Modify functions on client to log password on log-in such:

USERAUTH_PASSWD, Authenticates a session with username and password.

SSH_ASKPASS, Pass-phrase dialog.

Server

Modify functions on server to log password on request such:

AUTH_PASSWORD, Tries to authenticate the user using password.

SSHPAM_RESPOND, Tries to authenticate the user with PAM (Pluggable authentication modules).

Exfiltration methods

Once credentials are stealed, attackers need to exfiltrate them:

Exfiltration by local file

Easy method: credentials are stored inside a file in the server,
hidden in filesystem (e.g.: .so in /usr/bin or .h in /usr/local/include).
Problem: attackers needs to have a way back into the system.

Exfiltration by C&C server

Complex method: send credentials over the network instead of local file.
Problem: network communications are logged.
Some backdoor encrypt communication with a symmetric key.

Exfiltration by email

In some rare cases credentials are sent by email.
Problem: hardcode email address in the binary.

Backdoor mode

Permanent Method to connect back to the compromised machine,

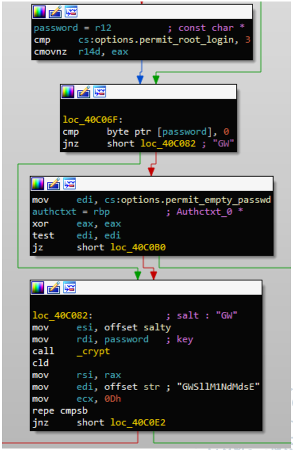
with the following features:

Hardcoded password, compare client password with a hardcoded password.

Configuration and log, change daemon configuration to permit full access and disable logging features in order to not leave traces on the system.

Environment variables, change environment variables such as HISTFILE.

Hooked functions, modify all functions for logging and debugging.



Backdoor password verification

Part III

Backdoors families

OpenSSH backdoors galaxy

OpenSSH backdoor families according to ESET research

OpenSSH backdoors summary (1/2)

Family	Network exfiltration	Local exfiltration	Backdoor mode	Source Code available	Documented	Anti-logging	Obfuscations used
Abafar	-	✓	✓	✓	✓	✓	-
Alderaan	-	✓	✓	✓	-	-	-
Anoat	-	✓	✓	-	-	✓	String-stacking
Akiva	-	✓	✓	✓	-	-	-
Ando	SMTP	✓	✓	✓	-	✓	-
Atollon	-	✓ encrypted	✓	-	-	✓	Encrypted strings and string-stacking
Batuu	-	✓ encoded	-	-	✓	-	-
Bespin	-	✓	-	-	-	✓	-
Bonadan	UDP	✓	✓	-	-	-	-
Borleias	UDP	✓	-	-	-	-	-

OpenSSH backdoors summary (2/2)

Family	Network exfiltration	Local exfiltration	Backdoor mode	Source Code available	Documented	Anti-logging	Obfuscations used
Chandrila	UDP	✓	✓	-	-	-	Encrypted strings
Crait	UDP	✓	✓	-	-	✓	Encrypted strings
Coruscant	HTTP	✓	✓	-	-	✓	-
Endor	SMTP	✓	✓	✓	-	✓	UPX (Some variants only)
Jakku	HTTP	-	✓	-	-	✓	Encrypted strings
Kamino	HTTP	-	✓	-	✓	✓	Encrypted strings
Kessel	HTTP, TCP, DNS	✓ encrypted	-	-	-	✓	Encrypted strings
Mimban	TCP	-	✓	-	-	✓	Encrypted strings
Onderon	-	✓	✓	✓	-	✓	-
Polis Massa	SMTP	✓ encoded	✓	✓	-	✓	-
Quarren	-	✓	✓	-	-	✓	-

Chandrila

Save authentication method, username and password base64-encoded.

Exfiltration of credentials via local file or sent via UDP to a C&C server.

Distinctive feature: **can receive commands through the SSH password**.
Two passwords are hardcoded in the backdoor: one to login in the server and another to execute commands, by appending data to the password.

In particular:

C0011455OpenSSHd backdoor password for command line.
C001145SOpenSSHd${CMD} execute CMD on server.

Powerful backdoor mode as attacker can execute command without a shell.

Bonadan

Backdoor fork a new thread inside the main function: this thread periodically calls two functions and pause for five minutes.

First function check if there is any **cryptocurrency miner** installed on the system and removes it.

Second function connect to the C&C server and send several information about the host over UDP (such as current username, OS version, external IP address, CPU and RAM models, speed of the miner).

Backdoor receive an answer from the C&C server that can containing a specific command like: create a shell, execute a command on machine, updates the configuration, launch a cryptocurrency mining module.

The backdoor mines the **Monero cryptocurrency** as part of a mining pool.

Problem: need to store wallet information inside the server.

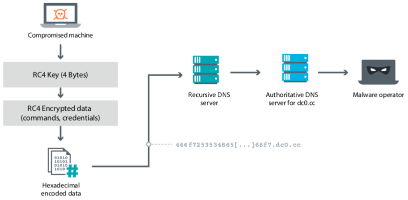
Kessel

Most advanced and recent between all the families. This backdoor includes two main features: a bot functionality and credentials stealing.

Bot feature: a bot is launched at the beginning of the OpenSSH main function, generate an ID based on MAC address, collect system information then starts two threads.

First thread periodically send encrypted information to a C&C server and eventually create a reverse shell with a specified machine. Second thread repeatedly queries a TXT records in a custom DNS server to get commands.

Credential stealing: reimplements two malicious functions in SSH daemon (SSH_LOGIN and USER_AUTH_PUBKEY) to steal remote host/username, password and local username (or private/public key). Exfiltrate stolen credentials using HTTP, TCP or DNS protocol.



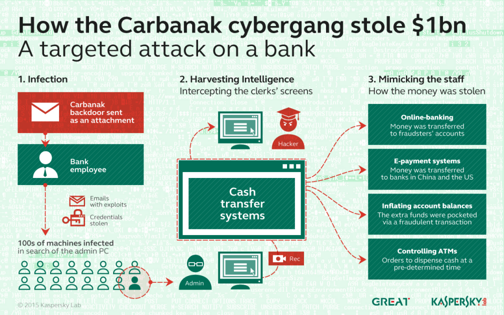
DNS exfiltration schema

Kamino

First variant of this backdoor already seen in 2013 (as documented in **[3]**).

Used together with an Apache module called DarkLeech to redirect internet traffic.

After few years the same backdoor was used again to attack Russian Banks by a group known as Carbanak, resulting in ~**900 million dollars**.



Carbanak attack (source: Kaspersky[7])

Part IV

Honeypot

Definition and goals

A honeypot is a **ICT resource** used for monitoring, detecting and analyzing attacks. It has no production value, no real sensible data.

Can be classified by its implementation (virtual or physical), purpose (prevention, detection, research) and level of interaction (low, middle, high) in particular:

Low-interaction honeypots simulate only some aspects of the system, while limiting the ability of the attacker. More easy to deploy but can get a limited amount of information.

High-interaction honeypots based on a real OS, the attackers gets full access to the system and can be compromised completely (with an higher risk). Harder to deploy and manage but can get a full overview of the attackers.

Many OpenSSH honeypot solutions exists but due to their popularity they are easy to detect.

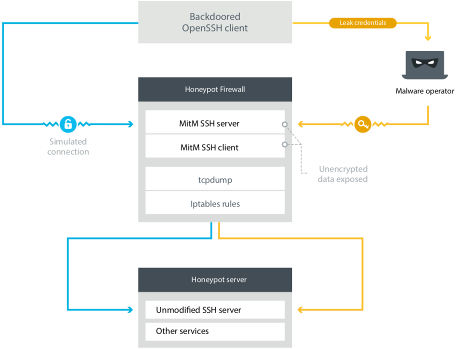
An high-interaction custom honeypot was chosen in order to maximize the information gathered from the attackers.

Honeypot structure and strategy

Credentials are leaked in two different ways.

For a client backdoor is enough to **log into the honeypot server**.

More complicate is for a daemon backdoor, one solution is to **install the daemon backdoor** on the honeypot server, log in with a safe client and remove the backdoor, in order to simulate the backdoor detection in the attacker eyes.



Honeypot infrastructure

Observed interaction: Minban

Both client and server backdoors were available, of course the backdoor client was used to leak the credentials.

The attackers behind the backdoor logged in to the honeypot after few hours of leaking and thanks to the architecture the commands executed on the server were logged.

```
# First connection from 31.184.196[.]57 (RU).
ls -la
ps aux
df -h
unset HISTFILE
cd /var/www
ls
ls -la html/
exit

# Second connection from 31.184.196[.]57 (RU).
unset HISTFILE
ls -la
cat .bash_history
cat /etc/shells
cd /var/lib/mysql
ls -la
ps aux
w
ssh -V
unset HISTFILE
exit
```

Attackers command captured for Minban

Attackers logged in manually from a **Russian IP address**, try to clean command history, check some files and finally **check the version of OpenSSH binary**.

Observed interaction: Borleias

More interesting the results of the Borleias backdoor: the attackers first logged into the honeypot in less than 24 hours and repeat for **more than 10 times** within four days.

- The attackers used **Tor** to login, in order to not leave trace.
- A classic version of the OpenSSH client was used to connect.
- They managed to get the credential leaked in the Mimban operations: this means that there is a **connection between the two backdoors** (maybe same operators or the credentials were sold somewhere).
- Some basic checks at beginning then more interesting operations in the few days later, in particular they drop a **new version of the backdoor**.
- Gather all the information about the server, exfiltrate them and clean all the command history.
- **Meticulous attention to details** (check of running process, timestamps of files, logged-in users between the execution of each command, ...)

Part V

Mitigation

Preventing compromise of SSH servers

Very difficult to determine the infection vector used to install these OpenSSH backdoors.

In the operation Windigo two different vector was used:

An attack of the website ***kernel.org*** (the official repository of the linux kernel source code) to inject malicious code.

An attack to ***cpanel.net***, the most famous software to manage websites and hosting.

Obvious but important recommendation is to install software only from trusted sources, e.g. checks sources in /ETC/APT/SOURCES.LIST for debian-derived linux distribution or install only trusted package from the AUR (Arch User Repository) for ArchLinux distribution.

Correct OpenSSH configuration

It is possible that some attackers could be using **brute-force** to gain access through SSH password authentication.

Good practice is to have long and complex passwords in order to prevent a successful brute-force. A better solution is to **disable password authentication** and use only a key-based authentication.

Set PasswordAuthentication no in /etc/ssh/sshd_config

If password login is the only choose setup a limit to failed attempt.

Another good practice (almost standard today) is to **disable remote root login**, in order to prevent login without a named user account. A user can have administrative privileges but can be easily identifies instead of share root password among admins.

Set PermitRootLogin no in /etc/ssh/sshd_config

The most efficient solution would be to use a **multi-factor authentication** (such as SMS-based verification, a security token or other). OpenSSH still doesn't support this kind of authentication but it can be achieved through an external extension like *google-authenticator-libpam*.

Check logs and network traffic

Enable logs for every critical service.

Periodically **backup logs** in an external server/device.

Check for suspicious operations in log files on the server.

```
Jul 17 16:29:24 informateci sshd[1707]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=37.187.19.222  user=root
Jul 17 16:29:26 informateci sshd[1707]: Failed password for root from 37.187.19.222 port 37893 ssh2
Jul 17 16:29:26 informateci sshd[1707]: Received disconnect from 37.187.19.222 port 37893:11: Bye Bye [preauth]
Jul 17 16:29:26 informateci sshd[1707]: Disconnected from 37.187.19.222 port 37893 [preauth]
```

Failed attempt of login in a server

Setup a firewall and try to monitor all suspicious network traffic (difficult if a lot of traffic is present but can be automatized), block traffic from all unused ports.

Detect compromised SSH tools

A **Malware Indicators of Compromise** (IoC) by ESET is publically available `https://github.com/eset/malware-ioc`.
The tool can detect malicious OpenSSH files on the system.

A good operation is to **verify integrity** of OpenSSH binaries but ELF file format does not support signatures (unlike PE in Windows or Mach-O on macOS).

On a Debian-based distribution commands DEBSUMS or DPKG -V can be used to compare installed softwares with a manifest stored on disk, however a local manifest can be easily changed. The only solution is to compare the metadata of the .DEB package against the one in the **Debian official repositories**.

Backdoor could also be in an **shared library** instead of client/daemon binary: a modified library can change the behaviour of any application that use it. None of the 21 families seems to use this technique, however Ebury backdoor comes with an **altered version of *libkeyutils.so***, which is loaded by all OpenSSH process.

Conclusion

- A compromised server can compromise other servers and millions of users (see Carbanak operation).
- Install software only from trusted origin.
- Keep system up-to-date (upgrade always to the last *stable* version).
- Correct configure software.
- Prefer public/private key login instead of password login.
- Setup a multi factor authentication.
- Check logs periodically for suspicious activities.
- Keep track of external connections and setup correctly a firewall.

References

[1] The Dark Side of the ForSSHe, ESET
https://www.welivesecurity.com/wp-content/uploads/2018/12/ESET-The_Dark_Side_of_the_ForSSHe.pdf

[2] The Dark Side of the ForSSHe, ESET
https://www.welivesecurity.com/2018/12/05/dark-side-of-the-forsshe/

[3] Linux/SSHDoor.A Backdoored SSH daemon that steals passwords, ESET
https://www.welivesecurity.com/2013/01/24/linux-sshdoor-a-backdoored-ssh-daemon-that-steals-passwords/

[4] Operation Windigo, ESET
https://www.welivesecurity.com/wp-content/uploads/2014/03/operation_windigo.pdf

[5] ESET discovers 12 previously undetected Linux backdoors, ESET
https://www.eset.com/int/about/newsroom/press-releases/research/eset-discovers-12-previously-undetected-linux-backdoors/

[6] Openssh backdoor used on compromised Linux servers, randhome
https://www.randhome.io/blog/2016/08/01/openssh-backdoor-used-on-compromised-linux-servers/

[7] Carbanak APT: The great bank robbery, Kaspersky
https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08064518/Carbanak_APT_eng.pdf