CONCURRENT BEHAVIOUR ANALYSIS: RESILIENT INDICATORS OF EMERGENT EXPLOITS

Dennis R. Moreau, Ph.D.
RSA / Office of the CTO
Static Indicators Grow Stale Quickly

MW behavior changes faster than indicators propagate
MW Exhibit Many Detectable Behaviors …

But we can’t know which behaviors a priori … so, we should watch lots of different sensors
and nervous MW behaves differently

You can’t fool ALL of the sensors, All of the time … so, we should watch lots of different of sensors

and MW evolves … rapidly

... so, we watch for degree of inconsistency, anomaly, outliers, change ... rather than scoring on static patterns
<table>
<thead>
<tr>
<th>Line</th>
<th>Instruction</th>
<th>Source Code</th>
<th>Additional Details</th>
</tr>
</thead>
</table>
| 69   | invoke-static {v2, v3, v4}, Lcom/android/system/admin/xfclicl.c::>cOlClOoo(III)Ljava/lang/String; | | - Time: 170899  
  - param0: openConnection  
  - param1: null  
  - Return:  
    - openConnection  
    - public java.net.URLConnection java.net.URL.openConnection() throws java.io.IOException |
| 73   | invoke-virtual {v0, v2, v3}, Ljava/lang/Class::>getMethod(Ljava/lang/String; [Ljava/lang/Class;,)Ljava/lang/reflect/Method; | | - Reflective invoke: java.net.URL.openConnection  
  - Return:  
  - Time: 170918  
  - param0: http://www.androfox.com/load.php  
  - param1: null  
  - Return:  
| 78   | move-result-object p0 | | |
| 79   | try_end_71: const/16 v0, 0xb | | |
| 80   | const/16 v1, -0x90 | | |
| 81   | const/16 v3, 0x32 | | |
## OBAD.A Endpoint Behavior

### Strings
- eCzyf2UidGhilw==
- su -c 'id'
- read

<table>
<thead>
<tr>
<th>Position</th>
<th>Instruction</th>
<th>Meta Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>try_start_0.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>invoke-static {}.getRuntime().getRuntime();</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>move-result-object v0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>const/16 v1, 0x12</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>const/16 v2, 0xa</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>const/16 v3, -0x1c</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>invoke-static {v1, v2, v3}, Lcom/android/system/admin/OcoolClc;-&gt;cOlcOo(III)java/lang/String;</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>move-result-object v1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>invoke-static {v1}, Lcom/android/system/admin/ocOlClCo;-&gt;ooOolC(Ljava/lang/String;):java/lang/String;</td>
<td>- Time: 144500&lt;br&gt;- param0: [B@a06aa5f0&lt;br&gt;- param0: su -c 'id'&lt;br&gt;- param0: 7375202D632027696427&lt;br&gt;- Return:&lt;br&gt;- su -c 'id'&lt;br&gt;- Time: 144500&lt;br&gt;- param0: eCzyf2UidGhilw==&lt;br&gt;- Return:&lt;br&gt;- su -c 'id'&lt;br&gt;- Time: 144551&lt;br&gt;- param0: su -c 'id'&lt;br&gt;- Return:&lt;br&gt;- Process[pid=2369]</td>
</tr>
<tr>
<td>11</td>
<td>move-result-object v1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>invoke-virtual {v0, v1}, Ljava/lang/Process;-&gt;exec(Ljava/lang/String;):Ljava/lang/Process;</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>move-result-object v6</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>invoke-virtual {v6}, Ljava/lang/Process;-&gt;getInputStream(Ljava/io/InputStream;</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>move-result-object v7</td>
<td></td>
</tr>
</tbody>
</table>
Obad.A: C2 Behaviors

► Send text messages. Parameters contain number and text. Replies are deleted.
► PING.
► Receive account balance via USSD.
► Act as proxy (send specified data to specified address, and communicate the response).
► Connect to specified address (clicker).
► Download a file from the server and install it.
► Send a list of applications installed on the smartphone to the server.
► Send information about an installed application specified by the C&C server.
► Send the user’s contact data to the server.
► Remote Shell. Executes commands in the console, as specified by the cybercriminal.
► Send a file to all detected Bluetooth devices.
**BLYPT**

- iexplore.exe
- jp2launcher.exe
- java.exe

**JAVA_EXPLOYT.H**

- 300e7c35-2ca05d71

**<Compromised Site>**

- Download
- Execute
- Inject

**EXPLORER.EXE**
Behavior Analysis
Network Behavior: Anomalous Web Interaction (SilverTail)
Network Behavior - Characterize

- Characterize
  - Sequence
  - Graph Measures
  - Frequency
  - Dynamics
  - ...

Example

Anonymized use-case from a customer

This user did A-B-C-A-B-C-A-B-C-A-B-C-A-B-C-A.

<table>
<thead>
<tr>
<th>Expected</th>
<th>Refrequency</th>
<th>Frequency</th>
<th>Anomaly</th>
</tr>
</thead>
<tbody>
<tr>
<td>A -&gt; B Transition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60.2791</td>
<td>9.81287</td>
<td>1.59744</td>
<td>0.260049</td>
</tr>
<tr>
<td>62</td>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5.9662</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| B -> C Transition |
| 960.428   | 72.6302     | 5.4925    | 0.415358 |
| 962       | 71          | 5         | 1       |
| 1        | 2           | 3         | 4       |
| 6.50075  |

| C -> A Transition |
| 2504.13   | 859.581     | 295.065   | 101.286  |
| 2529      | 830         | 287       | 102      |
| 1        | 2           | 3         | 4       |
| 0.676293 |
EP Behavior: Injection (ECAT)

OS Loading
*Address FixUp, Link Resolution*

A
- .dll on Disk

Simulated Loading

B
- .dll in RAM

B'
- anticipated .dll image

ECAT test: B ≠ B’ triggers alert
EP Behavior & Inconsistency:

**Behavior**

- Hooking and Consequent Action
- Privilege Escalation
- Log Sequences
- Resource Consumption
- ...

**Anomaly**

- Disk vs. RAM
- Threads vs. Processes
- Registry API vs. Hive Structure
- FS vs. Shadow FS
- Currency of Updated Lists
- ...

Ref: http://maec.mitre.org/about/docs/MAEC_Conficker_Issues_Challenges.pdf
Example: <Signal to Noise> ++

- Single Sensor - Baselining Xtime, Xpopulation, Xtrans
  - Change over Time from User “normal”
  - Difference from Population “normal”
  - Difference from Session type “normal”
  - Difference for an accession “device”
  - Clustering – RT vs. Batch
  - Sensor signal mining across parameter spaces
  - …
Behavior Sensor Examples

► Web Session Navigation Patterns - Silvertail
► Netflow and DNS Lookup Patterns – LosAlamos PathScan
► Traffic and flow patterns – NetWitness Parsers + Meta + Sandbox Based Behaviors
► Endpoint Anomalous State and Behavior – Ex. Injection + Network Activity - ECAT
► Kernel Hooking - AutoVAC – Texas A&M
► Hybrid Static/Dynamic + Recipe Driven SandBox Analysis – Joe Security, Joe Sandbox
Concurrent Analysis
Concurrent Behavior Analysis

► Behavior: Sensed Change ▽ Time vs. Matched String
► Coherence: Inconsistency vs. Signature
► Dimension: Multiple Aspects vs. Single Aspect Alerts
► Composition: Anomalies vs. Focused Indicator Scoring Threshold
Opportunity: Multi-modal Behavior
Ex. NW Behavior + EP Behavior

Composed Higher Confidence Behavioral Indicator

Orchestrate linkage on Asset ID : IP, MAC, UUID, Hosting Stack, ...

!(EP) ⇒ !(NW)
Examples: <Signal to Noise> ++

**Behavior Outlier**
- http get
- Count Per Source
- Seasonality Filter
- Hodrick-Prescott

**Behavior Sequence**
- DNS
- Lateral Event
- Recon Event
- Whitelist

**Concurrent Behavior**
- VPN Device
- Ground Speed Event
- New Destination Event
- Score Filter
- 5000
- 200
- 100
- Intersect
- Rank Filter
- 1

**State**
Complex Emergent Behavior: Labeling & Propagation

► Track User, Asset, Transaction profiles – label suspicious entities based on modeled constraints
► Taint subsequent interactions – suspicion propagation.
► Reputation as a operational signal
  ► Action-ability: Score indicates where to look first.
  ► Action-ability: Adjust sensors thresholds and possibly, Shields!
  ► Action-ability: Confirm mitigation/remediation.
Ex. Complex Emergent Behavior Characterization

- Clustering Dimensions for Graphs
  - Describing Behavior Graphs (comparison)
    - Frequency Domain
    - Complexity Measures and Metrics
    - Shape Measures
    - …
  - Association across aspects on common entities
    - EP Identifiers/Account/Service/Process …
    - User Identifiers
    - Session Identifiers
    - Application/Service/Site Identifiers
    - …
Ex. Complex Emergent Behavior Recognition

► Can leverage sensor specific mining with no change

► As We Approach Mining Across Sensors – New Issues
  ► Bayesian, SVM, Neural Networks, Decision Trees, …
  ► But need to understand base data and model statistics
    ► Signal and Variable Dependence
    ► Underlying Distribution
    ► Data and Model Interactions
<maecPackage:Analyses>
<maecPackage:Analysis id="maec-example-ana-1" method="dynamic" type="triage">
<maecPackage:Summary>Dynamic (behavioral) triage.</maecPackage:Summary>
<maecPackage:Findings_Bundle_Reference bundle_idref="maec-example-bnd-1"/>
<maecPackage:Tools> <maecPackage:Tool id="analysis-tool-1">
    <cyboxCommon:Name>Anubis</cyboxCommon:Name>
    <cyboxCommon:Vendor>IsecLab</cyboxCommon:Vendor>
</maecPackage:Tool></maecPackage:Tools>
</maecPackage:Analysis>
<maecPackage:Analysis id="maec-example-ana-2" method="dynamic" type="triage">
<maecPackage:Summary>Dynamic (behavioral) triage.</maecPackage:Summary>
<maecPackage:Findings_Bundle_Reference bundle_idref="maec-example-bnd-2"/>
<maecPackage:Tools> <maecPackage:Tool id="analysis-tool-2">
    <cyboxCommon:Name>ThreatExpert</cyboxCommon:Name>
    <cyboxCommon:Vendor>Symantec</cyboxCommon:Vendor>
</maecPackage:Tool></maecPackage:Tools>
</maecPackage:Analysis>
</maecPackage:Analyses>
Multi-Behavior MAEC 4.0

<maecPackage:Findings_Bundles>...

<cybox:Associated_Objects> <cybox:Associated_Object id="maec-anubis_to_maec-obj-1">
    <cybox:Properties xsi:type="FileObj:FileObjectType">
        <FileObj:File_Name>oembios.exe</FileObj:File_Name>
        <FileObj:File_Path>C:\WINDOWS\system32\</FileObj:File_Path>
    </cybox:Properties>
    <cybox:Name xsi:type="maecVocabs:SynchronizationActionNameVocab-1.0">create mutex</cybox:Name>
</cybox:Associated_Object>

<cybox:Associated_Objects> <cybox:Associated_Object id="maec-anubis_to_maec-obj-2">
    <cybox:Properties xsi:type="WinMutexObj:WindowsMutexObjectType">
        <MutexObj:Name>__SYSTEM__91C38905__</MutexObj:Name>
    </cybox:Properties>
</cybox:Associated_Object>

<cybox:Associated_Objects> <cybox:Associated_Object id="maec-anubis_to_maec-obj-3">
    <cybox:Properties xsi:type="WinRegistryKeyObj:WindowsRegistryKeyObjectType">
        <WinRegistryKeyObj:Key>software\microsoft\windows nt\currentversion\winlogon</WinRegistryKeyObj:Key>
        <WinRegistryKeyObj:Hive>HKEY_LOCAL_MACHINE</WinRegistryKeyObj:Hive>
        <WinRegistryKeyObj:Values>
            <WinRegistryKeyObj:Value>
                <WinRegistryKeyObj:Name>userinit</WinRegistryKeyObj:Name>
                <WinRegistryKeyObj:Data>C:\WINDOWS\system32\userinit.exe,
                C:\WINDOWS\system32\oembios.exe</WinRegistryKeyObj:Data>
            </WinRegistryKeyObj:Value>
        </WinRegistryKeyObj:Values>
    </cybox:Properties>
</cybox:Associated_Object>

</maecPackage:Bundle>
Multi-Indicators in MAEC 4.0

- MAEC Bundle
  - Beaconing Behavior
    - URL
    - IP Addr.
  - Residency Behavior
    - File
    - Reg. Key

- Manual Dyn/Static Analysis
- Automatic Dyn/Static Analysis

Outputs:
- SNORT
- OVAL
Open Questions for Concurrent Behavioral Analysis
Open: Multiple Time Bases

- Smart sensors can introduce non-deterministic delays
  - Lack of Clock Synchronization
  - Buffering prior to recognition … and alert
  - Sandbox execution (usually has a timeout)
  - Batch analytics can be long running
  - Non-linear analytic approaches (UTM attack permutations over vulnerability graphs)
- …
- Need correlation that is less sensitive to clock de-sync
- Need to reason using partial ordering or relaxation of dependencies – transformation (ex. f-domain)
- Need interval logics for reasoning about time
Open: Policy/Filter Conflict …

Sensors can be filters … and affect visibility

- Policy distributed across different interacting PDPs
- Policy expressed in many different policy languages and logics: Snort, YARA, Parsers ..
- Potential policy interactions: What does FW + IPS +WF mean? Same as WF + IPS + FW?
- Some devices fail open, some fail closed under load

Beyond UTM
Open: Automated Reasoning About Complex Emergent Behavior

- Belief & Plausibility Evolution Across Sensors
  - Dempster Schaffer revisited – Fusion of belief constraints from different sources
    - Explicit provision from negative (conflicting) evidence
    - Posture + Behavior + Reputation + Multi-Anomaly
- Transferable Belief Models
- Theory of Hints
- ...

RSA CONFERENCE EUROPE 2013
Immediate Application

► Improve Signal to Noise Ratio on Anomaly Alerts
  ► Score = (S1 + S2 + … Sn)/n  \( Si := \) degree of anomaly
  ► Very simplistic combination: \( 0 \leq Si \leq 1 \), 0 so, \( 0 \leq \text{Score} \leq 1 \)
  ► Can add decay for aging of sequence of anomaly scores
  ► Can add variance from average for sequence anomaly

► Additional Aspect
  ► Use anomalous traffic alert to trigger endpoint consistency scan (dll, reg, filesystem, threads, hooks…)
  ► Use anomalous endpoint score to trigger packet capture and deeper traffic analysis
Summary

► Emergent malware is increasingly dynamic and increasingly coopts legitimate and operationally essential (hard to block) characteristics (legitimate IPs, Services, APPs, Protocols, …)

► Reasoning across massive volumes of otherwise legitimate static indicators is probably not the answer

► Multi-aspect concurrent behavior analysis allows us to automatically improve anomaly indications

► Multi-aspect concurrent behavior analysis allows us to establish a more comprehensive, and therefore actionable context
Thank you!

Dennis R. Moreau, Ph.D.
RSA / Office of the CTO

dennis.moreau@rsa.com
www.rsa.com
Appendix
Ex. Complex Emergent Behavior Clustering

- Mean shift clustering
- Principal Component Analysis (PCA)
- Bilateral filtering
- Expectation Maximization – Shape Sensitive
- K-Means – Similar Extents
Ex. Streaming Clustering

► Classification of Data Streams – Micro Clustering
► Ensemble Classification
► On Demand Classification
► Lite Weight Classification – adaptive granularity fluctuating data rates
► ANNCAD - adaptive nearest neighbor clustering for data streams
► SCALLOP - scalable clustering on decision patterns – continuous rule updates
Ex. Streaming Clustering

► Velocity Density Methods
  ▶ Dimensional evolution – drifting normal to avoid false positives

► Stream Cubes
  ▶ High dimensionality distance measures under adaptive means

► Realtime synopsis (descriptions)
  ▶ Stability under load and change in real time p-stable
  ▶ Equi-depth multi dimensional histograms – distance is a centroid distance

► Streaming k-means
Ex. Streaming Analytic Technique

**Adaptation of Clustering Techniques**

- **Continuous K-Means**
- Good for identifying clusters as they emerge and merge
  - Tracked over time to support cluster change analysis
  - Not precise enough to drive splits, so:

```
    Stream-ish over data
```

```
Clustering over abstracts of data: On Demand
```
Streaming vs. Batch Analytics

Real-time – protection plausible