Operating with EmPyre

By Steve Borosh

Published: 2018-04-11 · Archived: 2026-04-05 18:27:25 UTC

If you're reading this post, I sincerely hope you've already started with reading @harmj0y's first blog post about EmPyre located here:http://www.harmj0y.net/blog/empyre/building-an-empyre-with-python/

This post is second in what will be a great series to help user's understand and operate using EmPyre. Let's get started!

Operating in an OS X environment may seem like a daunting task. Many people are under the assumption that merely using a Mac computer makes you or your organization secure. This blog post will cover why that is not necessarily true, how an attacker can effectively operate in an OS X or mixed environment and what defenders can do to avoid having their OS X infrastructure breached.

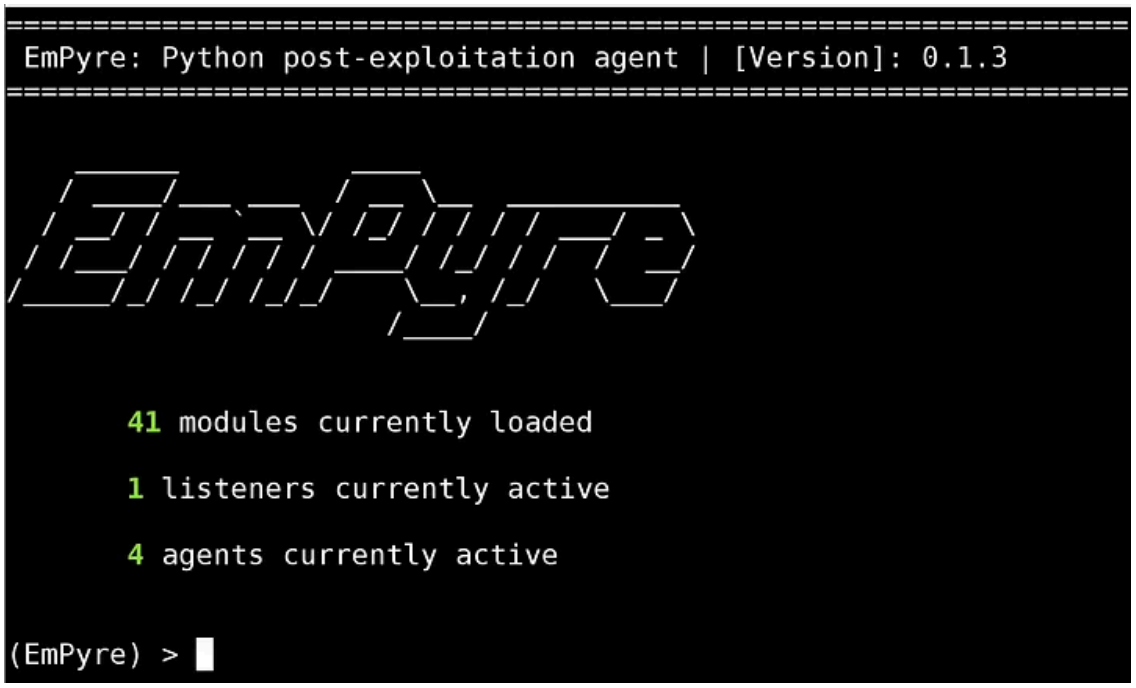
Every year, CVE Details reports on the number of distinct vulnerabilities found in software and operating systems. In 2015 OS X topped the list with a recorded 416 reported vulnerabilities.

Press enter or click to view image in full size

	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Mac Os X	Apple	OS	416
2	Iphone Os	Apple	OS	384
3	Flash Player	Adobe	Application	314
4	Air Sdk	Adobe	Application	246
5	AIR	Adobe	Application	246
6	Air Sdk & Compiler	Adobe	Application	246
7	Internet Explorer	Microsoft	Application	231
8	Chrome	Google	Application	187
9	Firefox	Mozilla	Application	178
10	Ubuntu Linux	Canonical	OS	166
11	Windows Server 2012	Microsoft	OS	155
12	Windows 8.1	Microsoft	OS	151
13	Windows Server 2008	Microsoft	OS	150
14	Windows 7	Microsoft	OS	147
15	Windows 8	Microsoft	OS	146

Compare that to Windows Server 2012's 155 and you can see a huge difference in statistics. Operating from an OS X platform certainly does not mean you are more secure these days. Malware authors have taken notice to the rising market share of the OS X operating system and the numbers of malware for OS X are also climbing. In 2016, Carbon Black released a report titled "2015: The Most Prolific Year for OS X Malware". In this report, Carbon Black research found 948 malware samples compared to 180 total for the years 2010 to 2014. Attackers are certainly finding ways to operate on the OS X platform.

Press enter or click to view image in full size



As an attacker or security tester, options have been very limited on how to conduct operations against targets utilizing OS X. The Italian security firm “The Hacking Team” utilizes home-grown implants known as the Remote Code Systems (RCS) compromise platform to operate on OS X environments. For the rest of the world, you either have to create your own Remote Access Trojan or find another method for continuous operations. Enter EmPyre, an OS X/Linux offshoot of the PowerShell Empire project.

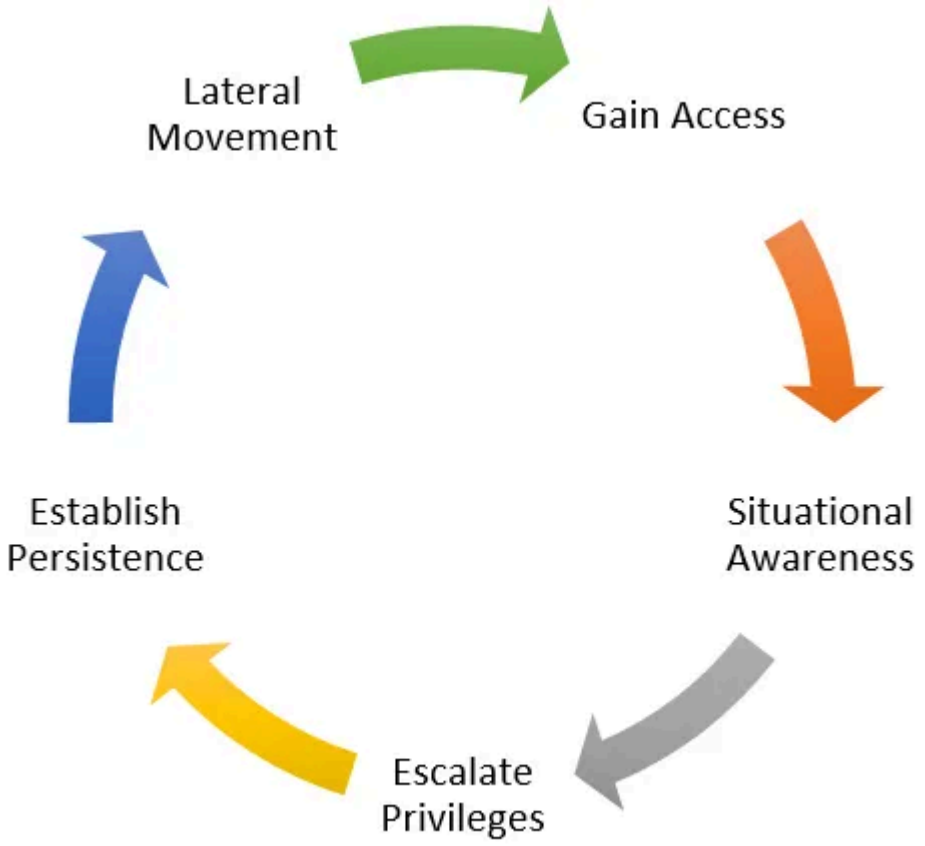
EmPyre was initially developed by @harmj0y in response to a client’s need for testing OS X platforms. The initial post on EmPyre may be found **here** containing more details of the RAT’s infrastructure and communications platform. I’m going to cover some of the tradecraft that was built into the RAT to support continuously operating in an OS X environment.

Get Steve Borosh’s stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

While operating from an OS X environment brings its own challenges, the methodology commonly used for penetration tests or red teams still applies.



During a penetration test or red team, gaining access can either be manually seeded by the client Point of Contact (POC) or phishing may be required. If you must conduct phishing to gain access, options are limited in comparison to those available to testers targeting Windows systems. Currently, EmPyre supports two payloads that may be used in a phishing attack. These payloads are Microsoft Office macros or an HTML page that calls an Applescript launcher based on CVE 2015–7007.

OS X environments provide some native situational awareness commands that typically aren't available on other operating systems. Some examples are "pbpaste" for grabbing clipboard contents, "screencapture" for grabbing screenshots, and "curl" which and be useful for downloading files or data exfiltration. For the EmPyre RAT, we've used some lessons learned during operations to decrease the chances of detection. We've also taken these native methods and created non-native Python modules that perform the same action and are harder to detect. With EmPyre you're also able to run port scans, query active directory, dump hashes, and perform all the standard post-exploitation functions.

Press enter or click to view image in full size

```
Description:
  Extracts found user hashes out of
  /var/db/dslocal/nodes/Default/users/*.plist

Options:

  Name            Required    Value                   Description
  ----            --------    -------                 -----------
  Agent           True        EAPT3NMG9JJMUVCV        Agent to execute module on.

(EmPyre: situational_awareness/host/osx/hashdump) > execute
(EmPyre: situational_awareness/host/osx/hashdump) >
[('administrator', 'ml$38167$abf5b16afd213da5f741f7e7022460f0890fc1d9795e431ae463a68439e64e
cd9e4e4a2918eb0ca0a8c9720301d6cacc6deb92a186532a43d7828402747e046b2ab418475342400267410c74a
b45c2b43b96a0e2900e1d234c8f25582ce29598f349fb63d7428edc03a3f6ba5202133071f3c7e7101c71753')]
```

Privilege escalation from EmPyre is currently limited to spawning an agent using the “sudo” command. There have recently been several local privilege escalation exploits released for OS X in 2016. These have yet to be built into EmPyre and would be great way for the community to provide support to the project.

OS X has several mechanisms available to obtain persistence. Cronjobs allow for time-based persistence, login hooks allow for user login persistence, launch daemons that persist through reboots and much like Windows DLL hijacking, there’s DyLib hijacking based on the research of @patrickwardle. All of these methods have been built into EmPyre.

Finally, lateral movement is the last portion of tradecraft to cover. With Windows, there are many luxuries such as WMI, Pass-the-hash, executing files over UNC, WinRM and Remote Desktop Protocol. OS X provides us with SSH, if it’s enabled. Lessons learned from engagements show that it is usually turned on in a corporate environment as administrators need to admin somehow. EmPyre has modules to either launch SSH commands or send a launcher string for a new agent to a remote host. Pivoting from OS X to Windows becomes even trickier as there currently isn’t a solid Pass-the-Hash solution for OS X. EmPyre does, however, have a module to exploit JBoss on Windows via Java Serialization and that can send an agent callback to another Empire server.

In closing, we now understand that organizations who utilize OS X as a security boundary may not be doing themselves justice without a proper defense-in-depth approach. As research shows, OS X is prone to vulnerabilities just like other operating systems and software. With the proper tools such as EmPyre, a security tester can effectively perform security testing through a pure OS X or mixed environment. What does this mean for the blue teams out there? Email filters, blocking macros, host-based protection, network heuristics and log aggregation all still play in the defense-in-depth approach. We know that OS X has several commands that we should look for. Most of your users aren’t going to be running “pbpaste” from the terminal. Most users aren’t going to curl data out of your network either. Monitoring for subtleties like this can be a huge tip-off of malicious activity in the network.

Stay tuned for more in-depth blogs from other ATD members!

Get started with EmPyre here
