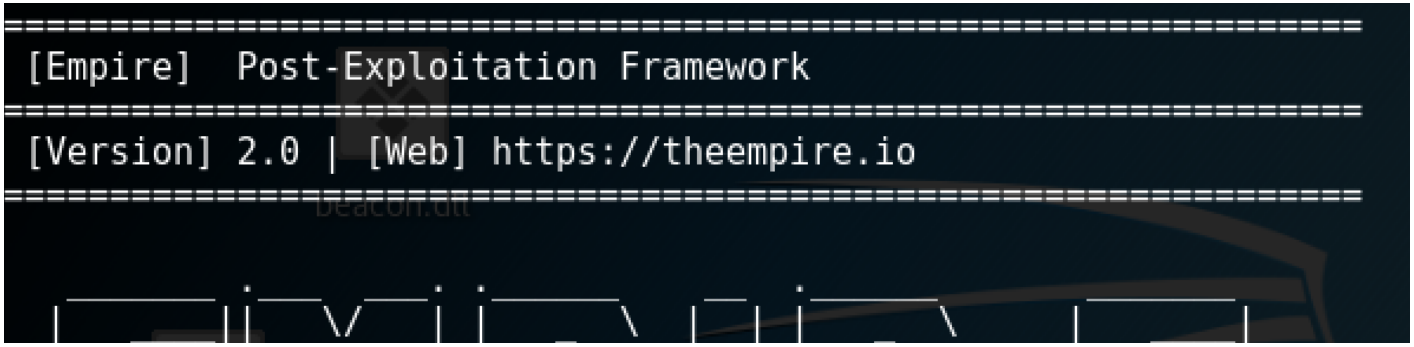
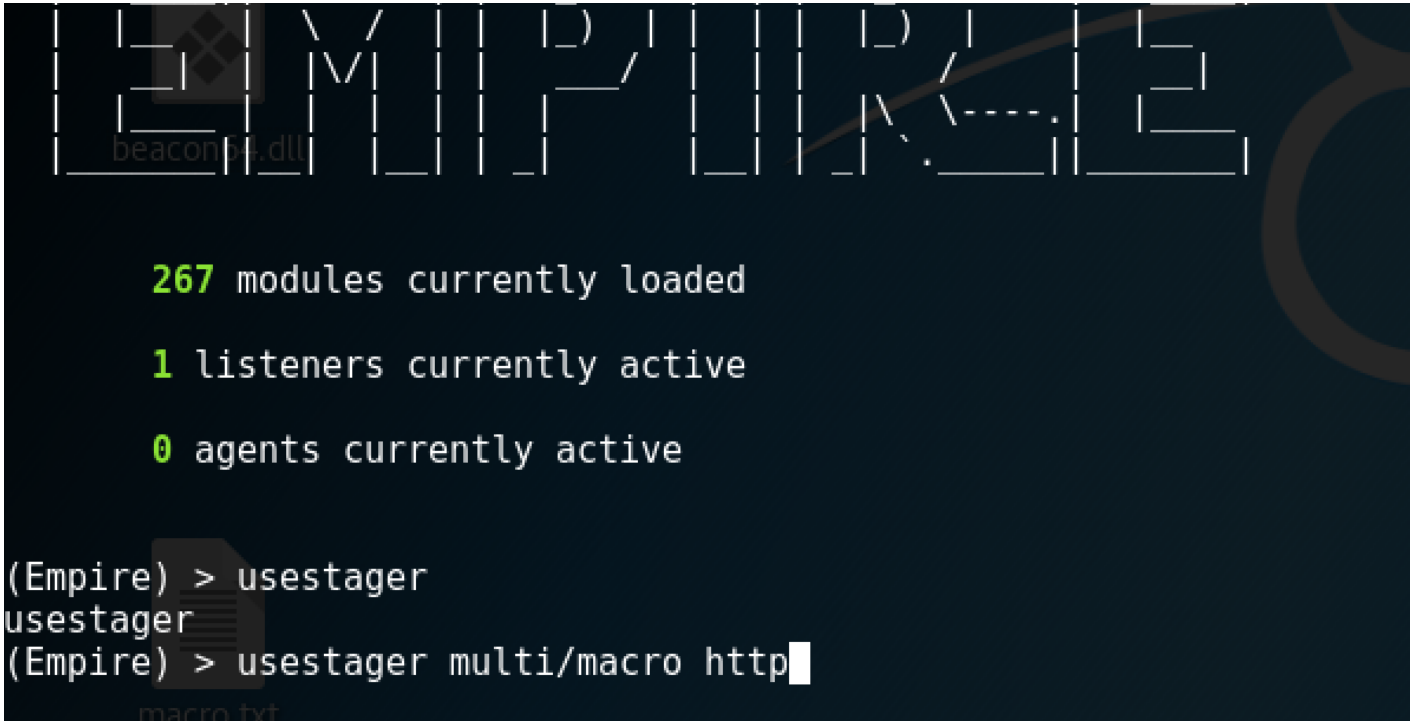
Tim MalcomVetter Follow
Red Team Leader at Fortune 1. I left my clever profile in my other social network: https://www.linkedin.com/in/malcomvetter
Aug 23, 2017 · 3 min read

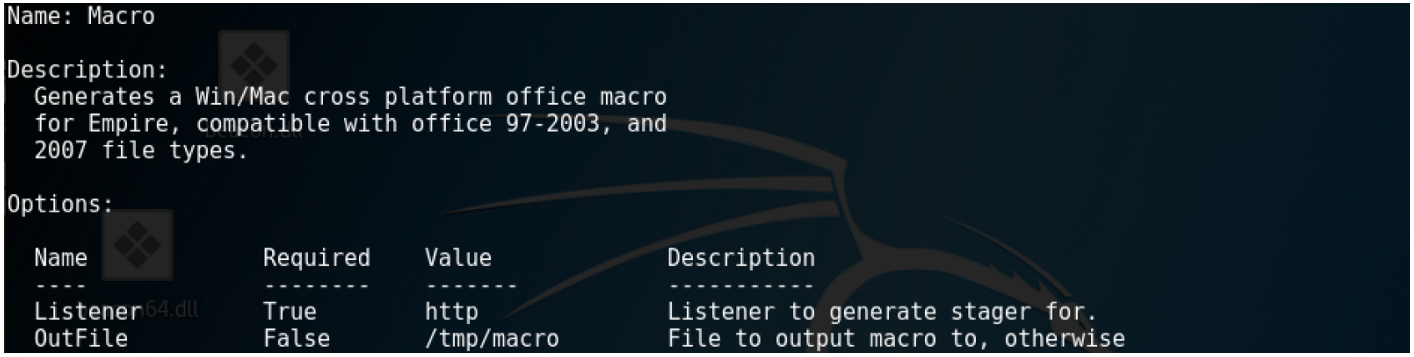
Multi-Platform Macro Phishing Payloads

With enterprises deploying Mac endpoints, a red team phishing panacea just might be a Microsoft Office Word document or Excel spreadsheet that has an OS agnostic macro payload. The team at Black Hills InfoSec have a good walk through how to do that manually with Empire. My team stumbled upon a similar setup using a combination of Empire and Cobalt Strike payloads, but now … the hard work is done for you: we have published an updated Empire module based on @enigma0x3's and @harmj0y's original platform specific macro modules.





In Empire, setup a listener and then run “usestager multi/macro” followed by “info” to see and set options.



```
                                        displayed on the screen.
 Proxy           False      default     Proxy to use for request (default, none,
                                        or other).
 SafeChecks      True       True        Switch. Checks for LittleSnitch or a
                                        SandBox, exit the staging process if
                                        true. Defaults to True.
 Language        True       powershell  Language of the stager to generate.
 PixelTrackURL   False      http://127.0.0.1/tracking?source=URL to add in pixel tracking which OS
                                        attempted macro opening, useful for
                                        shell debugging and confirmation.
 UserAgent       False      default     User-agent string to use for the staging
                                        request (default, none, or other).
 ProxyCreds      False      default     Proxy credentials
                                        ([domain\]username:password) to use for
                                        request (default, none, or other).
 StagerRetries   False      0           Times for the stager to retry
                                        connecting.


(Empire: stager/multi/macro) >
```

This will generate a macro that you can copy/paste straight into the macro editor of your document for delivery to your phishing target.

On the top of the macro, there are some platform detection if statements that will load up the necessary dylibs for Macs running Office 2016, which by default will run inside the MacOS sandbox. After that are the usual hooks of the macro functions that fire when documents are opened. The rest of the document is an if statement branch for OS version. If MacOS, there is a shell command to call curl to a tracking URL, followed by a python specific payload just like in the previous Mac specific Empire macro payload. If Windows, there

is an XMLHTTP call to a tracking URL, followed by the powershell specific payload just like in the previous Windows specific Empire macro payload.

The tracking URLs allow the adversary to do some analytics around whether the document was opened without a shell phoning home (i.e. endpoint protections or network IPS blocking execution or communication). Tracking URLs come in handy when a new shell phones home to Empire and you need to figure out if the target opened the macro again or if a persistence hook fired instead. By default, the Tracking URLs point to localhost, so no real impact to OPSEC if you forget to set them, but you can choose to use a third-party pixel/ad tracking service, or you can spin up your own webserver on your own domain and substitute that URL into the Empire options.

If you want to use different payload types per OS, you can use Empire to generate the multiplatform macro and then replace the OS specific sections of the if branch. For example, generate a Cobalt Strike macro payload and then copy/paste the bulk of that payload into the Windows section over the top of the Windows Empire payload in this new Empire multiplatform macro. That way you can switch up TTPs and still retain cross-platform support.

Once you get a call back from a MacOS instance of Office 2016, you'll note your working directory is probably "~/Library/Containers/com.microsoft.Word" or similar. The Library Container path is an indicator you're in the sandbox and you'll be limited on what you can do. Empire used to have some Mac specific bugs, especially when executing within the sandbox, such as determining the host's IP address. The sandbox puts limitations on python's ability to use the socket library, so we introduced some exception handling and some bash magic to scrape the IP out of "ifconfig" to fix it.

One action an adversary can perform while in the sandbox is still to use the venerable AppleScript ("osascript") tool to generate a phishing prompt for the user's password, but there are sandbox limitations affecting that tool as well. However, @DisK0nn3cT from my team added a neat little "sandbox mode" feature to that Empire prompt module ("usemodule python/collection/prompt") to evade some of the sandbox's limitations using legacy MacOS file paths to present the icons and staying within the current process (the normal mode spawns a new process). So an adversary can still put up a good MacOS phish prompt.

Big thanks to my team, especially @DisK0nn3cT, @h1ghtopfade, and “Not Charles” for help with development and testing.

Hacking | Penetration Testing | Infosec

Like what you read? Give Tim MalcomVetter a round of applause.

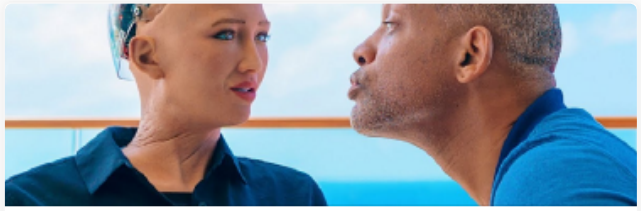
From a quick cheer to a standing ovation, clap to show how much you enjoyed this story.

169

Tim MalcomVetter

Red Team Leader at Fortune 1. I left my clever profile in my other social network: https://www.linkedin.com/in/malcomvetter

Follow



Also tagged Hacking

How I Hacked Into One of the Most Popular Dating Websites

Zaid Daba'een

4.7K

Also tagged Hacking

How North Korean hackers became the world’s greatest ban...

Patrick Winn

5.8K

Also tagged Hacking ★

The Man Who Cracked the Lottery

New York Times ...

1.8K


11 min read


17 min read


10 min read

Responses

Write a response...