Software Assurance Automation: Enabling Security and Resilience throughout the Software Lifecycle

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
Cyber Security & Communications
US Department of Homeland Security
Public/Private Collaboration Efforts for Security Automation and Software Supply Chain Risk Management

Next SwA Working Groups sessions: 27-29 Nov 2012 at MITRE, McLean, VA
Exploitable Software Weaknesses (CWEs) are sources for future Zero-Day Attacks.
Software Assurance Addresses Exploitable Software: Outcomes of non-secure practices and/or malicious intent

Exploitation potential of vulnerability is independent of “intent”

Software Assurance (SwA) is the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software throughout the life cycle.*

*Intentional vulnerabilities: spyware & malicious logic deliberately imbedded (might not be considered defects)

‘High quality’ can reduce security flaws attributable to defects; yet traditional S/W quality assurance does not address intentional malicious behavior in software

From CNSS Instruction 4009 “National Information Assurance Glossary” (26APR2010)
Challenges in Mitigating Risks Attributable to Exploitable Software and Supply Chains

Several needs arise:

- Need internationally recognized standards to support security automation and processes to provide transparency for more informed decision-making for mitigating enterprise risks.
- Need ‘Assurance’ to be explicitly addressed in standards & capability benchmarking models for organizations involved with security/safety-critical applications.
- Need more comprehensive diagnostic capabilities to provide sufficient evidence that “code behavior” can be well understood to not possess exploitable or malicious constructs.
- Need rating schemes for software products and supplier capabilities.
Enterprises seek comprehensive capabilities to:

- Avoid installing software with MALWARE pre-installed.

- Determine that no publicly reported VULNERABILITIES remain in code prior to operational acceptance, and that future discoveries of common vulnerabilities and exposures can be quickly patched.

- Determine that exploitable software WEAKNESSES that put the users most at risk are mitigated prior to operational acceptance.
Challenges in Preventing and Responding to Cyber Incidents

- “Silos” of operation
- Proprietary reporting formats

Needs arise:
- Need standards to support security automation and processes to support exchange of information and cyber indicators relative to incident management and response.

Software Assurance Forum & Working Groups*
Homeland Security

Cybersecurity and Communications

DHS CS&C Software Assurance (SwA) Program

Advances security and resilience of software throughout the lifecycle; focuses on reducing exploitable software weaknesses and addresses means to improve capabilities that routinely develop, acquire, and deploy resilient software.

- Serves as a focal point for interagency public-private collaboration to enhance development and acquisition processes, capability benchmarking and rating schemes to address software security needs.
  - Hosts interagency Software Assurance Forums, working groups and training to provide public-private collaboration in advancing software security and providing publicly available resources.
  - Provides collaboratively developed, peer-reviewed information resources on Software Assurance, via journals, guides & on-line resources suitable for use in education, training, and process improvement.
  - Provides input and criteria for leveraging international standards and maturity models used for process improvement and capability benchmarking of software suppliers and acquisition organizations.

- Enables software security automation and measurement capabilities through use of common indexing and reporting capabilities for malware, exploitable software weaknesses, cyber indicators and attacks which target software.
  - Collaborates with national & international standards organizations and industry to create standards, metrics and certification mechanisms from which products and tools could be qualified for software security verification.
  - Manages programs for Malware Attribute Enumeration Classification (MAEC), Common Weakness Enumeration (CWE), Common Attack Pattern (CAPEC) & Cyber Observable eXpression (CybOX).
  - Manages programs for Common Vulnerabilities & Exposures (CVE) and Open Vulnerability & Assessment Language (OVAL) that provide information feeds for continuous monitoring, security content automation, vulnerability databases, and security/threat alerts from many organizations.
Software Assurance

Software Assurance (SwA) is the level of confidence that software functions as intended and is free from vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software throughout the life cycle.*

Derived From: CNSSI-4009

Automation

Languages, enumerations, registries, tools, and repositories throughout the Lifecycle

Including design, coding, testing, deployment, configuration and operation
Automation is *one piece* of the SwA puzzle.
Many DHS, DoD, and NIST sponsored efforts are key to changing how software-based systems are developed, deployed & operated securely. These are (or are becoming used in) international standards.
Making Security Measurable (MSM): You Are Here

- Software Assurance
- Enterprise Security Management
- Threat Management

Design → Deploy → Build → Assess → Test → Deploy

Vulnerabilities: CVE, CWE, CAPEC, MAEC, CybOX, IODEF
Exploits: CWE, CAPEC, MAEC, CybOX, IODEF
Attacks: CWE, CAPEC, MAEC, CybOX, IODEF
Malware: CWE, CAPEC, MAEC, CybOX, IODEF

CWE, CAPEC, CWSS, CWRAF
CPE, CCE, OVAL, OCIL, XCCDF, AssetId, ARF
CWE, CAPEC, MAEC, CybOX, IODEF
Cyber Threats Emerged Over Time

1980's
- Password guessing
- Exploiting known vulnerabilities
- Burglaries
- Packet spoofing

1990's
- GUI intruder tools
- Hijacking sessions
- Back doors
- Internet social engineering attacks
- Password cracking

2000's
- Executable code attacks (against browsers)
- Automated widespread attacks
- Automated probes/scans
- Email propagation of malicious code
- Network management diagnostics

2010's
- www attacks
- Diffuse spyware
- Increase in wide-scale Trojan horse distribution
- Increase in tailored worms
- Sophisticated command & control
- Anti-forensic techniques
- Home users targeted
- Distributed attack tools
- Techniques to analyze code for vulnerabilities without source code
Solutions Also Emerged Over Time

- **1980’s**
  - hijacking sessions
  - disabling audits
  - Internet social engineering (attacks)
  - password guessing
  - exploiting known vulnerabilities
  - packet spoofing

- **1990’s**
  - GUI intruder tools
  - executable code attacks (against browsers)
  - automated widespread attacks
  - hijacking sessions
  - password cracking

- **2000’s**
  - automated probes/scans
  - widespread attacks using NNTP to distribute attack
  - widespread attacks on DNS infrastructure
  - sniffers
  - automated widespread attacks
  - anti-forensic techniques
  - Internet social engineering attacks
  - widespread denial-of-service attacks

- **2010’s**
  - automated probes/scans
  - email propagation of malicious code
  - DDoS attacks
  - binary encryption
  - increase in tailor-made worms
  - sophisticated command & control
  - diffuse spyware
  - increase in wide-scale Trojan horse distribution
  - Windows-based remote controllable Trojans (Back Orifice)
  - techniques to analyze code for vulnerabilities without source code
  - threats to home users targeted

- **2010’s**
  - distributed attack tools
  - techniques to analyze code for vulnerabilities without source code
  - Windows-based remote controllable Trojans (Back Orifice)
  - increase in wide-scale Trojan horse distribution
  - diffuse spyware
  - distributed attack tools
  - home users targeted
  - sophisticated command & control

**Attack Sophistication**

**Solutions Also Emerged Over Time**

- **1980’s**
  - password guessing
  - hijacking sessions
  - disabling audits
  - Internet social engineering (attacks)
  - password cracking

- **1990’s**
  - GUI intruder tools
  - executable code attacks (against browsers)
  - automated widespread attacks
  - hijacking sessions
  - password cracking

- **2000’s**
  - automated probes/scans
  - widespread attacks using NNTP to distribute attack
  - widespread attacks on DNS infrastructure
  - sniffers
  - automated widespread attacks
  - anti-forensic techniques
  - Internet social engineering attacks
  - widespread denial-of-service attacks

- **2010’s**
  - automated probes/scans
  - email propagation of malicious code
  - DDoS attacks
  - binary encryption
  - increase in tailor-made worms
  - sophisticated command & control
  - diffuse spyware
  - increase in wide-scale Trojan horse distribution
  - Windows-based remote controllable Trojans (Back Orifice)
  - techniques to analyze code for vulnerabilities without source code
  - threats to home users targeted

- **2010’s**
  - distributed attack tools
  - techniques to analyze code for vulnerabilities without source code
  - Windows-based remote controllable Trojans (Back Orifice)
  - increase in wide-scale Trojan horse distribution
  - diffuse spyware
  - distributed attack tools
  - home users targeted
  - sophisticated command & control
Architecting Security with Information Standards for COIs

- Asset Management
- Vulnerability Management
- Configuration Management
- Threat Management
- System Development
- System Certification
- Intrusion Detection
- Incident Management
- Change Management
- Trust Management
- Identity Management
- Central Reporting
Asset Management  
Vulnerability Management  
Configuration Management  
Threat Management  
System Development  
System Certification  
Intrusion Detection  
Incident Management  
Change Management  
Trust Management  
Identity Management  
Central Reporting
Operational Enterprise Networks

Development & Sustainment Security Management Processes

- Asset Inventory
- Configuration Guidance Analysis
- Vulnerability Analysis
- Threat Analysis
- Intrusion Detection
- Incident Management

Operations Security Management Processes

- Assessment of System Development, Integration, & Sustainment Activities and Certification & Accreditation

Operational Enterprise Networks

- Trust Management
- Enterprise IT Change Management
- Identity Management
- Centralized Reporting

Enterprise IT Asset Management

- Asset Inventory
- Configuration Guidance Analysis
- Vulnerability Analysis
- Threat Analysis
- Intrusion Detection
- Incident Management

Operations Security Management Processes

- Assessment of System Development, Integration, & Sustainment Activities and Certification & Accreditation

Operational Enterprise Networks

- Trust Management
- Enterprise IT Change Management
- Identity Management
- Centralized Reporting

Enterprise IT Asset Management
## Cyber Ecosystem Standardization Efforts

<table>
<thead>
<tr>
<th>Question</th>
<th>Standard/Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>What IT systems do I have in my enterprise?</td>
<td>CPE (Platforms)</td>
</tr>
<tr>
<td>What known vulnerabilities do I need to worry about?</td>
<td>CVE (Vulnerabilities)</td>
</tr>
<tr>
<td>What vulnerabilities do I need to worry about right now?</td>
<td>CVSS (Scoring System)</td>
</tr>
<tr>
<td>How can I configure my systems more securely?</td>
<td>CCE (Configurations)</td>
</tr>
<tr>
<td>How do I define a policy of secure configurations?</td>
<td>XCCDF (Configuration Checklists)</td>
</tr>
<tr>
<td>How can I be sure my systems conform to policy?</td>
<td>OVAL (Assessment Language)</td>
</tr>
<tr>
<td>How can I be sure the operation of my systems conforms to policy?</td>
<td>OCIL (Interactive Language)</td>
</tr>
<tr>
<td>What weaknesses in my software could be exploited?</td>
<td>CWE (Weaknesses)</td>
</tr>
<tr>
<td>What attacks can exploit which weaknesses?</td>
<td>CAPEC (Attack Patterns)</td>
</tr>
<tr>
<td>How can we recognize malware &amp; share that info?</td>
<td>MAEC (Malware Attributes)</td>
</tr>
<tr>
<td>What observable behavior might put my enterprise at risk?</td>
<td>CybOX (Cyber Observables)</td>
</tr>
<tr>
<td>What events should be logged, and how?</td>
<td>CEE (Events)</td>
</tr>
<tr>
<td>How can I aggregate assessment results?</td>
<td>ARF (Assessment Results)</td>
</tr>
<tr>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>What IT systems do I have in my enterprise?</td>
<td>CPE (Platforms)</td>
</tr>
<tr>
<td>What known vulnerabilities do I need to worry about?</td>
<td>CVE (Vulnerabilities)</td>
</tr>
<tr>
<td>What vulnerabilities do I need to worry about right now?</td>
<td>CVSS (Scoring System)</td>
</tr>
<tr>
<td>How can I configure my systems more securely?</td>
<td>CCE (Configurations)</td>
</tr>
<tr>
<td>How do I define a policy of secure configurations?</td>
<td>XCCDF (Configuration Checklists)</td>
</tr>
<tr>
<td>How can I be sure my systems conform to policy?</td>
<td>OVAL (Assessment Language)</td>
</tr>
<tr>
<td>How can I be sure the operation of my systems conforms to policy?</td>
<td>OCIL (Interactive Language)</td>
</tr>
<tr>
<td>What weaknesses in my software could be exploited?</td>
<td>CWE (Weaknesses)</td>
</tr>
<tr>
<td>What attacks can exploit which weaknesses?</td>
<td>CAPEC (Attack Patterns)</td>
</tr>
<tr>
<td>How can we recognize malware &amp; share that info?</td>
<td>MAEC (Malware Attributes)</td>
</tr>
<tr>
<td>What observable behavior might put my enterprise at risk?</td>
<td>CybOX (Cyber Observables)</td>
</tr>
<tr>
<td>What events should be logged, and how?</td>
<td>CEE (Events)</td>
</tr>
<tr>
<td>How can I aggregate assessment results?</td>
<td>ARF (Assessment Results)</td>
</tr>
</tbody>
</table>

Standardization Efforts leveraged by the Security Content Automation Protocol (SCAP)
Efforts focused on mitigating risks and enabling more robust continuous monitoring and faster incident response

<table>
<thead>
<tr>
<th>Question</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>What IT systems do I have in my enterprise?</td>
<td>CPE (Platforms)</td>
</tr>
<tr>
<td>What known vulnerabilities do I need to worry about?</td>
<td>CVE (Vulnerabilities)</td>
</tr>
<tr>
<td>What vulnerabilities do I need to worry about right now?</td>
<td>CVSS (Scoring System)</td>
</tr>
<tr>
<td>How can I configure my systems more securely?</td>
<td>CCE (Configurations)</td>
</tr>
<tr>
<td>How do I define a policy of secure configurations?</td>
<td>XCCDF (Configuration Checklists)</td>
</tr>
<tr>
<td>How can I be sure my systems conform to policy?</td>
<td>OVAL (Assessment Language)</td>
</tr>
<tr>
<td>How can I be sure the operation of my systems conforms to policy?</td>
<td>OCIL (Interactive Language)</td>
</tr>
<tr>
<td>What weaknesses in my software could be exploited?</td>
<td>CWE (Weaknesses)</td>
</tr>
<tr>
<td>What attacks can exploit which weaknesses?</td>
<td>CAPEC (Attack Patterns)</td>
</tr>
<tr>
<td>How can we recognize malware &amp; share that info?</td>
<td>MAEC (Malware Attributes)</td>
</tr>
<tr>
<td>What observable behavior might put my enterprise at risk?</td>
<td>CybOX (Cyber Observables)</td>
</tr>
<tr>
<td>What events should be logged, and how?</td>
<td>CEE (Events)</td>
</tr>
<tr>
<td>How can I aggregate assessment results?</td>
<td>ARF (Assessment Results)</td>
</tr>
</tbody>
</table>
Evolution of Standardized Representations - Sharing

Vulnerabilities
Weaknesses
Attack Patterns
Malware Behavior
Cyber Observables
Threat Indicators

Imports & Extends:
- Object
- Defined Objects
- Actions

MAEC
CAPEC
CybOX

Malware
Log Events
Attack Patterns
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 standardized language for encoding and communicating information about cyber observables ([http://cybox.mitre.org](http://cybox.mitre.org))
What is STIX

Structured Threat Information eXpression

Language
Specify  Capture  Characterize  Communicate

Cyber Threat Information
Community-driven
Consistency  Clarity  Support automation
Structuring Threat Information for Sharing

Why were they doing it?

Why should you care about it?

What exactly were they doing?

What you are looking for

Who was doing it?

What were they looking to exploit?

Where was it seen?

What should you do about it?
• Org C must understand *each* format in use and try to map across formats – sacrificing time and potentially losing information
• Duplication of effort at each organization in the exchange is expensive and does not scale
Enabling Cross-Vendor Sharing

• Org C only needs to understand one format – no need to map and no information loss

• Each vendor maps their internal representations to the common format *once* – efficient and scalable
# CybOX Tool Roadmap

<table>
<thead>
<tr>
<th>Tool</th>
<th>Class</th>
<th>Language</th>
<th>Availability</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenIOC -&gt; CybOX</td>
<td>Translator</td>
<td>Python</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>CybOX -&gt; OpenIOC</td>
<td>Translator</td>
<td>Python</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>Snort -&gt; CybOX</td>
<td>Translator</td>
<td>Python</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>Email -&gt; CybOX</td>
<td>Transform</td>
<td>Python</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>CybOX -&gt; Snort</td>
<td>Translator</td>
<td>Python</td>
<td>11/12</td>
<td>New BSD</td>
</tr>
<tr>
<td>IODEF -&gt; CybOX</td>
<td>Translator</td>
<td>Python</td>
<td>11/12</td>
<td>New BSD</td>
</tr>
<tr>
<td>CybOX -&gt; Suricata</td>
<td>Translator</td>
<td>Python</td>
<td>12/12</td>
<td>New BSD</td>
</tr>
<tr>
<td>Suricata -&gt; CybOX</td>
<td>Translator</td>
<td>Python</td>
<td>12/12</td>
<td>New BSD</td>
</tr>
<tr>
<td>CybOX Python Bindings</td>
<td>Bindings</td>
<td>Python</td>
<td>Now</td>
<td>New BSD</td>
</tr>
</tbody>
</table>

BSD license is a very open license, allowing you to do practically anything with the software. It’s less restrictive than the GPL, but more restrictive than the Public Domain. There are only a couple precepts that must be adhered to, when using this license:

- You are free to redistribute the software, in binary or source form, as long as the copyright, conditions and disclaimer are present.
- You cannot use the name of originating organization or contributors to promote derivatives of the software, without written consent.

If adhered to, you are free to modify, copy and redistribute BSD-licensed software in either source or binary form as you see fit. You are not required to return code or patches to the upstream BSD-licensed software. You are free to change the license, or charge for derivatives, of the software, be it commercial or proprietary.
<table>
<thead>
<tr>
<th>Tool</th>
<th>Class</th>
<th>Language</th>
<th>Current Support</th>
<th>Avail.</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAEC/CybOX Python Bindings</td>
<td>Bindings</td>
<td>Python</td>
<td>MAEC v2.1/CybOX 1.0</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>MAEC -&gt; OVAL</td>
<td>Translator</td>
<td>Python</td>
<td>MAEC v2.1</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>Anubis -&gt; MAEC</td>
<td>Translator</td>
<td>Python</td>
<td>MAEC v2.1/CybOX 1.0</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>ThreatExpert -&gt; MAEC</td>
<td>Translator</td>
<td>Python</td>
<td>MAEC v1.1</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>GFI Sandbox -&gt; MAEC</td>
<td>Translator</td>
<td>Python</td>
<td>MAEC v2.1</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>FireEye -&gt; MAEC</td>
<td>Translator</td>
<td>Python</td>
<td>n/a -- in develop.</td>
<td>TBD</td>
<td>New BSD</td>
</tr>
<tr>
<td>Norman Sandbox -&gt; MAEC</td>
<td>Translator</td>
<td>Python</td>
<td>n/a -- in develop.</td>
<td>TBD</td>
<td>New BSD</td>
</tr>
<tr>
<td>PEFile -&gt; MAEC</td>
<td>Translator</td>
<td>Python</td>
<td>n/a -- in develop.</td>
<td>10/12</td>
<td>New BSD</td>
</tr>
<tr>
<td>MAEC -&gt; Suricata</td>
<td>Translator</td>
<td>Python</td>
<td>n/a -- in develop.</td>
<td>12/12</td>
<td>New BSD</td>
</tr>
<tr>
<td>MAEC -&gt; HTML</td>
<td>Translator</td>
<td>XSLT</td>
<td>MAEC v2.1</td>
<td>Now</td>
<td>New BSD</td>
</tr>
<tr>
<td>MAEC Comparator</td>
<td>Analysis</td>
<td>Python</td>
<td>MAEC v2.1</td>
<td>Now</td>
<td>New BSD</td>
</tr>
</tbody>
</table>
Suricata Translators

• **MAEC -> Suricata**
  – Provides an automated way of creating complex signatures for HTTP traffic (and others) based on MAEC’s characterization of malware behavior

• **CybOX -> Suricata**
  – Allows for the creation of Suricata rules based on abstract cyber observables, including those exchanged in threat representations like STIX

• **Suricata -> CybOX**
  – Facilitates sharing and better understanding of rules via transformation into a standard representation
SwA Working Group Sessions: 27-29 Nov 2012 @ MITRE in McLean, VA

SwA Forum – Next session: 5-7 Mar 2013 @ NIST in Gaithersburg, MD

SwA Websites: www.us-cert.gov/swa

Email: software.assurance@dhs.gov

Making Security Measureable: measurablesecurity.mitre.org

See Language for sharing exchange of indicators and correlation of incident information -- Cyber Observables eXpression (CybOX) at http://cybox.mitre.org
Next SwA Working Groups sessions: 27-29 Nov 2012 at MITRE, McLean, VA
Next SwA Working Groups at MITRE, McLean, VA – 27-29 2012

SOFTWARE ASSURANCE FORUM

“Building Security In”
https://buildsecurityin.us-cert.gov/swa

Joe Jarzombek, PMP, CSSLP
Director for Software Assurance
Cyber Security & Communications
Department of Homeland Security
Joe.Jarzombek@hq.dhs.gov
(703) 235-3673