Rekalling a Volatile Past

A brief guide to memory forensics

#whoami

- Graduating from EMU in December
- Self-identified Blue Teamer
- Worked as DNR for Snap Inc
- Contributor to Rekall

Overview of the talk

- What is memory forensics?
- Toolset
- Where to start investigating
- Plugins for days
- Live Demo!

What I will not cover

- How to grab memory
  - http://www.brimorlabsblog.com/2016/12/live-response-collection-bambiraptor.html
  - OSXPmem
  - Snapshot a virtual machine while running, collect the vmem
- How to install the tools
  - Read the documentation :D

What is memory forensics?

- Backstep - What is digital forensics?
- Focuses on volatile data on a machine
- Critical in an investigation
- Can see processes running and network connections
- Provides Indicators of Compromise (IOCs) disk forensics can’t

Toolset

Volatility

- Python
- Works with Windows, Linux, Mac (kind of)

Rekall

- Fork of Volatility
- Focuses on efficiency and robustness

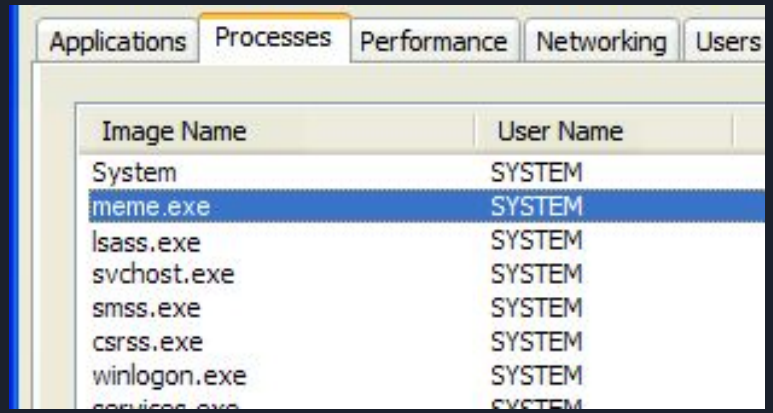
Investigating: Processes

What does a process look like?

How does this help us in Memory Forensics?

Process based plugins:

- pslist
- psscan
- pstree
- psxview



Investigating: Network

How do network connections help us?

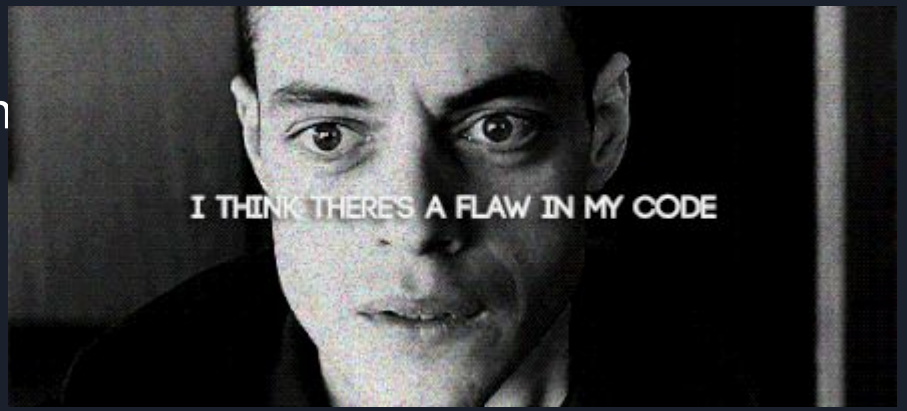
Network based Plugins

- connscan
- connections
- netscan
- sockets

Investigating: Code Injection

Common ways for code to fly “under the radar”

- Remote DLL Injection
- Remote Code Injection
- Hollow Process Injection



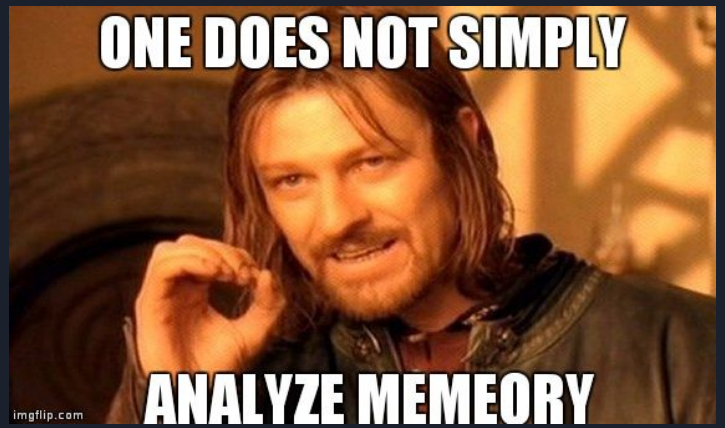
Code Injection plugins

- dlllist
- ldrmodules
- malfind

Demo Time!

Anti-Memory Forensics Techniques

- Create decoys
- Disrupt the collection
  - Remove the kernel debug table



Questions?

Contact me:

- @serenity in Arbsec Slack
- me@jessicawilson.us
- @Sweetserenity28 on twitter