Analysis of WellMail malware's Command and Control (C2) server

pwc.co.uk/issues/cyber-security-services/insights/wellmail.html

Copy link

17/09/20

Earlier this year, we published analysis of WellMess malware that had been used to target organisations involved in COVID-19 vaccine development. In a follow up article, we gave further technical analysis of its Command and Control (C2) server.

Our investigation into WellMess has also included a second C2 server from a closely related malware family called WellMail.

Again the threat actor behind the tool has demonstrated strong operational security considerations as part of its development and usage of the backdoor, but left a small amount of evidence which indicates it has been targeting COVID-19 research. The malware also follows in WellMess's footsteps of utilising a two-tier C2 protocol to distance the threat actor from any of its backdoors.

In this article we look at the capabilities of WellMail – the analysis is taken from our private intelligence reporting services.

WellMail analysis

Our previous WellMess reporting found common strings in the client and server code of:

```
C:/Server/BotUI/App_Data/Temp
```

By carrying out a similar analysis on a WellMail backdoor sample from the July NCSC advisory, we identified another file not included in the advisory.

Filename	mail.sh
Filetype	ELF 64-bit
SHA-256	93e71fa0f4c5909a5b69033ac39b4664d10e9ed35fa995cf797e3a9990fbb751
First seen	2020-05-04 03:41:50

Both the WellMail backdoor and this new sample contain the string:

```
C:/Server/Mail/App_Data/Temp
```

This new file is written in Go and is highly likely a first stage C2 for receiving and sending data to the WellMail backdoor. This C2 only supports communication via mutual TLS and does not support communications via unsecured channels. However, just like the WellMess C2, the WellMail C2 supports additional commands that are not present in the WellMail backdoor and are highly likely used by the threat actor to manage uploading and downloading of victim data from the C2.

The similarities between this WellMail C2 and the previously mentioned WellMess C2 go beyond just the environment paths embedded in the binaries, as the way the servers are designed is also extremely similar. A breakdown of shared details is shown in Table 1.

Table 1 - C2 similarities

Feature	Description
Platform	Both servers are written in the Go programming language and compiled as 64-bit ELF files
Network protocol	Both servers implement communication via mutual TLS with HTTP

CA certificates	Both servers have hard coded Certificate Authority (CA) certificates with similar metadata
SSL certificates	Both servers require the current working directory to contain an SSL private key and certificate with filenames of serverkey.pem and server.crt respectively.
Logging	Both servers implement the open source Go library, lumberjack, to log runtime information
Cookie control	Both servers use the names and contents of Cookie headers from clients to specify commands to execute
Data storage	Both servers use specific folders to send and receive data for each command
Architecture	Both servers work as a staging point to receive data from malicious backdoors and hold it until commands from an upstream device request the data

Each HTTP request received from a connected client is parsed to gather the Cookie headers in the request which are used to determine the command to carry out. The name of the received cookie is used as the command to execute and the contents of the cookie are optionally used as a parameter to pass to the command handler. The supported commands that can be received by this WellMail C2 are detailed in Table 2.

Table 2 - Supported commands

Cookie Name	Description
first	Initial beacon from the WellMail backdoor containing victim environment information
inbox	Receives data from the WellMail backdoor and saves it to the C2
dwnmail	Stages the first inbox or error data found to be exfiltrated
spam	Saves a script file to the C2 ready to be sent on to a WellMail backdoor

new	Searches for data received by the first command and sends it back to the client
error	Saves the received data as a tmp.error file and stores it in the same place as the inbox data
delete	Deletes a file on the C2 specified by a filepath in the cookie contents
script	Searches for script files to upload to the WellMail backdoor and sends any found

WellMail capabilities

The WellMail backdoor has very limited functionality and the following analysis is taken from a 64-bit ELF sample.

Filename	vigrd
Filetype	ELF 64-bit
SHA-256	85e72976b9448295034a8d4c26462b8f1ebe1ca0a4e4b897c7f2404d0de948c2
First seen	2020-05-25 12:42:22

The malware uses a hardcoded SSL private key, certificate and CA certificate to authenticate with the server via mutual TLS. All communications are carried out using the net.http Go package using a hardcoded C2 IP address and POST requests.

An initial beacon is sent to the C2 which contains a cookie with a name of first and a value that contains:

```
hard_coded_string%2FMD5_hash
```

The hardcoded string is taken from the malicious binary and the MD5 hash is generated from the victim's IP address, USER environment variable and the previously mentioned hardcoded string. The content of the POST request consists of a | separated string containing:

- The victim's IP address,
- The victim's USER environment variable; and,

- The MD5 hash from the cookie value.

An example would be:

```
1.2.3[.]4|root|MD5_hash
```

The response from the C2 is checked to see if it contains any cookies. If there are any, it then attempts to extract and run a gzipped script from the content of the response.

After the initial beacon the malware then generates a random number between two and four which it uses to sleep between subsequent connection attempts to the C2.

Each of these connection attempts creates 1,847 random bytes of data to use as the content of the POST request and sets the cookie name as script with a value of the hardcoded string. The response is checked for cookies and runs a script in the same way as the initial beacon.

The scripts are run by saving the received script from the C2 to disk and passing it into os.exec.Command with /bin/sh and running it with the output saved to a .gz file. If any errors occur during the command's execution, then the malware sends a POST request with a cookie named error. If no errors occur, then the saved file is sent as the contents of the POST request with a cookie named inbox.

In summary, the WellMail backdoor supports the commands:

- first,
- inbox,
- script; and,
- error.

This leaves the following commands, supported by the WellMail C2, unaccounted for:

- dwnmail,
- spam,
- new; and,
- delete.

Potential second stage C2

The functionality of these unused commands matches those expected to assist exfiltration of data from the C2 that was received from WellMail backdoors. Figure 1 shows the expected sequence of commands that the WellMail C2 will handle to transfer data between the threat actor and a WellMail backdoor.

Figure 1 - WellMail communication flow

Figure 2 - Communication key

The WellMail C2 analysed lacks the capability to send the responses of scripts run by the WellMail backdoor onto a further threat actor controlled box. However, looking at the implementation of the dwnmail command, there are suggestions that the threat actor uses alternative software running on the WellMail C2 in order to read the data returned from the WellMail backdoor.

The dwnmail command searches for any .gz or .error files on the C2 that have been sent from a WellMail backdoor by an inbox or error command. If it finds a file, then it copies it to the /var/www/html/mail/ folder and also sends back the fully qualified file path of the new file as the contents of the 200 OK response. This folder is one of the default locations that linux email servers use as their root directory. It is possible that the threat actor has set up an email server on the C2 that it logs into in order to retrieve the files stored on the C2. After accessing the files by this method, the threat actor can send a delete command with the file path returned by the dwnmail command in order to remove the sensitive material from the C2.

It is likely that the threat actor behind WellMail and WellMess accesses its C2 machines from behind Tor and while using a VPN service. We have observed a substantial amount of connections being made to the C2 IP addresses from several different commercial VPN providers. These same C2 IPs also have connections from Tor exit nodes, which could be occurring due to a dropped connection from the VPN provider. This would align with the strong OPSEC practices observed from the threat actor behind these tools in having multiple layers of security in case one fails.

Targeting

When looking at the environment paths found in the WellMess and WellMail samples a pattern emerges. Both of the C2 binaries have /G/ID after the /Temp folder in their strings, whereas the backdoor samples for WellMess and WellMail from 2019 and 2020 both have just the /ID in the string. The ID in most samples appears to be a number, which presumably increments with each new target of the malware. We have seen samples with numbers up to 73, indicating that there have potentially been 73 targets of the WellMess and WellMail malware. When the ID is not a number it contains a string, which is likely related to either the device or service the malware is masquerading as or the organisation that is being targeted.

The ID values we observed are:

- 4
- 12
- 36
- 72
- 73
- watchdogd
- migration

- agent.sh
- wdns
- SangForPromote.exe

Two of the targeted entities likely correspond to a Canadian vaccine research company and the University that stood up the Research Headquarters for Epidemic Prevention and Control with the Chinese Academy of Science. This would align with the claims by NCSC that the threat actor behind WellMess and WellMail has targeted institutions researching a COVID-19 vaccine.

Conclusion

The WellMail C2 is extremely similar to the WellMess C2 making it almost certain that the two tools are developed and deployed by the same threat actor. The WellMail C2 and backdoor have slightly less functionality than the corresponding WellMess malware but are still effective for running malicious scripts on victim machines.

It is unclear based on the slight differences in functionality whether the two malware families are used in tandem or whether one is an evolution of the other and is now preferred by the threat actor.

The presence of particular file paths in the two malware families point to exploitation of COVID-19 research facilities by the threat actor, which confirms the information previously put forward by NCSC with regards to targeting.

We continue to track the WellMess and WellMail malware.

- TTP’s
- IoC's

TTP’s

Command and Scripting Interpreter: Unix
Shell - https://attack.mitre.org/techniques/T1059/004/

Application Layer Protocol: Web Protocols- https://attack.mitre.org/techniques/T1071/001/

Data Encoding: Standard Encoding - https://attack.mitre.org/techniques/T1132/001/

Exfiltration Over Command and Control Channel - https://attack.mitre.org/techniques/T1041/

Indicator	Type
85e72976b9448295034a8d4c26462b8f1ebe1ca0a4e4b897c7f2404d0de948c2	SHA256

93e71fa0f4c5909a5b69033ac39b4664d10e9ed35fa995cf797e3a9990fbb751	SHA256
52c6651b6bd0c5940fd0de8e885f5ef8d0292142	SHA1
5e0b5869d98c93cd7f7d925f04a49bd638590ec0	SHA1
d5c26128127f2fac6e3ff2c87b473d74	MD5
3293b12b7622f484b69819217ed8af85	MD5
vigrd	Filename
mail.sh	Filename
111.90.150[.]140	IP

Contact us

Form

Hide