Command and Control Mechanism Trends in Exploit Kits, RATs, APTs, and Other Malware

January 13, 2016

Mark Mager
US-CERT Code Analysis Team

Homeland Security

National Cybersecurity and Communications Integration Center

Agenda

- About Me
- 2015 Year in Review
- Malware Crash Course
- Other Malware
- Remote Access Tools
- Exploit Kits
- APTs
- Conclusion

About Me

- Mark Mager
- US-CERT Code Analysis Team
- phia LLC
- Reverse Engineer / Software Engineer
- Away from daily malware analysis for 4 years
  - Returned in 2015
    - What's changed?

2015 Year in Review

- Hacking Team
- OPM
- Hacktivism
  - Anonymous
  - Terrorist cells
- Cybercrime
  - Carbanak
  - Premera Blue Cross
- See hackmageddon Cyber Attacks Timeline…

Malware Crash Course

- Malware Roles
  - Contained within one or more files
    - Initial Attack Vector
    - Launcher
    - Dropper
    - Downloader
    - Command and Control
    - Advanced Malicious Capabilities
      - Keylogging, process enumeration, reverse shell
    - Persistence

Malware Crash Course

- Malware Categories
  - Remote Access Tools
  - Exploit Kits
  - Advanced Persistent Threats
  - Other Malware…

Malware Crash Course

- Analysis Techniques
  - Dynamic Analysis
    - Virtualization
    - Debugging
    - Unpacking
  - Static Analysis
    - Disassemby / decompilation
    - Deobfuscation
  - Live Dynamic Analysis
    - Non-attribution
    - Payload retrieval

Caveats

- Derived from own research, analysis
  - Curated dataset
- Primarily Windows samples
- Generalized info
  - Limited indicators
  - OPSEC
- Timeline: February 2015 – November 2015

Low-Hanging Fruit: Office Documents

- Decoy document text
  - Enable macros, please?
- Obfuscated macros
  - Commercial obfuscation
    - CrunchCode
- Multi-stage
  - VBScript, batch files, PowerShell
- Payloads directly downloaded and executed
  - Spray and pray spear phishing
  - No attempt at obscuring comms
    - GET stage2.exe HTTP/1.1

Low-Hanging Fruit: Office Documents

- Decoy document text
  - Enable macros, please?
- Obfuscated macros
  - Commercial obfuscation
    - CrunchCode
- Multi-stage
  - VBScript, batch files, PowerShell
- Payloads directly downloaded and executed
  - Spray and pray spear phishing
  - No attempt at obscuring comms
    - GET stage2.exe HTTP/1.1

Low-Hanging Fruit: Office Documents

- Callback URIs
  - Compromised sites
    - Wordpress blogs
      - *.*/wp-content/uploads/*

Remote Access Tools

- Gh0st RAT
  - Connects to C2 URI over TCP port 80
    - "HTTP\1.1 Sycmentec" header
      - 48 54 54 50 5c 31 2e 31 20 53 79 63 6d 65 6e 74 65 63 d3
    - variable length of null bytes
    - data pertaining to compressed / decompressed size of payload
    - zlib default compression header: 78 9c [4b]
    - encrypted (using Gh0st RAT's custom encryption routine) payload consisting of system information (e.g. operating system version, computer name, username) which has been compressed with zlib
  - No request method specified

Remote Access Tools

- PlugX
  - Connects to C2 over TCP port 80 then initiates a HTTP POST request
    - POST /update?id=00188d08 HTTP/1.1
      - Accept: */*
      - OldServer: 0
      - Check: 0
      - PostSize: 61456
      - PostSerial: 1
      - User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; InfoPath.2; SV1)
      - Host: xxx.xxx.xxx.xxx
      - Content-Length: 0
      - Cache-Control: no-cache

Remote Access Tools

- exeproxy
  - Ciphers 80 byte block of data
    - randomly-generated data and host's NetBIOS name
    - block is XORed using an embedded shifting XOR cipher
  - Establishes a secure session (using the embedded OpenSSL library) over TCP port 443 with C2 URI and sends the the data block
  - XOR decrypts response data from the C2 server with key that is included in the response
  - Validates the decrypted data and determines whether to continue its established session with the C2 or terminate the session

Remote Access Tools

- exeproxy
  - Anti-reversing techniques
    - Code blocks have no effect
    - Thwart reverse engineering efforts and obfuscate capabilities
    - Implemented via inline assembly
      - x86 instructions (e.g. pushf, popf) not often seen in compiled code
      - meaningless instructions (e.g. or ax, ax)
    - Conditional statements which always yield same value
    - Explicit preservation of registers via the stack
      - e.g. code blocks begin with several push instructions in a row and pushf and then end with the equivalent popf / pop instructions

Exploit Kits

- Modular
- Easy to use GUI
- Packed with exploits
- Amorphous, multi-stage payloads

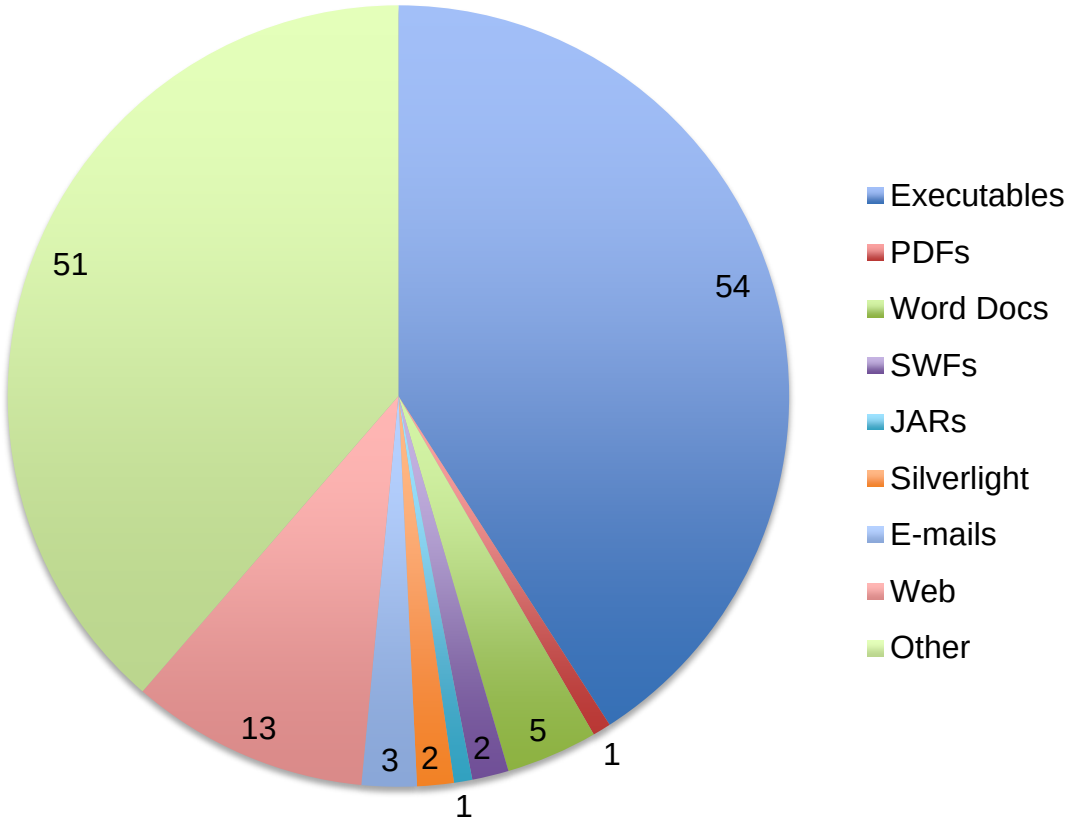
Angler Exploit Kit

- Multi-stage
- Multiple potential vectors
  - Java, Flash, Silverlight
  - Silverlight is not as easily reversed
    - Heavily obfuscated
      - Unprintable Unicode characters
      - Obscured control flow
        - » Nested, indirect function calls
      - Functionality spread across several classes

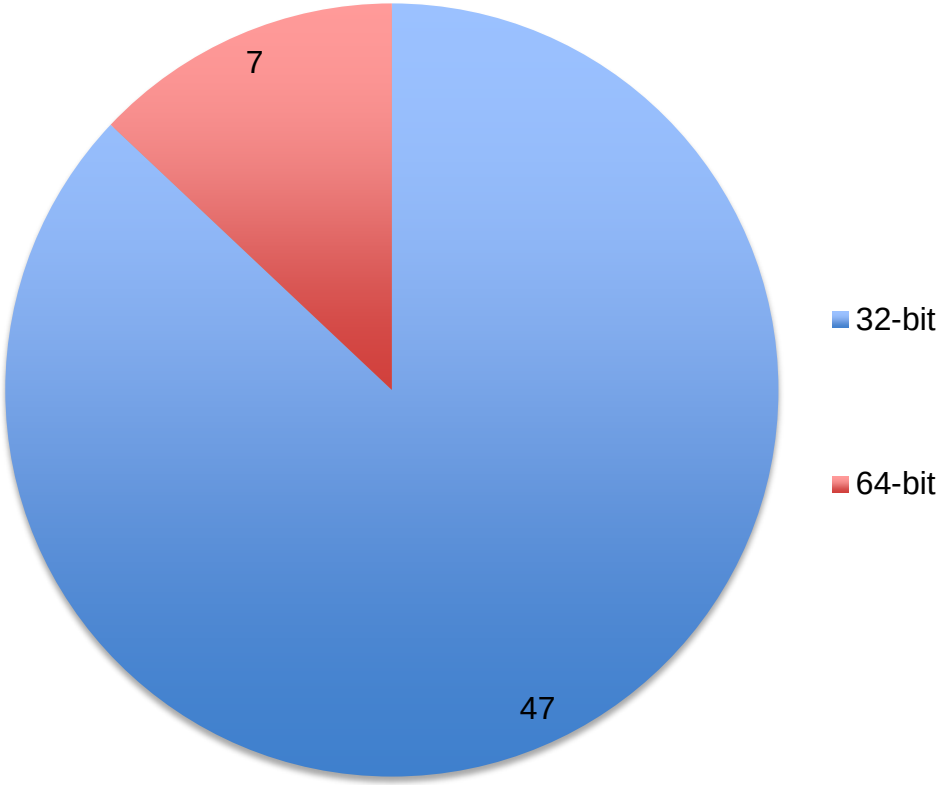
Advanced Persistent Threats

- Zero Days
  - Hacking Team SWF exploits
    - CVE-2015-5119
  - Quick turnaround
- Spear phishing still extremely common
  - Easiest initial exploit vector
  - HTTP GET requests to compromised sites for payloads
- Gh0st RAT Variant used in APT activity

Total Artifacts Analyzed



Executables Analyzed



Conclusion

- Advanced C2 techniques not always used
- Older malware and TTPs are still used
  - pwdump, ophcrack, Hacker's Door
  - RATs analyzed trace back several years
  - spray and pray is still very common
- Complex multi-stage / modular frameworks
- Turnaround for zero days is shortening
- Lack of 64-bit malware, rootkits

Thanks

- US-CERT Code Analysis Team
- Northrop Grumman
  - Rob Mangiante
- phia LLC
  - Chad Hein
- Rodney DeCarteret
- Tessa Strasser