Spying on botnets

Sandor Nemes

FireEye

About Me

- Current: Senior Security Researcher at FireEye iSIGHT Intelligence
- Former: Sophos, GE Capital, National Instruments
- What I do: reverse engineering, threat research, coding
- Certs: OSCP, CISSP

Views expressed here are my own and do not necessarily reflect the opinion of my employer.

Introduction

- Cyber threat maps
  - Can look really cool
  - But very limited practical use
- Data sources
  - AV endpoints
  - Network sensors
  - DNS data
  - C2 sinkholes
  - **Bot emulators**

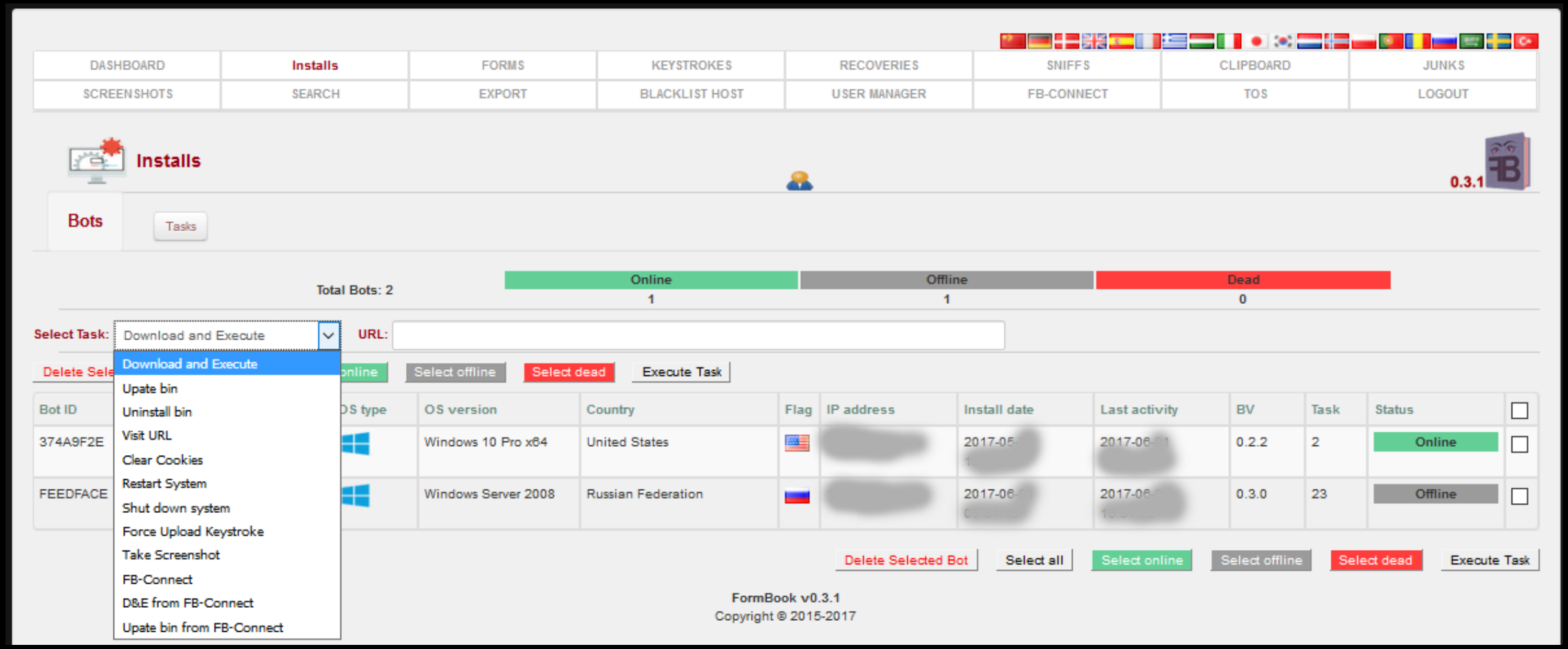
- Other than building awesome looking maps, bot emulators can be used to produce some really useful threat intelligence data for tracking threats and threat actors behind them

Botnets and bots

- Just a quick recap for those who don't do this as their daily job:
  - Private network of computers
  - Infected with malware
  - Controlled as a group
  - Without the consent of the owner
- Botnet architectures:
  - Client-server – uses one or more Command and Control (C2) servers
  - Peer-to-peer (P2P) – clients directly communicate with each other to relay commands
- Malware family examples:
  - Financial malware (Zeus, Emotet, Ursnif/Gozi/ISFB/Dreambot, Trickbot) – forms botnets
  - Ransomware – does not maintain a persistent control channel, thus does not form botnets

Botnets and bots

- Zombie computers of the botnet are controlled by the botnet operator via a web-based panel



How is it possible to extract threat intelligence data from botnets?

- Becoming part of a botnet begins with a malware infection
  - Having a malware sample set or a sample feed is a good start
- Data we are interested in:
  - Information on threat actors
  - Information about targets
- High level steps:

Identifying the malware family

- Looking at some detections on VirusTotal shows this is far from trivial:

- Challenges:
  - Executable packers/crypters
  - Packer/crypter reuse between families
  - Malware downloaders can confuse behavior based detections

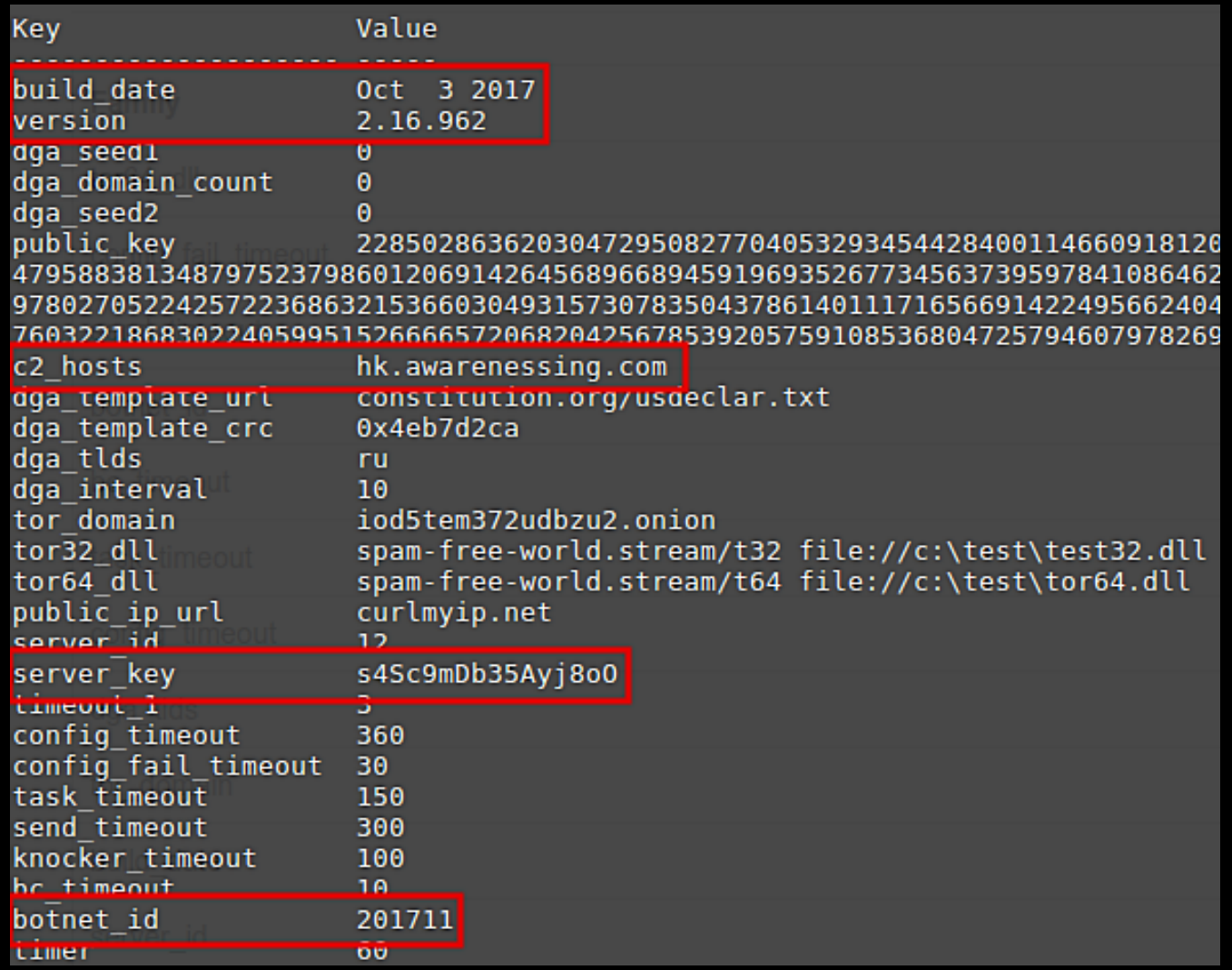
Extracting malware configuration

- Dynamic malware analysis systems / sandboxes come to the rescue
- Extracting approaches:
  - Process memory dumps
    - Results in many smaller memory dumps
    - Mostly event driven: freeing memory, process termination, first network event
    - If the malware sanitizes the memory after using it then it does not work well
  - VM memory dumps
    - Results in one huge memory dump
    - If malware does not execute properly (e.g. crashes) then it does not work
    - Allows for carving the configuration out using memory forensic tools, e.g. Volatility plugins

FireEye

Extracting malware configuration

- Data that we can get:
  - Botnet ID – to track threat actors
  - Version number – to track development
  - Encryption keys – to decrypt traffic
  - C2 servers – indicator of compromise
- Successful extraction also confirms the family
- Extractor scripts break easily with malware family updates
- What else can we do – what can we use the C2 servers for?



Key	Value
build_date	Oct 3 2017
version	2.16.962
dga_seed1	0
dga_domain_count	0
dga_seed2	0
public_key	22850286362030472950827704053293454428400114660918120
	47958838134879752379860120691426456896689459196935267734563739597841086462
	97802705224257223686321536603049315730783504378614011171656691422495662404
	76032218683022405995152666657206820425678539205759108536804725794607978269
c2_hosts	hk.awarenessing.com
dga_template_url	constitution.org/usdeclar.txt
dga_template_crc	0x4eb7d2ca
dga_tlds	ru
dga_interval	10
tor_domain	iod5tem372udbzu2.onion
tor32_dll	spam-free-world.stream/t32 file://c:\test\test32.dll
tor64_dll	spam-free-world.stream/t64 file://c:\test\tor64.dll
public_ip_url	curlmyip.net
server_id	12
server_key	s4Sc9mDb35Ayj8oO
timeout_1	5
config_timeout	360
config_fail_timeout	30
task_timeout	150
send_timeout	300
knocker_timeout	100
bc_timeout	10
botnet_id	201711
timer	60

Emulating bot communication

- Emulate the networking parts of the bot with a script
  - Just to clarify: this is NOT emulation as in “CPU emulation”, just the communication is emulated
  - We pretend to be an infected machine and we become part of the botnet
- Various levels of complexity based on the family and protocol
  - HTTP – just need to figure out the GET/POST parameters and the traffic encryption/decryption
  - For custom binary protocols this is much harder
- Data that can be seen from the server responses
  - Commands
  - Web injects
  - Configuration updates (including C2 server updates)
  - New malware samples

FireEye

Web injects

```
"injects": [
  {
    "set_url_1": [
      "http*://*acc*desjardins.com*"
    ],
    "data_before": "<html*<body*>",
    "inject_flags": 1,
    "data_inject": "<div id=\"_brows.cap\" style=\"position:fixed;top:0px;left:0
  },
  {
    "set_url_1": [
      "https://*bmo.com/onlinebanking/*"
    ],
    "data_before": "if (self == top) {",
    "data_inject": "document.documentElement.style.display = 'block';",
    "inject_flags": 1,
    "data_after": "} </script>"
  },
  {
    "set_url_1": [
```


FireEye

Emulating bot communication

- Challenges:
  - IP addresses / exit nodes
  - Malicious commands
  - Have to provide fake user data
    - System information
    - Task/window list
    - List of installed applications
  - Screenshots
    - Silently skip, and don't answer
    - Send black images of a certain resolution
    - Generate relatively credible screenshots
  - Example: Emotet – does not give you data, until you provide some data
  - Constantly changing families – emulators break silently

```
20:36:50,314 - INFO - main : Starting emulator...
20:36:50,314 - INFO - connect : Trying: 216.126.58.132:443
20:36:51,616 - INFO - proxy_web_request : Proxy web request to 'http://httpbin.org/ip'
20:36:51,620 - INFO - process_command : Request bot info
20:36:51,940 - INFO - process_command : Request loader info
20:36:52,261 - INFO - process_command : Request process info
20:36:52,580 - INFO - process_command : Request machine info
20:36:52,902 - INFO - process_command : Request bot status
20:36:53,226 - INFO - process_command : Request process and session list
20:36:53,640 - INFO - install_plugin : Install plugin 'get_info'
20:36:54,290 - INFO - process_command : Receiving file chunk for file #1...
20:36:54,610 - INFO - process_command : File download complete
20:36:55,176 - INFO - process_command : Plugin command for 'get_info'
20:36:55,504 - INFO - uninstall_plugin : Uninstall plugin 'get_info'
20:37:25,828 - INFO - process_command : Get option 'core_build_time'
20:52:04,064 - INFO - process_command : Save screenshot
21:16:47,463 - INFO - main : Emulator finished successfully.
```

Takeaways

- Information extracted from malware samples can be used to emulate communication
- Emulating the bot communication gives you data that can be used to track threat actors behind these botnets
- This information can be really valuable for banks and financial institutions who are usually the targets of these attacks
- Security teams can use this information to get a better understanding of how these threat groups operate and how they develop their capabilities