OpenIOC

PRESENTED BY **Sean Gillespie** | August 19, 2014

Terminology

- Tactics, Techniques, and Procedures (TTPs)
- Intelligence – Information about threat actor tools and TTPs
  - Tool usage
  - Domains
  - File attributes
- Evidence – The data in your collection systems
- Indicator of Compromise (IOC) – matching evidence to intelligence
- OpenIOC – An XML format for storing Indicators of Compromise

Overview

- Low Value Use Cases
  - Storage and Transfer of Intelligence
  - List Matching Based Detection
- High Value Use Cases
  - Rapid Scoping of Compromises
  - Sharing and Deployment of Operational TTPs
- Required Tools and Concepts
- The Power of OpenIOC
- OpenIOC and You
- PyIOCe
- Anatomy of OpenIOC
- Q&A

Storage and Transfer of Intelligence – Low Value

MD5	63d0…1bd7
Filename	winavg.dll
Size	38468
Compile Time	01-06-2011T15:35:32Z
Exports	ServiceMain
Sections	.text .data .idata .rsrc
Strings	Failed to open sockit

MD5	a95c...d150
Filename	winssh.dll
Size	38468
Compile Time	01-06-2011T15:35:32Z
Exports	ServiceMain
Sections	.text .data .idata .rsrc
Strings	Failed to open sockit

MD5	323b...16f1
Filename	winhlp.dll
Size	38468
Compile Time	01-06-2011T15:35:32Z
Exports	ServiceMain
Sections	.text .data .idata .rsrc
Strings	Failed to open sockit

MD5	831f…22ed
Filename	winzip.dll
Size	54862
Compile Time	05-24-2012T17:28:19Z
Exports	ServiceMain
Sections	.text .data .idata .rsrc

MD5	051b...f1ba
Filename	winrip.dll
Size	24218
Compile Time	07-14-2009T09:12:47Z
Exports	ServiceMain
Sections	.text .data .pinfo .rdata

YAHOO!

List Matching Based Detection – Low Value

- List maintenance
  - IOC quality
  - IOC ages
  - IOC duplication
- Results are difficult to correlate to original intelligence
- Wildly Inefficient
  - High false positive rate if IOCs aren't extremely specific
  - High false negative rate if IOCs aren't carefully crafted to detect variations
  - High true negative rate since most of what you are searching across does not match

Rapid Scoping of Compromises – High Value

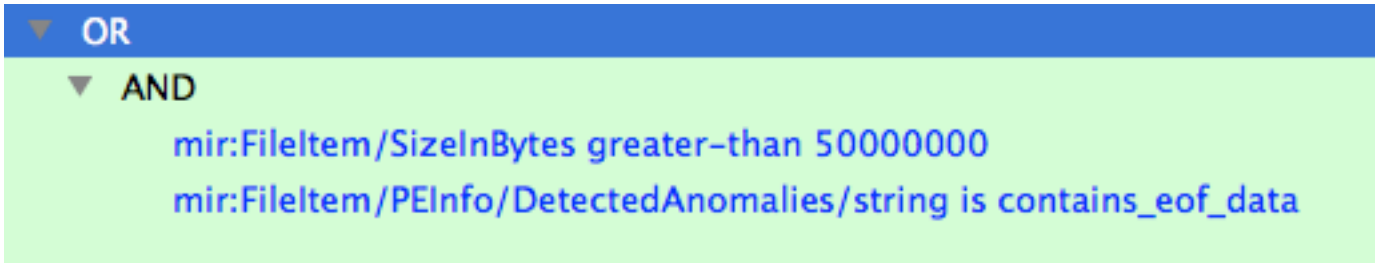
- Rapidly target specific aspects of a known compromise
  - Search for all executable files written to disk by ‘Bob’ in the last 30 days
  - Search for all instances of setup64.exe
  - Search for all registry entries containing \Oracle\
  - Search for MUICache/Prefetch evidence of setup.exe execution
- Large result sets mitigated by Incident Responders having contextual knowledge of the current threat
- Iterative process of refining IOCs and repeating searches

Sharing and Deployment of Operational TTPs – High Value

- Operational IOCs are meant to describe forensically interesting sources of data or unique behaviors and attributes of malicious activity
  - Persistence mechanisms
  - Suspicious file attributes
  - Suspicious process attributes
  - Execution history locations
  - Recently opened documents locations
  - Browser history locations
- Used for actively hunting interesting anomalies and rapidly processing forensic data
- Serve as quick and functional references for training and consistency

Examples

- Common Anti-Virus Evasion Technique



- Common Lateral Movement Technique

YAHOO!

Examples

- Common Default Hash Dumping Export

```
OR
    mir:FileItem/PEInfo/Exports/ExportedFunctions/string is Gethash
```

- Common Malware Misspelling

```
OR
  AND
      mir:ProcessItem/name matches ^s[vchosu0]t\.exe
      NOT mir:ProcessItem/path ends-with system32
      NOT mir:ProcessItem/name is svchost.exe
```

- Common Persistence Mechanism / Authentication Bypass Technique

```
OR
  AND
      mir:RegistryItem/ValueName contains Debugger
      mir:RegistryItem/Path contains Image File Execution Options
      NOT mir:RegistryItem/Text contains ntsd
      NOT mir:RegistryItem/Text contains windbg
```


YAHOO!

Required Tools and Concepts

- Editors – Tools to create and edit OpenIOC files
  - PyIOCe
  - Mandiant IOCe
- Operational Systems – Systems that gather data
  - Splunk, Snort, GRR, MIR, Volatility, Yara
- Parsers – Tools to turn OpenIOC files into operational inputs
  - MIR -> XPATH
  - GRR -> Flow inputs
  - Splunk -> Search
  - Snort -> Rules
  - Yara -> Sigs

The Power of OpenIOC

- OpenIOC allows for simple intuitive descriptions of complex patterns
  - Indicator Logic – AND, OR
  - Term Conditions – is, contains, matches, starts-with, ends-with, greater-than, less-than
  - Term Modifiers– negate, case-sensitive
- OpenIOC can be used for pivoting from known intelligence items
  - File info
    - File as a process
    - Registry values containing the filename
    - Execution history containing the filename (Prefetch/MUICache/Bit9)
- Sharing TTP based OpenIOC files
  - Does not reveal confidential information
  - Does not aid attackers

The Power of OpenIOC

- OpenIOCs are meant to be parsed to create inputs for operational systems
- OpenIOC is best as a method for exchanging TTPs not intelligence
  - How to search for X with System Y not just what is X
  - There are better formats and methods for exchanging intelligence data
- Global terms vs Operational System specific terms
  - Translation is complex and makes it easy to lose intended effects
  - Systems may have extreme variations in how they interpret and use terms

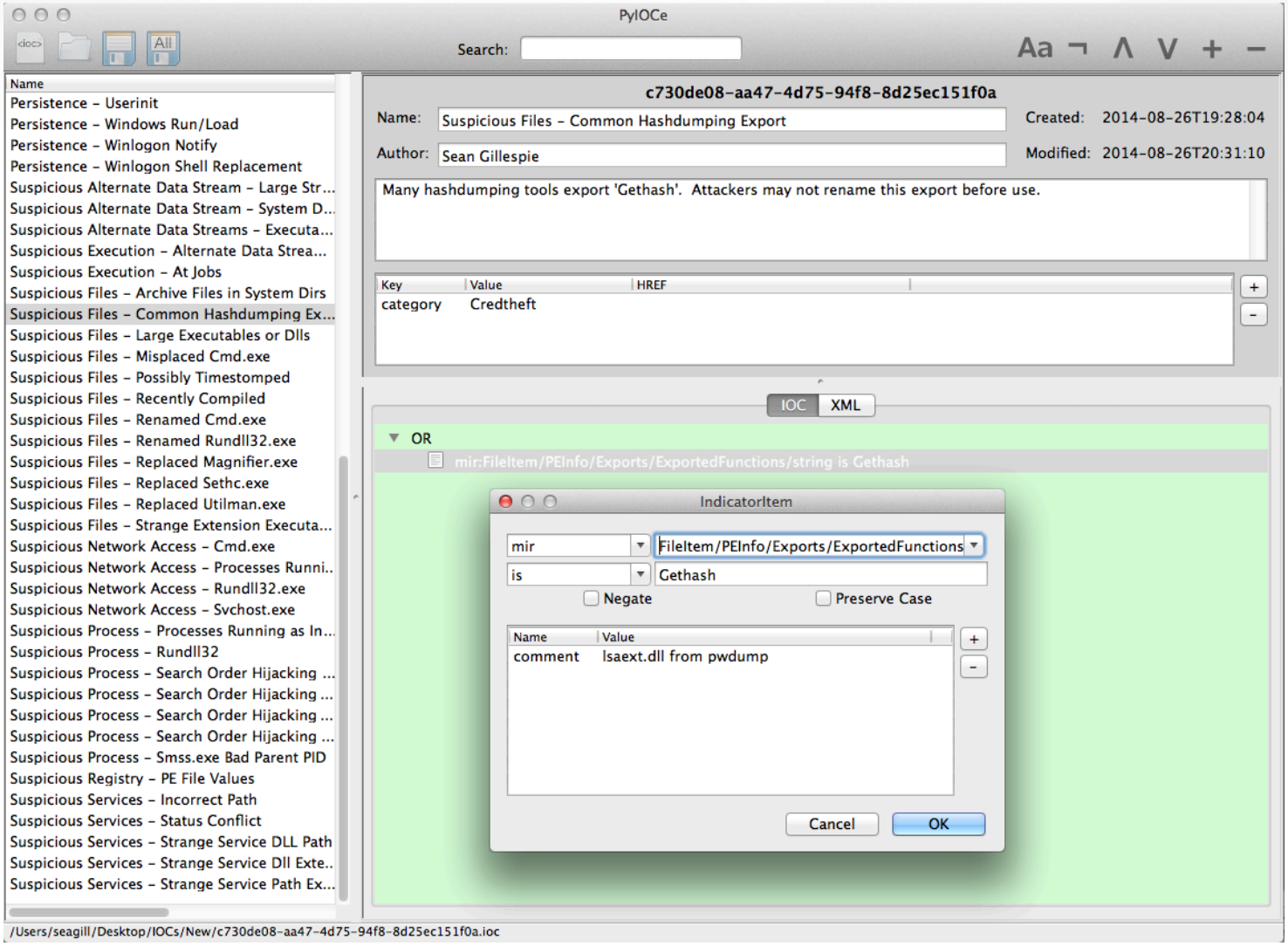
OpenIOC and You

- Define your own terms
  - Terms can be used to describe anything an operational system is aware of
    - Specific data points
    - Flagged anomalies
  - Create term names that make sense for your operational system
- Define your own parameters
  - Parameters can modify criteria or describe actions for operational systems
  - Create parameters that best reflect the capabilities of your operational system
- Build your own parsers
  - It is up to you to decide what your system can do by combining custom terms and parameters

PyIOCe Key Features

- Cross platform
- Keyboard driven
- Capable of working with OpenIOC 1.0 as well as 1.1
- IOC cloning for rapid duplication for testing
- Edit, import, and export of IndicatorTerm and Parameter lists to extend OpenIOCs for use with other operational systems
- Term Maps to group terms with roughly equivalent content to assist in translating for other operational systems

Anatomy of OpenIOC



YAHOO!

Anatomy of OpenIOC

```
<OpenIOC xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://openioc.org/schemas/OpenIOC_1.1" id="c730de08-aa47-4d75-94f8-8d25ec151f0a" last-
modified="2014-08-26T19:49:18" published-date="0001-01-01T00:00:00">
 <metadata>
   <short_description>Suspicious Files - Common Hashdumping Export</short_description>
   <description>Many hashdumping tools export 'Gethash'.  Attackers may not rename this export before use.
   </description>
   <authored_by>Sean Gillespie</authored_by>
   <authored_date>2014-08-26T19:28:04</authored_date>
   <links>
       <link rel="category">Credtheft</link>
   </links>
 </metadata>
```


YAHOO!

Anatomy of OpenIOC

```
<criteria>
  <Indicator id="9217c1ab-d389-454e-bf27-08ca1621c406" operator="OR">
    <IndicatorItem preserve-case="false" negate="false" id="02b53389-3c51-4651-9449-962a468cb6b8" condition="is">
      <Context document="FileItem" search="FileItem/PEInfo/Exports/ExportedFunctions/string" type="mir"/>
      <Content type="string">Gethash</Content>
    </IndicatorItem>
  </Indicator>
</criteria>
```

Anatomy of OpenIOC

```
 <parameters>
   <param id="5315f26b-4b28-4472-bb36-a398e2bbe6c2" ref-id="02b53389-3c51-4651-9449-962a468cb6b8"
     name="comment">
    <value type="string">pwdump</value>
   </param>
  </parameters>
</OpenIOC>
```


YAHOO!

Overview

- Low Value Use Cases
  - Storage and Transfer of Intelligence
  - List Matching Based Detection
- High Value Use Cases
  - Rapid Scoping of Compromises
  - Sharing and Deployment of Operational TTPs
- Required Tools and Concepts
- The Power of OpenIOC
- OpenIOC and You
- PyIOCe
- Anatomy of OpenIOC
- Q&A

Q&A