Cyber Observable eXpression (CybOX) Foundations

http://cybox.mitre.org

Sean Barnum

July 2012

Homeland Security

MITRE

Cyber Security Requires a Diverse Set of Activities

- **Secure Operations**
  - Understand Attacker Behavior
  - Detect & Prevent Attacks
  - Investigate & Address Attacks
  - High Fidelity Investigation
  - Find and Understand Malware
  - How do we know what has happened?
  - How do we know what is happening?
  - How do we feed risk decisions with situational awareness?
  - How do we share threat info?
  - What do attack patterns look like?
  - What does it look like?
  - What does it look like?
  - What does it look like?
  - What does it look like?
  - What does it look like?
  - What does it look like?
  - What does it look like?
  - What does it look like?
  - What does it look like?

Cyber Observables are focused at answering this question

- **Secure Development**
  - How do we build security in?
  - How do we make sure security works?
  - What does it look like?
  - What does it look like?

- **What is a cyber observable?**
  - **a *measurable event* or *stateful property* in the cyber domain**
    - **Some measurable events: a registry key is created, a file is deleted, an http GET is received, …**
    - **Some stateful properties: MD5 hash of a file, value of a registry key, existence of a mutex, …**

- **Cyber Observable eXpression (CybOX) is a structured language for encoding and communicating information about cyber observables (http://cybox.mitre.org)**

Cyber Observables Apply to Numerous Domains

- Threat assessment & characterization (detailed attack patterns)
- Malware characterization
- Operational event management
- Logging
- Cyber situational awareness
- Incident Response
- Digital Forensics
- Cyber Threat information sharing
- Etc.

CybOX Principles

- **Targeted Gestalt**
  - **CybOX is not targeted at a single cyber security use case but rather is intended to be flexible enough to offer a *common solution* for all cyber security use cases requiring the ability to deal with cyber observables. It is primarily intended as a piece of information infrastructure that domain-specific standards and solutions can build upon.**
- **Flexibility to express both instances and apriori potential patterns**
  - **It is also intended to be flexible enough to allow both the high-fidelity description of instances of cyber observables that have been measured in an operational context as well as more abstract patterns for potential observables that may be targets for observation and analysis apriori.**
- **Integrated automation vision**
  - **By specifying a common structured schematic mechanism for these cyber observables, the intent is to enable the potential for detailed automatable sharing, mapping, detection and analysis heuristics.**

A Brief History of Cyber Observables

- **September 2009: Originally introduced as part of CAPEC adornment to the structured Attack Execution Flow**
- **Summer 2010: CAPEC, MAEC & CEE teams collaborate to define one common structure to serve the common needs**
- **December 2010: Discussions with Mandiant for collaboration and alignment between Cyber Observables and Mandiant OpenIOC**
- **December 2010: Cyber Observables schema draft v0.4 completed**
- **May 2011: Spun off as independent effort called the Cyber Observable eXpression (CybOX)**
- **Oct 2011: CybOX website launched with initial schema Version 0.6**
- **Jan 2012: CybOX Version 0.7 released**
- **Mar-Apr 2012: CybOX Version 1.0(draft) released with formal specifications and initial set of transforms**

- **2010 - 2012: Ongoing discussions with a variety of operations players on requirements**

What sort of basic things can you do with CybOX?

- Almost every field is optional. This means you can use whatever is appropriate and ignore the rest.
- Layered typing structure enabling flexible use
- Built in extensibility mechanisms
- Can specify and characterize a wide range of cyber objects
- Can specify and characterize dynamic cyber events & actions
- Can specify and characterize complex actions
- Can define relational and logical compositions of multiple objects, actions, events and/or observables
- Define a wide myriad of potential observable pattern variations
  - at the logical composition level

    or
  - utilizing patterns at the Object attribute level including
    - Equals, Contains, IsInRange, IsInSet, Regex, etc.

      all of which allow the user to define an almost infinitely variable set of patterns and filters

Various Defined Object Schemas

- Account
- Address
- API
- Code
- Device
- Disk
- Disk Partition
- DNS Cache
- DNS_Record
- Email Message
- File
- GUI
- GUI Dialog Box
- GUI Window
- Library
- Linux Package
- Memory
- Mutex
- Network Flow
- Network Packet
- Network Route Entry
- Network Route
- Network Subnet
- Pipe
- Port
- Process
- Product
- Semaphore
- Socket
- System
- Unix File
- Unix Network Route Entry
- Unix Pipe
- Unix Process
- Unix User Account
- Unix Volume
- URI
- User Account
- User Session
- Volume
- Win Computer Account
- Win Critical Section
- Win Driver
- Win Event
- Win Event Log
- Win Executable File
- Win File
- Win Kernel
- Win Kernel Hook
- Win Handle
- Win Mailslot
- Win Mutex
- Win Pipe
- Win Network Route Entry
- Win Network Share
- Win Pipe
- Win Prefetch
- Win Process
- Win Registry Key
- Win Semaphore
- Win Service
- Win System
- Win System Restore
- Win Task
- Win Thread
- Win User Account
- Win Volume
- Win Waitable Timer
- X509 Certificate

…

(more on the way)

Where does CybOX fit for Indicator and Incident sharing?

CybOX Is

- Language for describing cyber observables
- Both static & dynamic (Stateful Measures & Events)
- Comprehensive
- Richly expressive
- Instance & Pattern
- Flexible & extensible
- Composable/Decomposable
- Bridging a wide range of use cases (Indicator sharing, incident investigation, malware analysis, attack patterns, event management, digital forensics, etc.)

CybOX Is Not

- About Incidents
- Sensitivity and data handling
- Impact characterizations
- Expectations and action guidance
- Transport layer

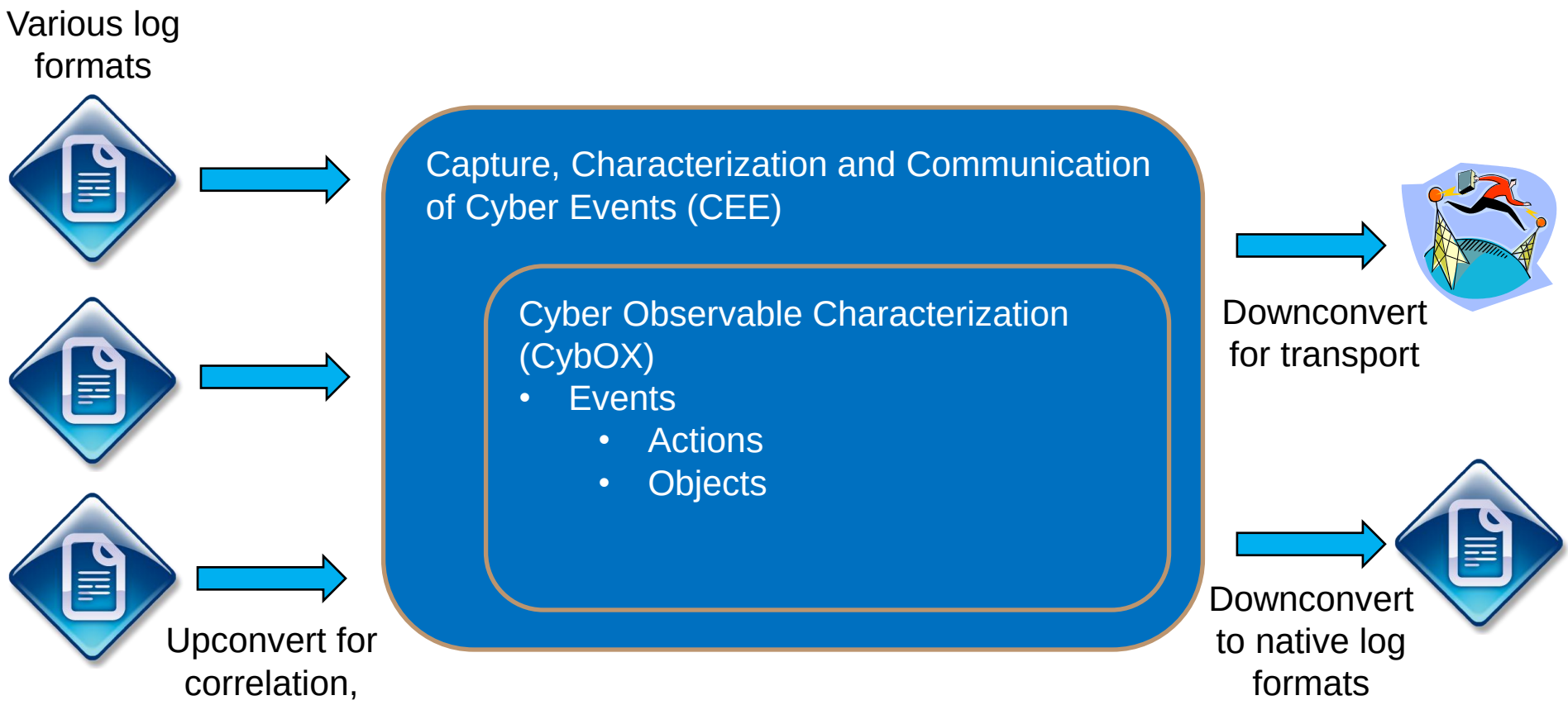
Wide Range of Supported Use Cases

- **Use Case Area: Event Management**
  - Producing Event Data (Log-Centric and Non-Log-Centric)
  - Exchanging Event Data
  - Analyzing Event Data
  - Querying Event Data
  - Composing Events
- **Use Case Area: Attack Patterns and Threat Characterization**
  - Characterizing Observable Evidence of Granular Attacker Actions
  - Characterizing Observable Evidence of Attacker Preparatory Probing Techniques
  - Characterizing Observable Evidence of Attacker Obfuscation Techniques
  - Characterizing Observable Evidence of Abstract Attack Patterns
- **Use Case Area: Cyber Threat Indicator Sharing**
  - Generating Cyber Threat Indicators
  - Exchanging Cyber Threat Indicators

Wide Range of Supported Use Cases (cont.)

- **Use Case Area: Attack Detection**
  - Detecting Dynamic In-Progress Attacks
  - Detecting Past Attacks
- **Use Case Area: Incident Investigation**
  - Correlating Incident Initiation Data
  - Excavating Incident Context
- **Use Case Area: Malware Analysis & Management**
  - Analyzing Malware Instances
  - Analyzing Malware Patterns
  - Hunting Malware Artifacts
  - Metadata Indexing Malware Collections
  - Exchanging Malware Characterizations
- **Use Case Area: Digital Forensics**
  - Conducting Digital Forensic Analysis
  - Managing Evidentiary Process

Use Case: Log Management



Use Case: Attack Pattern Characterization

Threat Information

- Attacker motivation
- Attacker behavior (TTP)
- Attacker capability
- Targeted attack surface
- Potential impact
- **How to detect threat**
- **How to identify threat**
- How to mitigate threat

Characterize, Codify, Communicate and Query on Common Patterns of Attack (CAPEC)

- Attack Flow Description

Cyber Observable Characterization (CybOX)

- Events
- Stateful Measures

- Required Skill & Knowledge
- Intent & Consequences
- Targeted Weaknesses
- Mitigations & Solutions
- Etc.

Use Case: Cyber Threat Indicator Generation

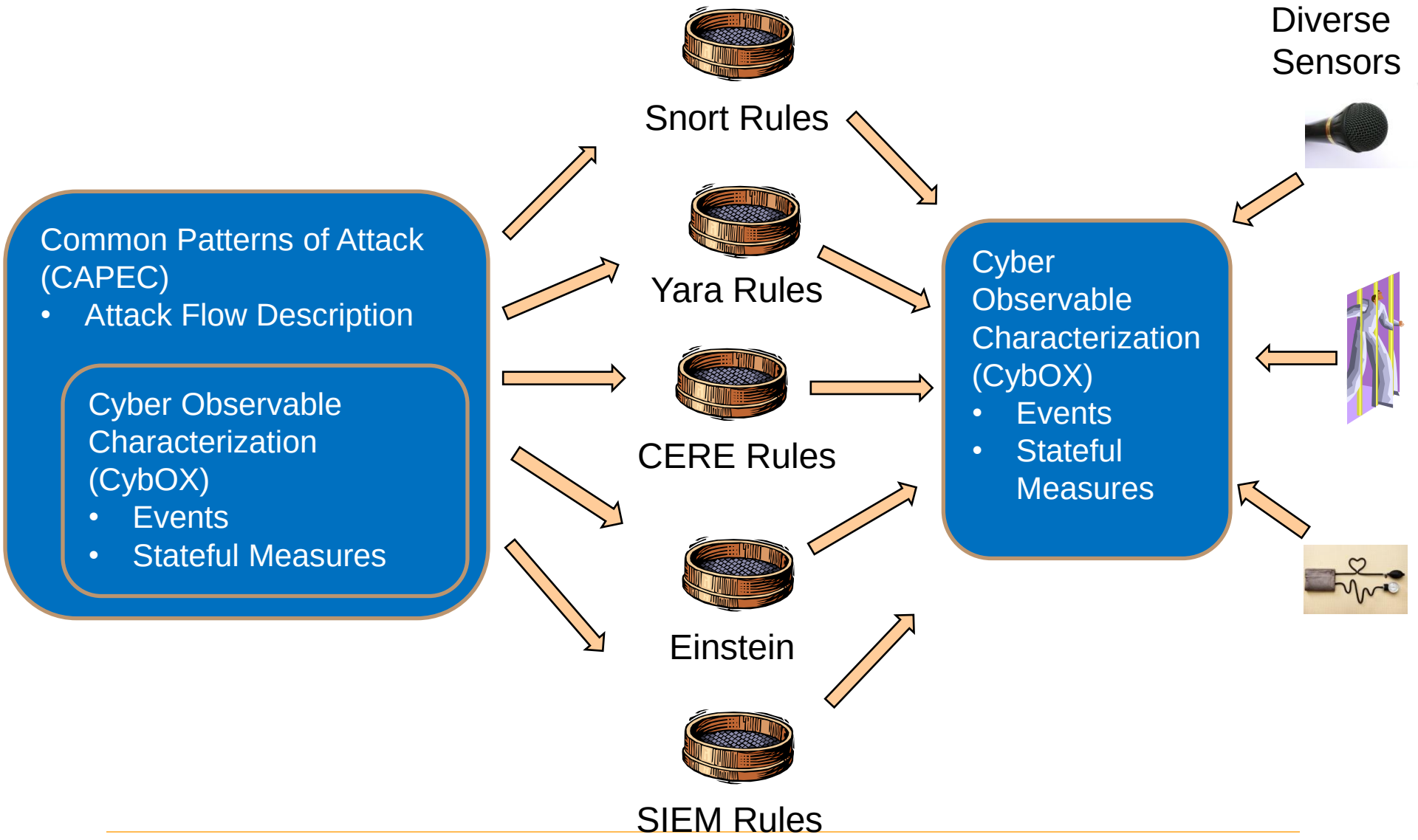
Cyber Threat Indicator Characterization (IndEX)

- Associated TTP
- Confidence
- Handling guidance
- Valid time windows
- Producer
- Etc.

Cyber Observable Characterization (CybOX)

- Events
- Stateful Measures

Use Case: Detect Malicious Activity



Where is CybOX today?

- CybOX Version 1.0(draft) released mid-April 2012
- Currently integrated into CAPEC
- Currently integrated into MAEC
- In process of being integrated into CEE
- Part of the strategic approach for EMAP
- Part of the strategic vision for IR/IM with US-CERT
- Currently integrated into efforts on Indicator schema development (IndEX)
- Currently being elaborated for Digital Forensics
- Currently under consideration as basis for several cyber threat information sharing communities
- Currently being evaluated for integration into multiple research projects

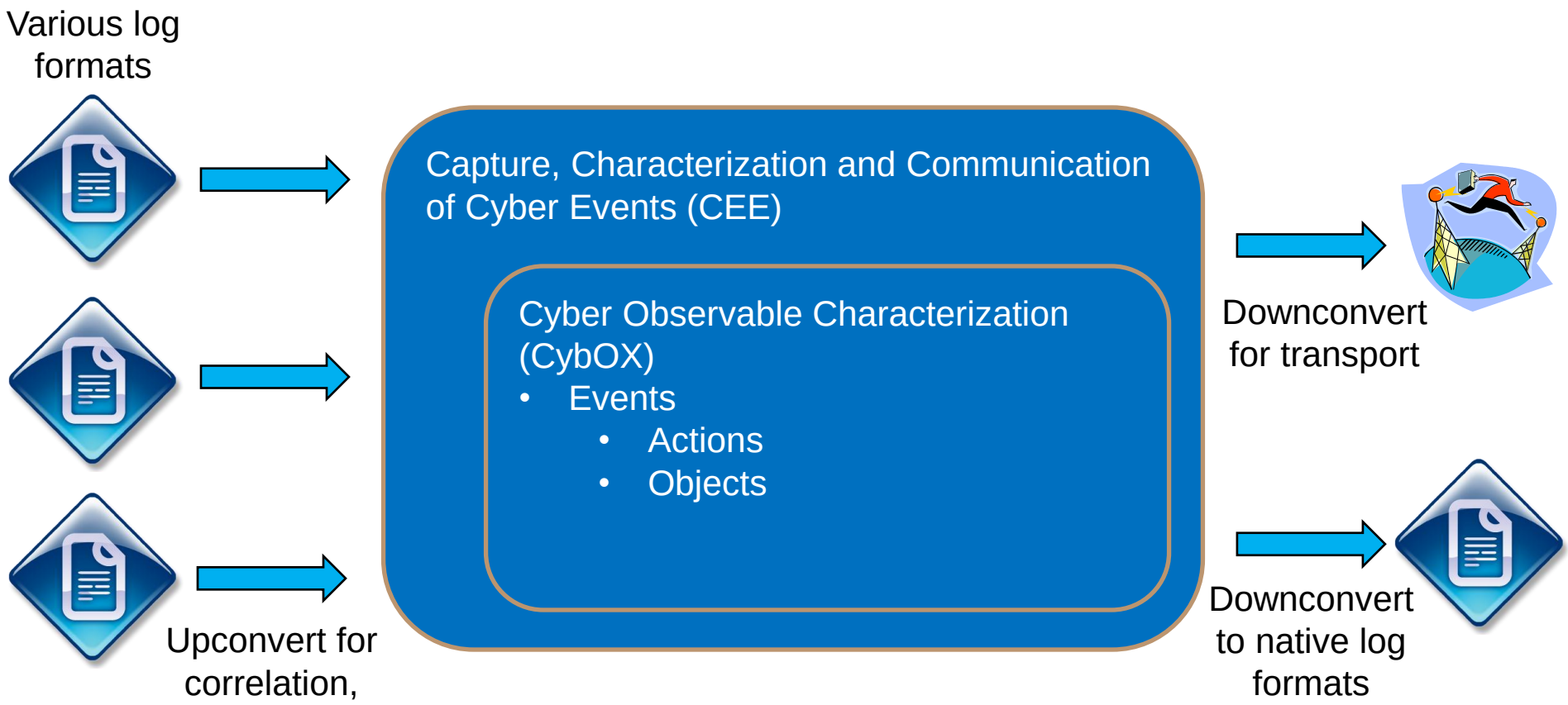
Questions / Comments?

Sean Barnum
sbarnum@mitre.org

Backup

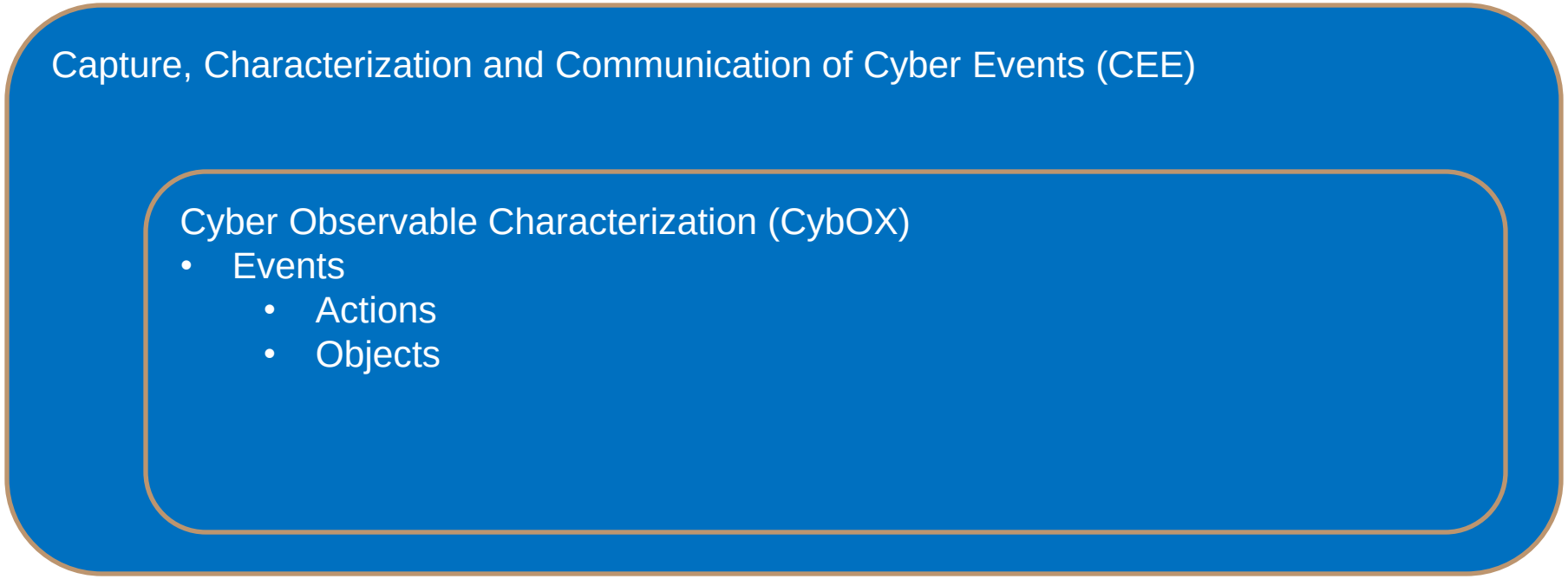
MITRE

Use Case: Log Management



Use Case: Event Management

The Event Management Automation Protocol (EMAP) is a program to establish a protocol to enable standardized content, representation, exchange, correlation, searching, storing, prioritization, and auditing of event records within an organizational IT environment



Use Case: Attack Pattern Characterization

Threat Information

- Attacker motivation
- Attacker behavior (TTP)
- Attacker capability
- Targeted attack surface
- Potential impact
- **How to detect threat**
- **How to identify threat**
- How to mitigate threat

Characterize, Codify, Communicate and Query on Common Patterns of Attack (CAPEC)

- Attack Flow Description

Cyber Observable Characterization (CybOX)

- Events
- Stateful Measures

- Required Skill & Knowledge
- Intent & Consequences
- Targeted Weaknesses
- Mitigations & Solutions
- Etc.

Use Case: Cyber Threat Indicator Generation

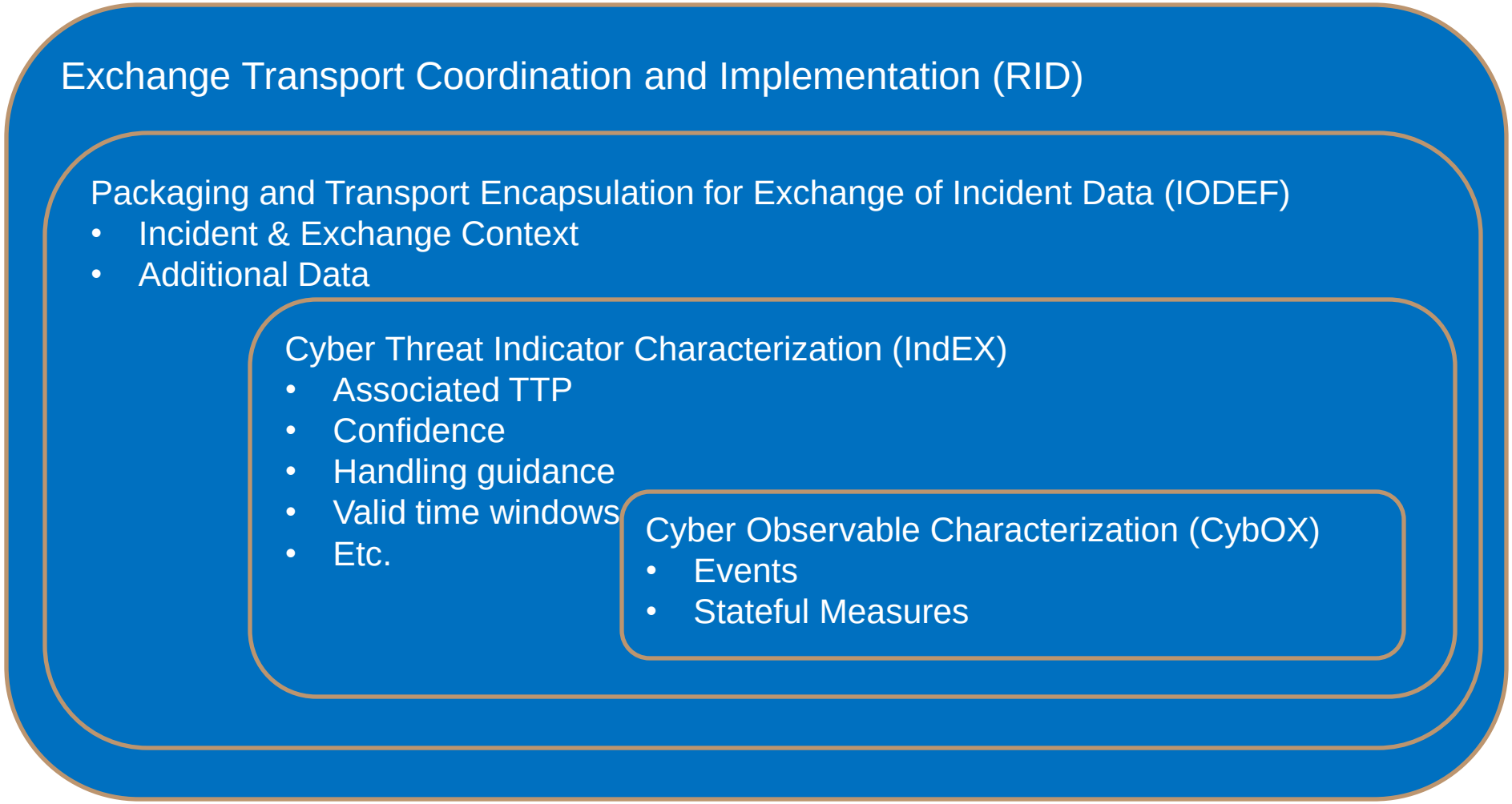
Cyber Threat Indicator Characterization (IndEX)

- Associated TTP
- Confidence
- Handling guidance
- Valid time windows
- Producer
- Etc.

Cyber Observable Characterization (CybOX)

- Events
- Stateful Measures

Use Case: Cyber Threat Indicator Exchange



Use Case: Malware Analysis & Exchange

Analysis and Characterization of Malware (MAEC)
- Mechanisms
- Behaviors

Cyber Observable Characterization (CybOX)
- Actions
- Objects

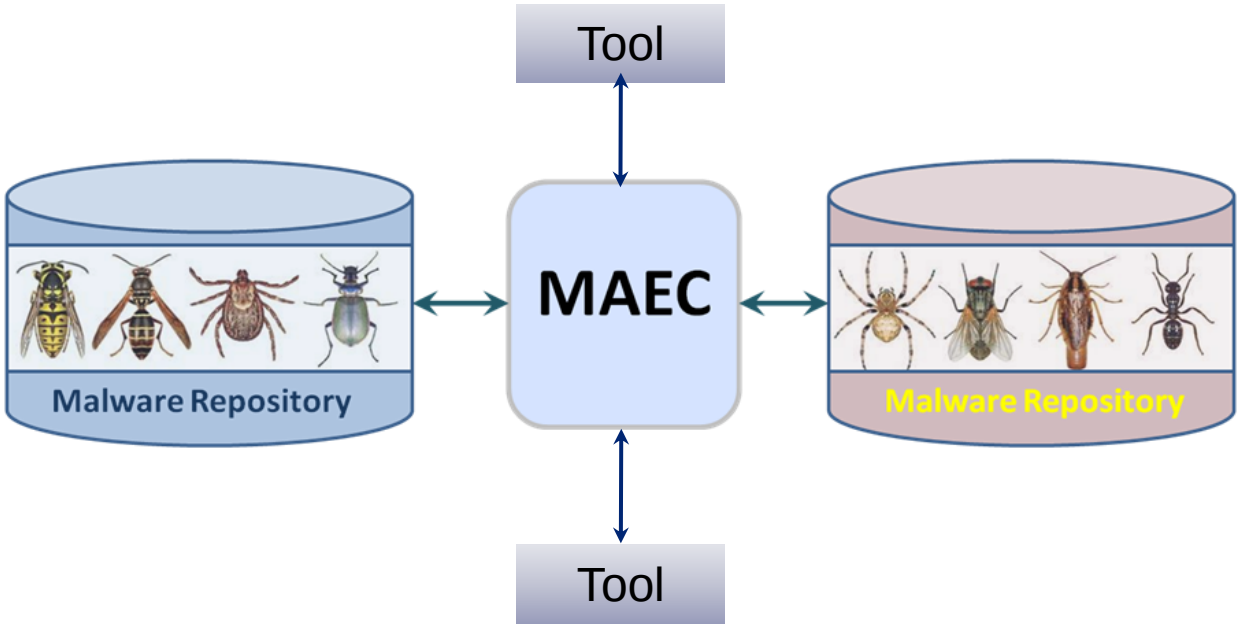
Use Case: Malware Analysis

- **Current:**
  - **Difficult to combine different analysis perspectives or tools**
  - **Difficult to share info**
  - **Difficult to recognize if malware has been seen before**
  - **Does not scale well**

- **CybOX(MAEC)-enabled:**
  - **Easier to integrate different forms of analysis, different tools and even information from different sources**
  - **Easier to share information**
  - **Easier to recognize malware (including variants and pertubations)**
  - **Enables automated interaction among the various dimensions of malware analysis**

MAEC(CybOX) Use Cases

- **Analysis**
  - **Help Guide Analysis Process**
  - **Standardized Tool Output**
  - **Malware Repositories**



```
<maec:MAEC_Bundle schema_version="2.0">
<maec:Behaviors>
 <maec:Behavior id="maec:storm:bhv:1" ordinal_position="1" successful="true">
  <maec:Description>A Storm Worm behavior that instantiates itself as a rootkit-enabled binary on a
system</maec:Description>
  <maec:Actions>
   <maec:Action id="maec:storm:act:1" ordinal_position="1" type="Create">
    <maec:Description>Storm first drops the malicious binary on the system</maec:Description>
    <maec:Affected_Objects>
     <maec:Affected_Object effect_type="Created" target_object_id="maec:storm:obj:1"/>
    </maec:Affected_Objects>
   </maec:Action>
   <maec:Action id="maec:storm:act:2" ordinal_position="2" type="Create">
    <maec:Description>Storm then instantiates the binary as a service</maec:Description>
    <maec:Affected_Objects>
     <maec:Affected_Object effect_type="Created" target_object_id="maec:storm:obj:2"/>
    </maec:Affected_Objects>
   </maec:Action>
  </maec:Actions>
 </maec:Behavior>
</maec:Behaviors>

 <maec:Objects>
  <maec:Object id="maec:storm:obj:1" Type="File">
   <observables:Defined_Object xsi:type="WinFileObj:Windows_File_Object_Type">
    <FileObj:File_Name>spooldr.exe</FileObj:File_Name>
    <FileObj:Full_Path>C:\Windows</FileObj:Full_Path>
   </observables:Defined_Object>
  </maec:Object>
  <maec:Object id="maec:storm:obj:2" Type="Key/Key Group">
   <observables:Defined_Object xsi:type="WinRegistryObj:Windows_Registry_Object_Type">
    <WinRegistryObj:Hive>HKEY_LOCAL_MACHINE</WinRegistryObj:Hive>
    <WinRegistryObj:Key>SYSTEM/CurrentControlSet\Services\spooldr</WinRegistryObj:Key>
   </observables:Defined_Object>
  </maec:Object>
 </maec:Objects>
</maec:MAEC_Bundle>
```

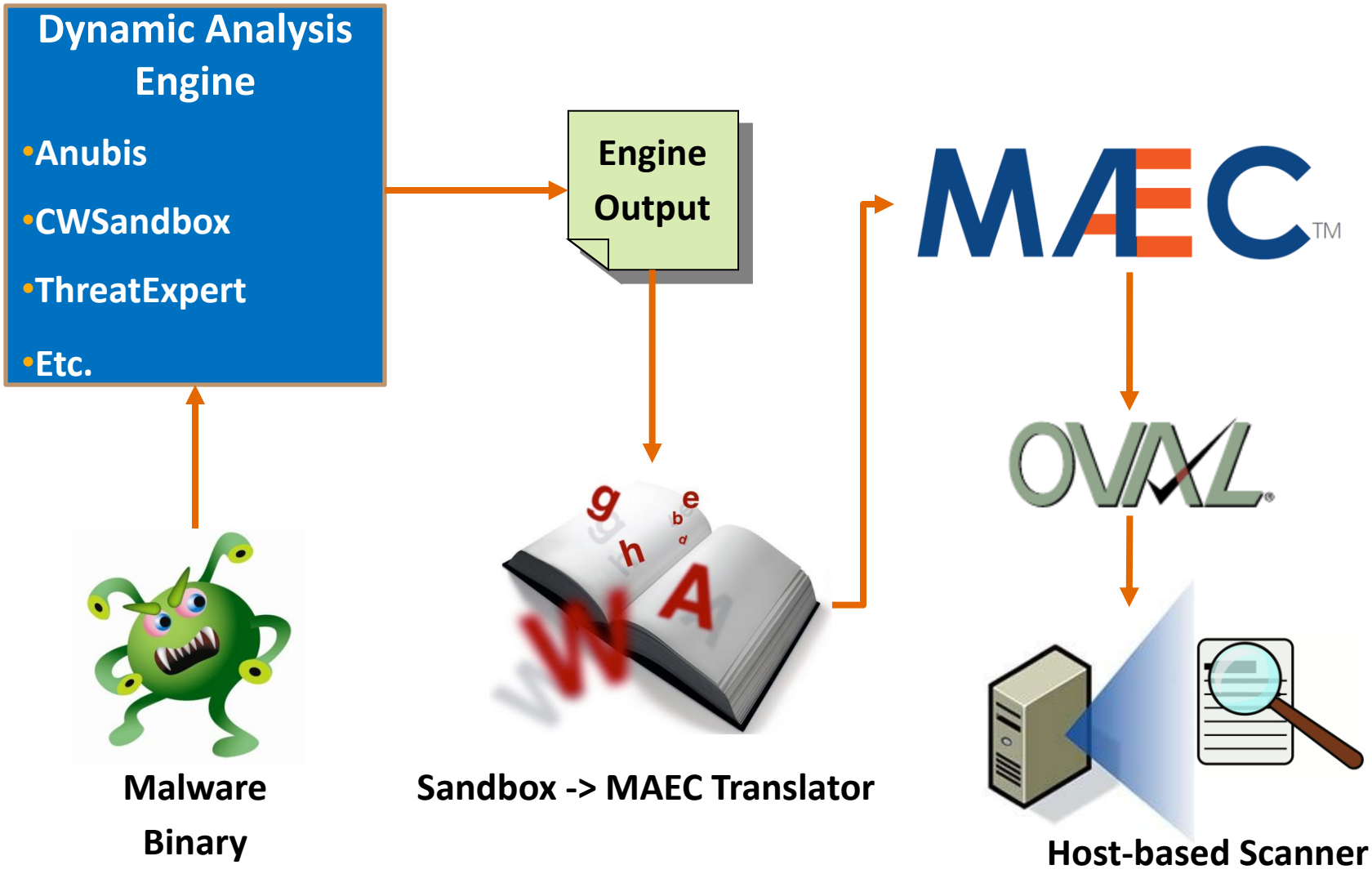
Portion of MAEC Analysis of Storm Worm

Use Case: Malware Artifact Hunting

- **Current:**
  - **Very manual**
  - **Often imprecise and inconsistent**
  - **Localized**

- **CybOX(MAEC)-enabled:**
  - **Very automated**
  - **Consistent**
  - **Enables broad, non-localized sharing and hunting**

Use Case: Host Based Detection



Real World Example: MAEC & Zeus Bot

Anubis Output*

Anubis → MAEC Translator Script

MAEC Output

```
MAEC_Bundle "1"
  Analyses
    Analysis "3"
      Subject
      Tools_Used
  Pools
    Action_Collection_Pool
      Action_Collection "2382"
      Action_Collection "2383"
      Action_Collection "2386"
      Action_Collection "2388"
      Action_Collection "2385"
    Object_Collection_Pool
      Object_Collection "2374"
      Object_Collection "2373"
      Object_Collection "2376"
      Object_Collection "2372"
      Object_Collection "2371"
```

OVAL Output

```
oval_definitions "http://oval.mitre.org/XMLSchema/oval-d
  generator MAEC XML to OVAL Script
    oval:product_name MAEC XML to OVAL Script
    oval:product_version 1.0
    oval:schema_version 5.7
    oval:timestamp 2010-11-08T09:26:33.347000
  definitions
  tests
    win-def:file_test "oval:maec_out:tst:1"
    win-def:file_test "oval:maec_out:tst:2"
    win-def:file_test "oval:maec_out:tst:3"
    win-def:file_test "oval:maec_out:tst:4"
    win-def:file_test "oval:maec_out:tst:5"
    win-def:file_test "oval:maec_out:tst:6"
    win-def:registry_test "oval:maec_out:tst:7"
    win-def:registry_test "oval:maec_out:tst:8"
    win-def:registry_test "oval:maec_out:tst:9"
    win-def:registry_test "oval:maec_out:tst:10"
  objects
    win-def:file_object "oval:maec_out:obj:1" C:\DO
    win-def:file_object "oval:maec_out:obj:2" C:\Doc
    win-def:file_object "oval:maec_out:obj:3" C:\Doc
    win-def:file_object "oval:maec_out:obj:4" C:\Doc
    win-def:file_object "oval:maec_out:obj:5" C:\Doc
    win-def:file_object "oval:maec_out:obj:6" C:\Doc
    win-def:registry_object "oval:maec_out:obj:7" H
    win-def:registry_object "oval:maec_out:obj:8" H
    win-def:registry_object "oval:maec_out:obj:9" H
    win-def:registry_object "oval:maec_out:obj:10" I
```

Anubis Sandbox

Zeus Binary

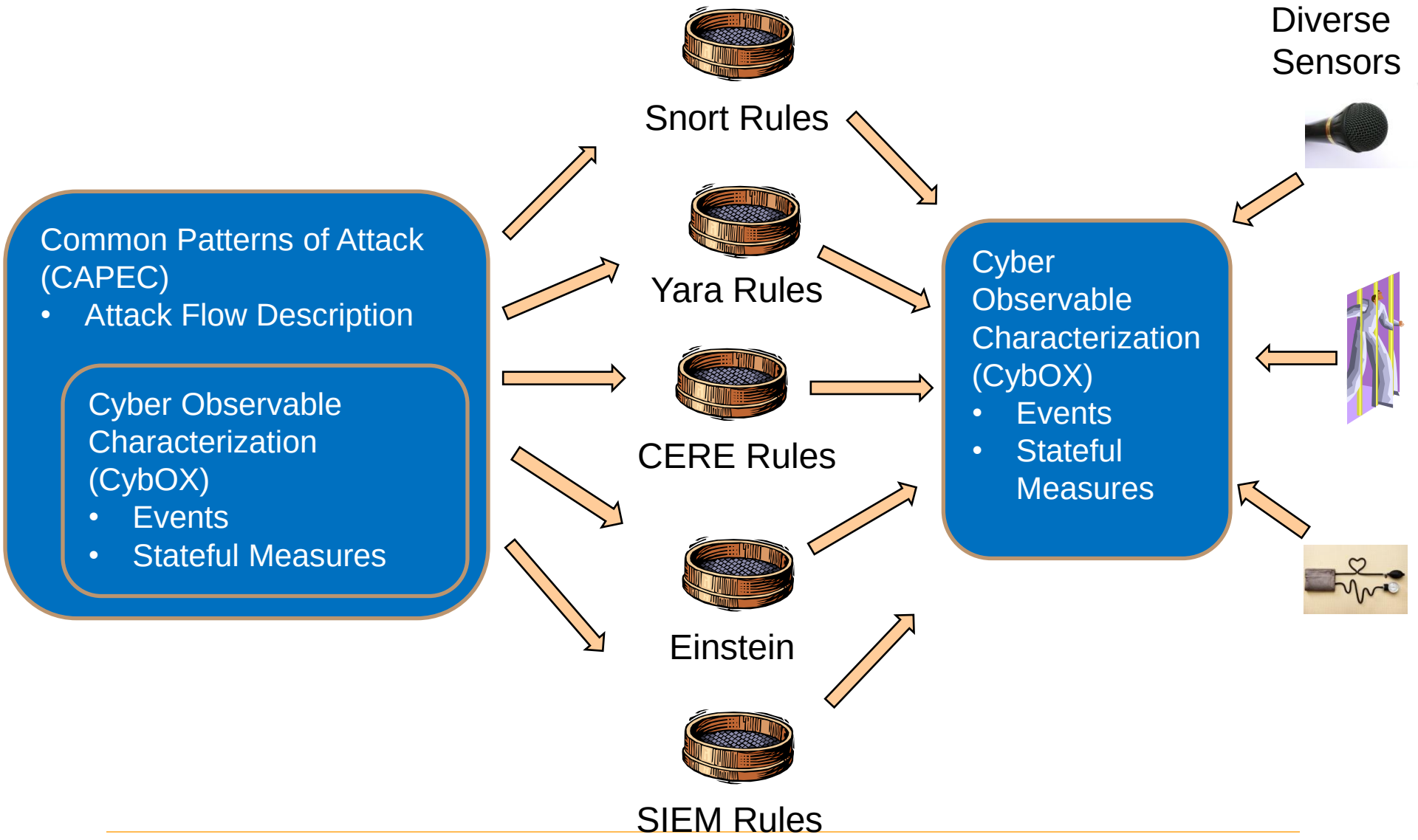
MAEC → OVAL Translator Script

*http://anubis.iseclab.org/?action=result&task_id=1167a57d1aa905e949df5d5478ab23bf9

```
<oval_definitions>
<definitions>
  <definition class="compliance" id="oval:storm:def:1" version="1">
    <metadata>
      <title>Storm Worm Test</title>
      <affected family="windows"/>
      <description>A test for a file and a registry key associated with the Storm Worm</description>
    </metadata>
    <criteria operator="AND">
      <criterion comment="File test" test_ref="oval:storm:tst:1"/>
      <criterion comment="Registry key test" test_ref="oval:storm:tst:2"/>
    </criteria>
  </definition>
 </definitions>
 <tests>
  <win-def:file_test id="oval:storm:tst:1" version="1" check="all" comment="test for a Storm file"
check_existence="all_exist">
    <win-def:object object_ref="oval:storm:obj:1"/>
  </win-def:file_test>
  <win-def:registry_test id="oval:storm:tst:2" version="1" check="all" comment="test for a Storm registry key"
check_existence="all_exist">
    <win-def:object object_ref="oval:storm:obj:2"/>
  </win-def:registry_test>
 </tests>
 <objects>
  <win-def:file_object id="oval:storm:obj:1" version="1">
    <win-def:path>C:\System</win-def:path>
    <win-def:filename>spooldr.exe</win-def:filename>
  </win-def:file_object>
  <win-def:registry_object id="oval:storm:obj:2" version="1">
    <win-def:hive>HKEY_LOCAL_MACHINE</win-def:hive>
    <win-def:key>SYSTEM/CurrentControlSet\Services\spooldr</win-def:key>
    <win-def:name xsi:nil="true"/>
  </win-def:registry_object>
 </objects>
</oval_definitions>
```

OVAL Checks Transformed from Portion of MAEC Analysis of Storm Worm

Use Case: Detect Malicious Activity



Use Case: Detect Malicious Activity

- **Current:**
  - **Manual effort to pull together data across many sensors**
    - **Results in limited situational awareness**
  - **Attack patterns and rules are typically too detailed (physical signatures) or ambiguous prose**
  - **High level of effort**
  - **High false negatives & positives**

- **CybOX-enabled:**
  - **Diverse set of sensors output data in common format**
  - **Attack patterns and rules can be defined in a uniform fashion**
  - **Pattern matching and analysis heuristics can be easily automated**

```
<Observables>
 <Observable>
    <Measure>
      <Pattern>
        <Event Type="File Ops (CRUD)">
          <Description>Storm drops a rootkit enabled binary on the system</Description>
          <Action Name="Create">
            <Object Type="File" Action_Role="Target">
              <Defined_Object xsi:type="WinFileObj:Windows_File_Object_Type">
                <FileObj:File_Name>spooldr.exe</FileObj:File_Name>
                <FileObj:Full_Path>C:\Windows</FileObj:Full_Path>
              </Defined_Object>
            </Object>
          </Action>
        </Event>
      </Pattern>
    </Measure>
 </Observable>
 <Observable>
   <Measure>
     <Pattern>
       <Event Type="Registry Ops">
         <Description>Storm creates a registry key to load itself as a service</Description>
         <Action Name="Create">
           <Object Type="Key/Key Group" Action_Role="Target">
             <Defined_Object xsi:type="WinRegistryObj:Windows_Registry_Object_Type">
               <WinRegistryObj:Hive>HKEY_LOCAL_MACHINE</WinRegistryObj:Hive>
               <WinRegistryObj:Key>SYSTEM/CurrentControlSet\Services\spooldr</WinRegistryObj:Key>
             </Defined_Object>
           </Object>
         </Action>
       </Event>
     </Pattern>
   </Measure>
 </Observable>
</Observables>
```

CybOX Content Transformed from Portion of MAEC Analysis of Storm Worm

Use Case: Incident Information (SCI) Exchange

Packaging and Transport Encapsulation for Exchange of Incident Data (IODEF)
- Incident & Exchange Context
- Additional Data

Cyber Observable Characterization (CybOX)
- Events
- Stateful Measures

Attack Patterns (CAPEC)

Malware Characterization (MAEC)

Etc.

Use Case: Incident Response Data Capture

- **Current:**
  - **Very manual**
  - **Inconsistent between analysts & organizations**
  - **Prose-based and imprecise**
  - **Difficult to automate capture and actionable alerts**

- **CybOX-enabled:**
  - **Improved consistency**
  - **Ability to tie everything together**
  - **Simplified and automated data capture**
  - **Alerts become actionable for automation**

Use Case: IR/IM Alerts

- **Current:**
  - **Typically unstructured prose**
  - **Labor intensive and slow**
  - **Limited actionable (in an automated fashion) data**

- **CybOX-enabled:**
  - **Structured and consistent**
  - **Alert generation can be much faster and less labor intensive**
  - **Potentially actionable in an automated context**

Use Case: Localized Forensic Analysis Utilizing a Potentially Diverse set of Forensic & Non-forensic Tools

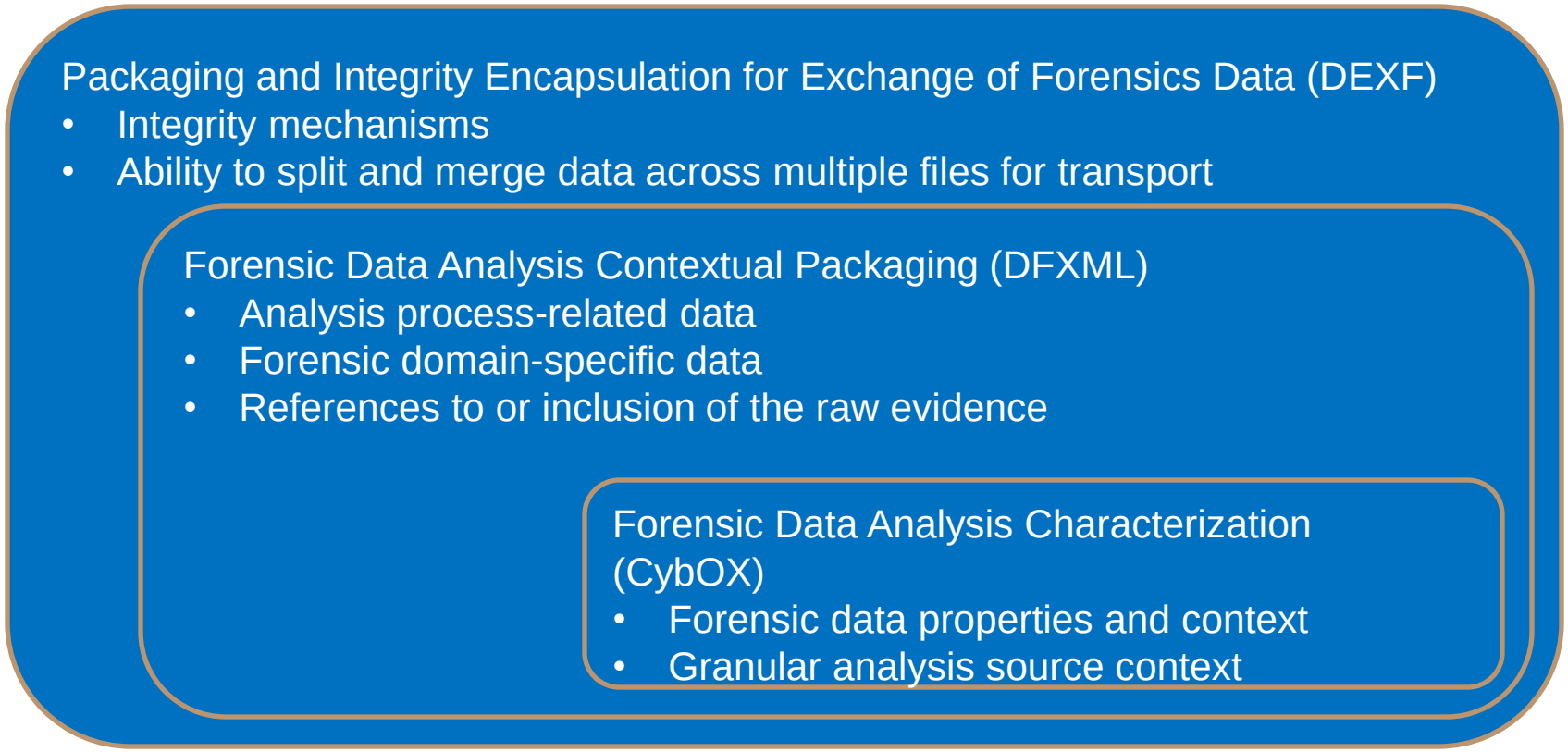
Forensic Data Analysis Contextual Packaging (DFXML)

- Analysis process-related data
- Forensic domain-specific data
- References to or inclusion of the raw evidence

Forensic Data Analysis Characterization (CybOX)

- Forensic data properties and context
- Granular analysis source context

Use Case: Forensics Exchange between Analysts Working the Same Problem



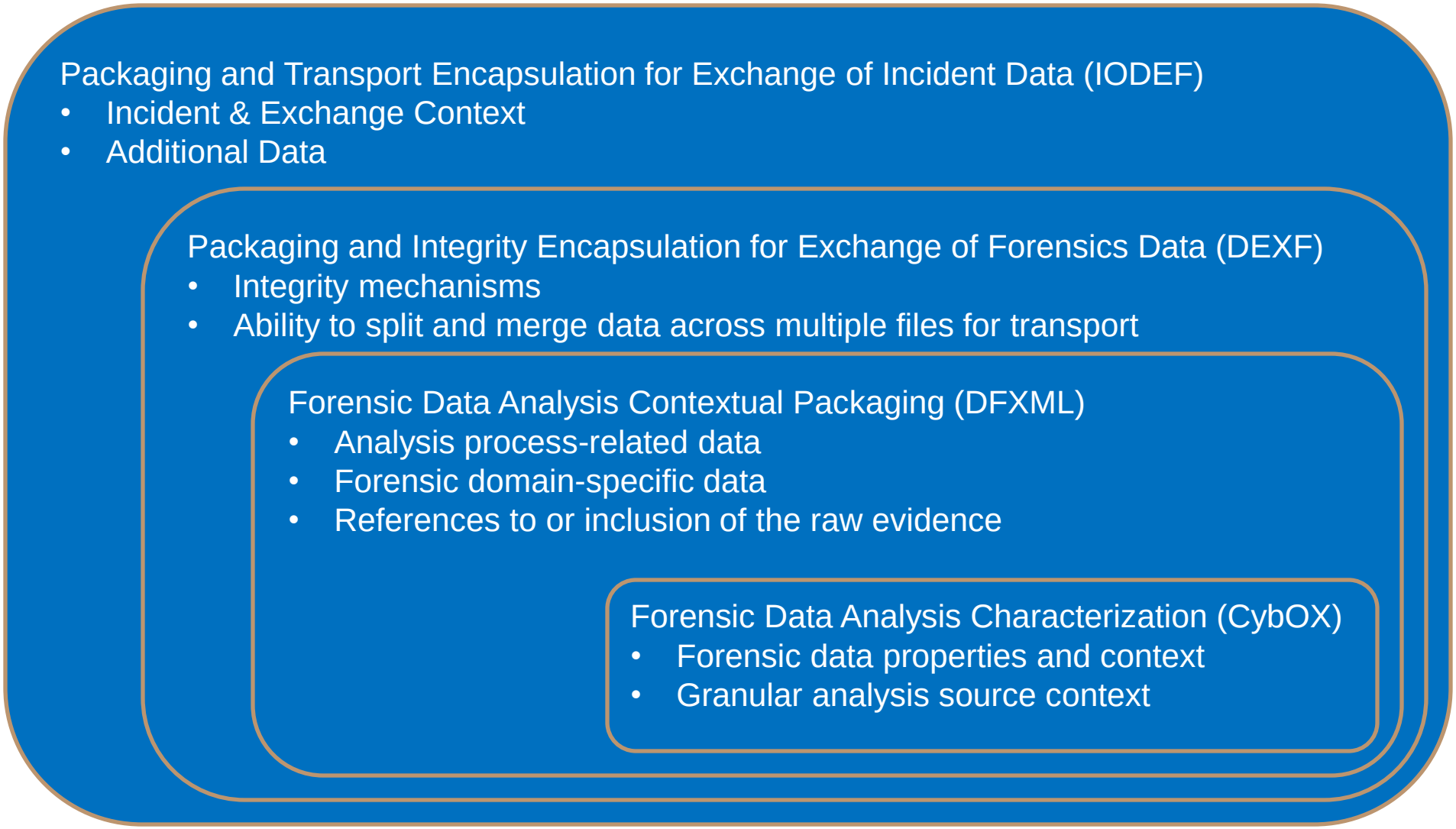
Use Case: Evidentiary Forensics Exchange (e.g. LE or IC)

Packaging and Integrity Encapsulation for Exchange of Forensics Data (DEXF)
- Integrity mechanisms
- Ability to split and merge data across multiple files for transport

Forensic Data Analysis Contextual Packaging (DFXML)
- Analysis process-related data
- Forensic domain-specific data
- References to or inclusion of the raw evidence

Forensic Data Analysis Characterization (CybOX)
- Forensic data properties and context
- Granular analysis source context

Use Case: Incident-related Forensics Exchange



Notional Flow of a Modern Security Incident

1. An attack on an information system occurs involving social engineering, vulnerability exploit, malware + command and control (C2).
2. CybOX-enabled operational sensors (IDS, host-based, etc.) pick up anomalous activity and report it in CEE/CybOX formats.
3. Automated analysis tools & rules attempt to match anomalous activity against CybOX-adorned CAPEC attack patterns but discover no matching patterns.
4. Incident is reported – Incident Response/Management process is initiated.
5. IR personnel capture discovered detail of incident in CybOX-compliant formats, including CEE.
6. IR personnel detect malware as part of the ongoing attack.

Notional Flow of a Modern Security Incident (cont.)

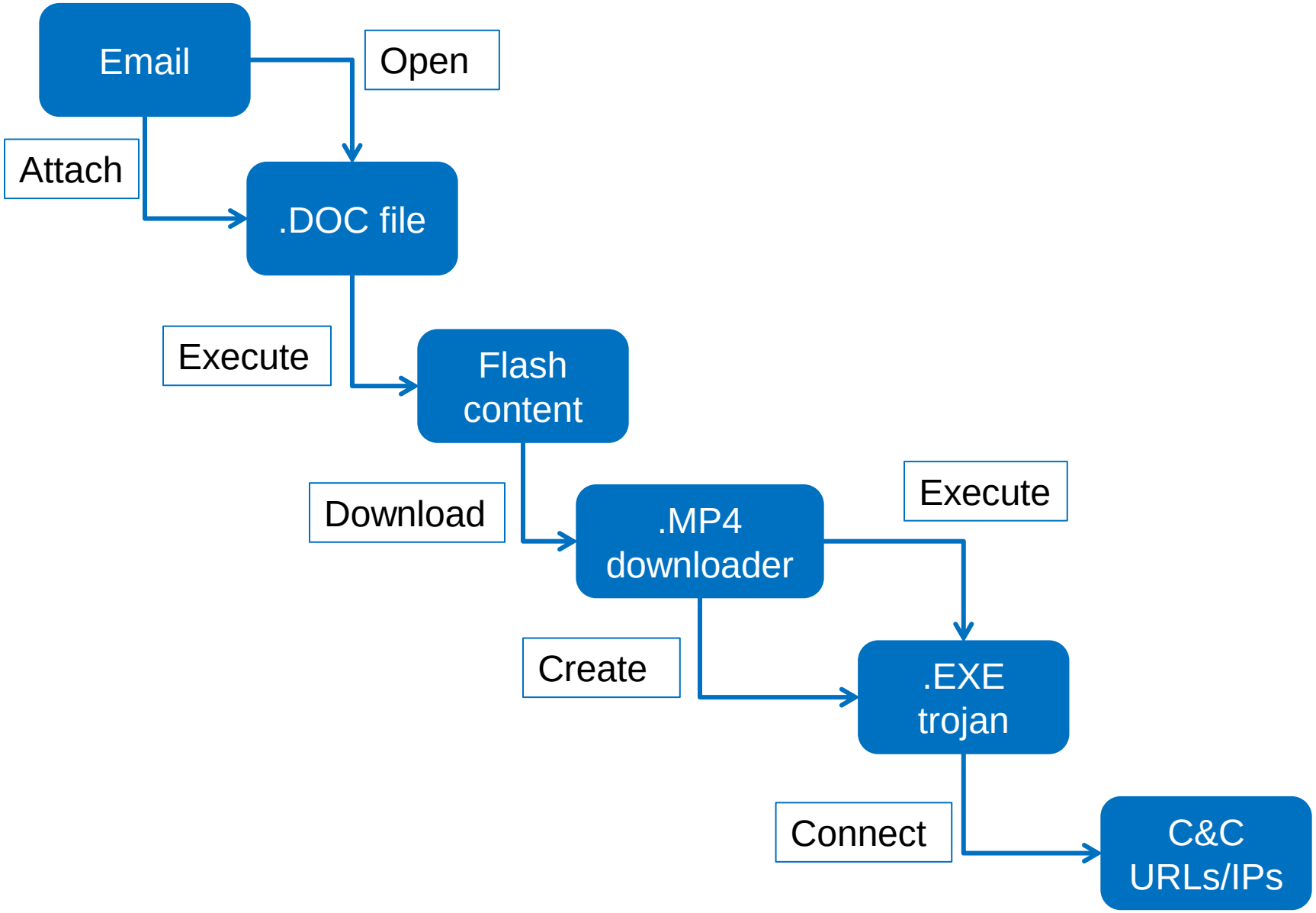
7. Malware undergoes automated analysis (dynamic and/or static) and results are captured in MAEC (CybOX-integrated) language.
8. Malware analysts are able to correlate the current malware instance with a broad range of pre-existing malware samples and analysis data from MAEC-enabled repositories and zoos.
9. Malware analysts capture new discovered detail of the malware in MAEC format, including the CWE or CVE exploited .
10. Sample and analysis data from current malware instance are entered into appropriate malware repositories and zoos.
11. CybOX observables of malware effects on hosts are extracted from MAEC content to generate OVAL checks to determine if any given host has been infected/affected by the current malware instance.
12. OVAL checks are distributed and run against other areas of the domain or enterprise to determine breadth of compromise.

Notional Flow of a Modern Security Incident (cont.)

13. IR/IM personnel apply appropriate mitigations/remediations to negate the effects of the attack.
14. A new CAPEC attack pattern is authored to describe this new observed attack behavior, and is adorned as appropriate with CybOX content observed for this pattern in the operational space.
15. IR/IM personnel issue relevant alerts for the observed incident including the new CAPEC pattern, MAEC bundle and related CEE/CybOX content.
16. Secure development takes advantage of this new CAPEC pattern to: define/refine appropriate security policy, training & requirements; guide security engineering (control selection), architectural risk analysis, secure code review and security testing; identify relevant CWE weaknesses, CVE vulnerabilities & CCE configuration issues; prioritize relevant CAPEC patterns based on real-world observed prevalence/frequency profiled through automated observation of CybOX patterns in the operational space .

Example: Real-world Recent Attack Campaign

- **Known as Iran-Oil and many other names**
- **Email phishing attack with malicious .DOC attachment that contains flash that downloads a malicious .MP4 file which installs and runs a trojan .EXE file which connects to multiple C&C URLs/IPs**



Simple Email Info

```
<cybox:Object ID="51359587-f201-4383-b032-5a64522fcd7d" Type="Email Message">
   <cybox:Defined_Object xsi:type="EmailMessageObj:Email_Message_Object_Type">
      <cybox:Attachments><cybox:File IDREF="49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7"
Type="File"/></cybox:Attachments>
      <cybox:Header>
            <cybox:Date Datatype="DateTime">March 2, 2012 7:42:24 EST</cybox:Date>
            <cybox:From category="e-mail">
               <addressObject:Address_Value
Datatype="String">wmorrison89@gmail.com</addressObject:Address_Value>
            </cybox:From>
            <cybox:Subject Datatype="String">Iran's Oil and Nuclear
Situation</cybox:Subject>
            <cybox:To category="e-mail">
               <addressObject:Address_Value
Datatype="String">william.abnett@gmail.com</addressObject:Address_Value>
            </cybox:To>
      </cybox:Header>
   </cybox:Defined_Object>
</cybox:Object>
```

```
<cybox:Object ID="51359587-f201-4383-b032-5a64522fcd7d" Type="Email Message">
        <cybox:Defined_Object xsi:type="EmailMessageObj:Email_Message_Object_Type">
            <cybox:Attachments><cybox:File IDREF="49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7"
Type="File"/></cybox:Attachments>
            <cybox:Header>
                <cybox:Date Datatype="DateTime">March 2, 2012 7:42:24 EST</cybox:Date>
                <cybox:From category="e-mail"><addressObject:Address_Value
Datatype="String">wmorrison89@gmail.com</addressObject:Address_Value></cybox:From>
                <cybox:Subject Datatype="String">Iran's Oil and Nuclear Situation</cybox:Subject>
                <cybox:To category="e-mail"><addressObject:Address_Value
Datatype="String">william.abnett@gmail.com</addressObject:Address_Value></cybox:To>
            </cybox:Header>
            <cybox:Raw_Header Datatype="String"><![CDATA[
Return-Path: <wmorrison89@gmail.com>
Received-SPF: pass (google.com: domain of wmorrison89@gmail.com designates
10.236.185.4 as permitted sender) client-ip=10.236.185.4;
Authentication-Results: mr.google.com; spf=pass (google.com: domain of
wmorrison89@gmail.com designates 10.236.185.4 as permitted sender)
smtp.mail=wmorrison89@gmail.com; dkim=pass header.i=wmorrison89@gmail.com
Received: from mr.google.com ([10.236.185.4]) by 10.236.185.4 with SMTP
id t4mr5301660yhm.129.1330692273662 (num_hops = 1); Fri, 02 Mar 2012
04:44:33 -0800 (PST)
MIME-Version: 1.0
Received: by 10.236.185.4 with SMTP id t4mr4236541yhm.129.1330692265380;
Fri,
02 Mar 2012 04:44:25 -0800 (PST)
Received: by 10.147.35.14 with HTTP; Fri, 2 Mar 2012 04:44:24 -0800 (PST)
In-Reply-To:
<CADY6HTa-jmaqmtVyyT-nLz6reztNjcs-617wL4bt9YBOGu+h4w@mail.gmail.com>
References:
<CADY6HTa-jmaqmtVyyT-nLz6reztNjcs-617wL4bt9YBOGu+h4w@mail.gmail.com>
Date: Fri, 2 Mar 2012 07:44:24 -0500
Message-ID:
<CADY6HTZ6oopY5v6WkYU81YcSQw3X124CK_Fx4jhnhUiU3Y9z6A@mail.gmail.com>
Subject: Iran's Oil and Nuclear Situation
From: william abnett <wmorrison89@gmail.com>
To: william.abnett <william.abnett@gmail.com>
Content-Type: multipart/mixed; boundary="20cf303f67fac8928804ba41efd5"]]>
            </cybox:Raw_Header>
        </cybox:Defined_Object>
</cybox:Object>
```

Malicious .DOC File (simple info)

```
<cybox:Object ID="49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7" Type="File">

   <cybox:Description><cybox:Text>The word document contains flash, which downloads a corrupted mp4 file. The mp4 file itself is not anything special but an 0C filled (22kb) mp4 file with a valid mp4 header.</cybox:Text></cybox:Description>

   <cybox:Defined_Object xsi:type="FileObj:File_Object_Type">
      <cybox:File_Name Datatype="String">Iran's Oil and Nuclear Situation.doc</cybox:File_Name>
      <cybox:Hashes><cybox:Hash>
            <cybox:Hash_Value Condition="Equals"
Datatype="BinHex">E92A4FC283EB2802AD6D0E24C7FCC857</cybox:Hash_Value>
            <cybox:Type Datatype="String">MD5</cybox:Type>
      </cybox:Hash></cybox:Hashes>
      <cybox:Size_In_Bytes Datatype="">106604</cybox:Size_In_Bytes>
   </cybox:Defined_Object>
</cybox:Object>
```

.MP4 Downloader Location

```
<cybox:Object ID="61041b8b-0c15-48a0-ac5f-b49488788010" Type="URI">
        <cybox:Defined_Object xsi:type="URIObj:URI_Object_Type"
Type="URL">
                <URIObj:Value Datatype="URI" Condition="Equals">
                        http://208.115.230.76/test.mp4
                </URIObj:Value>
        </cybox:Defined_Object>
</cybox:Object>
```

.MP4 Downloader (simple info)

```
<cybox:Object ID="8b463e0d-cc16-4036-950e-5eeb09bc51aa" Type="File">

   <cybox:Description><cybox:Text>This mp4 file causes memory corruption and code
execution via heap-spraying code injection.</cybox:Text></cybox:Description>

   <cybox:Defined_Object xsi:type="FileObj:File_Object_Type">
      <cybox:File_Name Datatype="String">test.mp4</cybox:File_Name>
      <cybox:Hashes>
            <cybox:Hash><cybox:Hash_Value Condition="Equals" Datatype="BinHex">
                  8933598C8B1FA5E493497B11C48DA4F2
            </cybox:Hash_Value>
            <cybox:Type Datatype="String">MD5</cybox:Type></cybox:Hash>
      </cybox:Hashes>
      <cybox:Size_In_Bytes Datatype="">22384</cybox:Size_In_Bytes>
   </cybox:Defined_Object>

   <cybox:Related_Objects>
      <cybox:Related_Object Relationship="Downloaded_By"><cybox:Object
IDREF="49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7" Type="File"/></cybox:Related_Object>
      <cybox:Related_Object Relationship="Downloaded_From"><cybox:Object
IDREF="61041b8b-0c15-48a0-ac5f-b49488788010" Type="URI"/></cybox:Related_Object>
   </cybox:Related_Objects>

</cybox:Object>
```

Trojan (us.exe)(simple info)

<cybox:Object ID="b7e0bc39-f519-4878-8fb0-5902554efe1c" Type="File">

<cybox:Description><cybox:Text>The file (us.exe MD5: FD1BE09E499E8E380424B3835FC973A8 4861440 bytes) is created in the logged in user %Temp% directory. The size of the embedded file is 22.5 KB (23040 bytes) and the size of the created us.exe is 4.63MB. It is an odd discrepancy until you look at the file and it looks like the code is repeated over and over - 211 times. The file resource section indicates the file is meant to look like a java updater, which is always larger than 22.5KB and that would explain all this padding, which is done at the time when the file is being written to the disk.</cybox:Text></cybox:Description>

<cybox:Defined_Object xsi:type="FileObj:File_Object_Type">
<cybox:File_Name Datatype="String">us.exe</cybox:File_Name>
<cybox:File_Path Datatype="String">%Temp%</cybox:File_Path>
<cybox:Hashes>
<cybox:Hash><cybox:Hash_Value Condition="Equals" Datatype="BinHex">
FD1BE09E499E8E380424B3835FC973A8
</cybox:Hash_Value><cybox:Type Datatype="String">MD5</cybox:Type></cybox:Hash>
</cybox:Hashes>
<cybox:Size_In_Bytes Datatype="">4861440</cybox:Size_In_Bytes>
</cybox:Defined_Object>
</cybox:Object>

Trojan (us-embedded.exe)

```
<cybox:Object ID="bed1ff22-08e8-4e04-b7ac-908b5271176f" Type="File">
        <cybox:Defined_Object xsi:type="FileObj:File_Object_Type">
                <cybox:File_Name Datatype="String">us-
embedded.exe</cybox:File_Name>                          <cybox:Hashes>
                        <cybox:Hash><cybox:Hash_Value Condition="Equals"
Datatype="BinHex">CB3DCDE34FD9FF0E19381D99B02F9692</cybox:Hash_Value>
                        <cybox:Type
Datatype="String">MD5</cybox:Type></cybox:Hash>
                </cybox:Hashes>
                <cybox:Size_In_Bytes Datatype="">23040</cybox:Size_In_Bytes>
        </cybox:Defined_Object>
        <cybox:Related_Objects>
                <cybox:Related_Object
Relationship="Contained_Within"><cybox:Object IDREF="b7e0bc39-f519-4878-8fb0-
5902554efe1c" Type="File"/></cybox:Related_Object>
        </cybox:Related_Objects>
</cybox:Object>
```

One pair of the C&C URLs/IPs

```
<cybox:Object ID="41b220d8-4c45-48de-9d08-30d661b2dc8e" Type="URI">
        <cybox:Defined_Object xsi:type="URIObj:URI_Object_Type" Type="URL">
                <URIObj:Value Datatype="URI"
Condition="Equals">www.documents.myPicture.info</URIObj:Value>
        </cybox:Defined_Object>
        <cybox:Related_Objects>
                <cybox:Related_Object
Relationship="Resolved_To"><cybox:Object IDREF="61aa225b-90ef-415c-8bbd-
a17282e457c9" Type="IP Address"/></cybox:Related_Object>
        </cybox:Related_Objects>
</cybox:Object>


<cybox:Object ID="61aa225b-90ef-415c-8bbd-a17282e457c9" Type="IP Address">
        <cybox:Defined_Object xsi:type="AddrObj:URI_Object_Type" category="ipv4-
addr">
                <AddrObj:Address_Value Datatype="String"
Condition="Equals">199.192.156.134</AddrObj:Address_Value>
        </cybox:Defined_Object>
</cybox:Object>
```

Trojan (us.exe) (with relationships)

```
<cybox:Object ID="b7e0bc39-f519-4878-8fb0-5902554efe1c" Type="File">

   <cybox:Description><cybox:Text>The file (us.exe MD5: FD1BE09E499E8E380424B3835FC973A8 4861440 bytes) is created in the logged in user %Temp% directory. The size of the embedded file is 22.5 KB (23040 bytes) and the size of the created us.exe is 4.63MB. It is an odd discrepancy until you look at the file and it looks like the code is repeated over and over - 211 times. The file resource section indicates the file is meant to look like a java updater, which is always larger than 22.5KB and that would explain all this padding, which is done at the time when the file is being written to the disk.</cybox:Text></cybox:Description>

   <cybox:Defined_Object xsi:type="FileObj:File_Object_Type">
                        <cybox:File_Name Datatype="String">us.exe</cybox:File_Name>
                        <cybox:File_Path Datatype="String">%Temp%</cybox:File_Path>
                        <cybox:Hashes><cybox:Hash><cybox:Hash_Value Condition="Equals"
Datatype="BinHex">FD1BE09E499E8E380424B3835FC973A8</cybox:Hash_Value><cybox:Type Datatype="String">MD5</cybox:Type></cybox:Hash></cybox:Hashes>
                        <cybox:Size_In_Bytes Datatype="">4861440</cybox:Size_In_Bytes>
   </cybox:Defined_Object>

   <cybox:Related_Objects>
                        <cybox:Related_Object Relationship="Created_By"><cybox:Object IDREF="8b463e0d-cc16-4036-950e-5eeb09bc51aa" Type="File"/></cybox:Related_Object>
                        <!-- The trojan connects to the following set of URLs/IPs for C&C -->
                        <cybox:Related_Object Relationship="Connected_To"><cybox:Object IDREF="41b220d8-4c45-48de-9d08-30d661b2dc8e" Type="URI"/></cybox:Related_Object>
                        <cybox:Related_Object Relationship="Connected_To"><cybox:Object IDREF="61aa225b-90ef-415c-8bbd-a17282e457c9" Type="IP
Address"/></cybox:Related_Object>
                        <cybox:Related_Object Relationship="Connected_To"><cybox:Object IDREF="568db11e-39ee-43d7-83d8-032bdec3801a" Type="URI"/></cybox:Related_Object>
                        <cybox:Related_Object Relationship="Connected_To"><cybox:Object IDREF="80bea4d1-0e70-4a03-a54f-e40373bf94f1" Type="IP
Address"/></cybox:Related_Object>
                        <cybox:Related_Object Relationship="Connected_To"><cybox:Object IDREF="af7cb3b6-d70b-4b3b-b24f-7cfad739710f" Type="URI"/></cybox:Related_Object>
                        <cybox:Related_Object Relationship="Connected_To"><cybox:Object IDREF="5ceb9d54-24e2-4627-948d-6b92ac81962a" Type="IP
Address"/></cybox:Related_Object>
   </cybox:Related_Objects>
</cybox:Object>
```

AND Composition of Objects

```
<cybox:Observable ID="47d6a950-884d-46b5-9938-ac5555065a81">

<!-- This composed observable defines a pattern that is true if the observed email exists AND the malicious .doc file exists AND the downloader .mp4 file exists AND the trojan .exe exists AND all three of the C&C IP addresses are seen-->

<!-- This yields a very tight filter that will have very low false positives but could miss almost any variation of the attack elements-->

        <cybox:Observable_Composition Operator="AND">
                <!-- "Iran-Oil" attack campaign email message with raw header-->
                <cybox:Observable IDREF="1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
                <!-- Iran-Oil corrupted .doc file-->
                <cybox:Observable IDREF="35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
                <!-- Iran-Oil invalid .mp4 downloader file-->
                <cybox:Observable IDREF="f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
                <!-- Iran-Oil .exe Trojan file-->
                <cybox:Observable IDREF="210f18f3-3874-4f9a-861d-71b328be90c6"/>
                <!-- The three Command and Control IPs-->
                <cybox:Observable IDREF="4e05804c-f552-44e1-9793-ff4bb7f88f9c"/>
                <cybox:Observable IDREF="1ea53b14-8fe9-467b-a298-62d9684e797d"/>
                <cybox:Observable IDREF="c78c0a83-6d14-45f8-827f-f758f0cd11ea"/>
        </cybox:Observable_Composition>
</cybox:Observable>
```

OR Composition of Objects

```
<cybox:Observable ID="94b0aa45-065e-486f-acaf-2d8e793f525e">

<!-- This composed observable defines a pattern that is true if the observed email exists OR the malicious .doc file exists OR the downloader .mp4 file exists OR the trojan .exe exists OR any of the three C&C IP addresses are seen-->

<!-- This yields a very loose filter that could have false positives but could catch numerous potential variations of the attack elements-->

        <cybox:Observable_Composition Operator="OR">
                <!-- "Iran-Oil" attack campaign email message with raw header-->
                <cybox:Observable IDREF="1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
                <!-- Iran-Oil corrupted .doc file-->
                <cybox:Observable IDREF="35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
                <!-- Iran-Oil invalid .mp4 downloader file-->
                <cybox:Observable IDREF="f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
                <!-- Iran-Oil .exe Trojan file-->
                <cybox:Observable IDREF="210f18f3-3874-4f9a-861d-71b328be90c6"/>
                <!-- The three Command and Control IPs-->
                <cybox:Observable IDREF="4e05804c-f552-44e1-9793-ff4bb7f88f9c"/>
                <cybox:Observable IDREF="1ea53b14-8fe9-467b-a298-62d9684e797d"/>
                <cybox:Observable IDREF="c78c0a83-6d14-45f8-827f-f758f0cd11ea"/>
        </cybox:Observable_Composition>
</cybox:Observable>
```

Receive Email Event

```
<cybox:Event Type="EmailOps">
        <cybox:Description>
                <cybox:Text>Receive "Iran-Oil" attack campaign email
message.</cybox:Text>
        </cybox:Description>
        <cybox:Actions>
                <cybox:Action Type="Receive">
                        <cybox:Associated_Objects>
                                <cybox:Associated_Object ID="51359587-f201-
4383-b032-5a64522fcd7d" Type="Email Message" AssociationType="Returned">

        <cybox:Defined_Object xsi:type="EmailMessageObj:Email_Message_Object_Type">
…

        </cybox:Defined_Object>
                                </cybox:Associated_Object>
                        </cybox:Associated_Objects>
                </cybox:Action>
        </cybox:Actions>
</cybox:Event>
```

Open .DOC file Event

```
<cybox:Event Type="FileOps">
        <cybox:Description>
                <cybox:Text>Open Iran-Oil corrupted .doc
file.</cybox:Text>
        </cybox:Description>
        <cybox:Actions>
                <cybox:Action Type="Access/Open">
                        <cybox:Associated_Objects>
                                <cybox:Associated_Object
ID="49d31c13-8d7b-4528-b8d6-ce8ed0d43ad7" Type="File"
AssociationType="Affected“/>
        </cybox:Associated_Objects>
                </cybox:Action>
        </cybox:Actions>
</cybox:Event>
```

Create Trojan .EXE Event

```
<cybox:Event Type="FileOps">
        <cybox:Description>
                <cybox:Text>Create Iran-Oil .exe Trojan
file.</cybox:Text>
        </cybox:Description>
        <cybox:Actions>
                <cybox:Action Type="Create">
                        <cybox:Associated_Objects>
                                <cybox:Associated_Object
IDREF="8b463e0d-cc16-4036-950e-5eeb09bc51aa" Type="File"
AssociationType="Initiating"/>
                                <cybox:Associated_Object
ID="b7e0bc39-f519-4878-8fb0-5902554efe1c" Type="File"
AssociationType="Affected“/>
                        </cybox:Associated_Objects>
                </cybox:Action>
        </cybox:Actions>
</cybox:Event>
```

C2 Connection Events

```
<cybox:Event Type="App Layer Traffic">
        <cybox:Description>
                <cybox:Text>Trojan .exe file connects out to C2 URLs/IPs.</cybox:Text>
        </cybox:Description>
        <cybox:Actions>
                <cybox:Action Type="Connect">
                        <cybox:Associated_Objects>
                                <cybox:Associated_Object IDREF="b7e0bc39-f519-4878-
8fb0-5902554efe1c" Type="File" AssociationType="Initiating"/>
                                <cybox:Associated_Object IDREF="41b220d8-4c45-48de-
9d08-30d661b2dc8e" Type="URI" AssociationType="Utilized"/>
                                <cybox:Associated_Object IDREF="61aa225b-90ef-415c-
8bbd-a17282e457c9" Type="IP Address" AssociationType="Utilized"/>
                                <cybox:Associated_Object IDREF="568db11e-39ee-43d7-
83d8-032bdec3801a" Type="URI" AssociationType="Utilized"/>
                                <cybox:Associated_Object IDREF="80bea4d1-0e70-4a03-
a54f-e40373bf94f1" Type="IP Address" AssociationType="Utilized"/>
                                <cybox:Associated_Object IDREF="af7cb3b6-d70b-4b3b-
b24f-7cfad739710f" Type="URI" AssociationType="Utilized"/>
                                <cybox:Associated_Object IDREF="5ceb9d54-24e2-4627-
948d-6b92ac81962a" Type="IP Address" AssociationType="Utilized"/>
                        </cybox:Associated_Objects>
                </cybox:Action>
        </cybox:Actions>
</cybox:Event>
```

Mixed Logic Composition

```
<cybox:Observable ID="7d932074-fded-4056-870e-dd51980501d4">

<!-- This composed observable defines a pattern that is true if (the receive email event occurs AND the create malicious
.doc file event occurs) OR (the download the downloader .mp4 file event occurs AND the create trojan .exe file event
occurs AND the execute trojan .exe file event occurs) OR the connect to all three of the C&C URLs/IPs event occurs-->

        <cybox:Observable_Composition Operator="OR">
                <cybox:Observable><cybox:Observable_Composition Operator="AND">
                        <!-- Receive "Iran-Oil" attack campaign email message -->
                        <cybox:Observable IDREF="1a937ec2-90ab-4e0e-a37c-db9b2e66a58e"/>
                        <!-- Open Iran-Oil corrupted .doc file-->
                        <cybox:Observable IDREF="35f04c28-5fd2-4d72-8aae-2ad04ee1811f"/>
                </cybox:Observable_Composition></cybox:Observable>
                <cybox:Observable><cybox:Observable_Composition Operator="AND">
                        <!-- Download Iran-Oil invalid .mp4 downloader file-->
                        <cybox:Observable IDREF="f005fbc6-7427-43ea-8e1e-9a341836f76b"/>
                        <!-- Create Iran-Oil .exe Trojan file-->
                        <cybox:Observable IDREF="210f18f3-3874-4f9a-861d-71b328be90c6"/>
                        <!-- Execute Iran-Oil .exe Trojan file-->
                        <cybox:Observable IDREF="b650c988-aac7-45ff-967d-9f1e5fc66161"/>
                </cybox:Observable_Composition></cybox:Observable>
                <!-- Trojan .exe file connects out to C&C URLs/IPs-->
                <cybox:Observable IDREF="a24ff8bc-b534-4616-838b-8bbe260a8e8f"/>
        </cybox:Observable_Composition>
</cybox:Observable>
```

- **This particular example is a good success example among existing sharing relationships but involved a great deal of extra effort due to lack of standardized structured representations**

Draft Structured Threat Information Architecture