MITRE ICS/SCADA Cyber Repository

Jackson Wynn, MITRE Corporation

International Atomic Energy Agency (IAEA) Research Coordination Meeting (RCM)

Erlangen, Germany
20-24 March 2017

Approved for Public Release; Distribution Unlimited: 17-0876.
Abstract

- MITRE CRP deliverables for FY17 include an open source catalog of CAPEC-like attack patterns specific to ICS/SCADA systems
  - Providing an extensible taxonomy for organizing ICS/SCADA attack patterns that promotes alternative search strategies

- MITRE now hosts an open source catalog called TARA in its corporate DMZ
  - This presentation discusses the catalog capability, its data model, and a MITRE-developed cyber risk assessment methodology that the catalog tool supports
Agenda

- TARA Catalog Tool
- Data Model Details
  - Vector Groups / Taxonomies
  - Attack Vectors
  - Countermeasures
  - Countermeasure Mappings
- Catalog Tool Demo
- Catalog Data Sources
- Threat Assessment & Remediation Analysis (TARA)
  - Methodology Description
Threat Assessment and Remediation Analysis (TARA) Catalog Tool

- Web-based capability used to compile and search for information about cyber attacks and countermeasures
  - Developed to support cyber risk assessments that apply MITRE-developed TARA methodology
Uses of the TARA Catalog Tool

- Casual browsing
- Compilation of attack vector and countermeasure information
- Taxonomy development
- Threat model development
TARA Catalog usage for the CRP

- The TARA catalog will support MITRE/University of Massachusetts Lowell (UML) IAEA research
  - Compilation of ICS/SCADA attack vectors and countermeasures
  - Development of ICS/SCADA cyber threat taxonomies
  - Development of cyber threat models of hypothetical nuclear facilities

- Read-only access to the catalog can be provided to IAEA Collaborative Research Program (CRP) participants
  - Emails will be sent to CRP participants with details on accounts and access
  - A catalog user guide is currently in development
TARA Data Model
Objectives of the TARA Catalog

- Provides a repository of Attack Vector (AV) and Countermeasure (CM) data used in TARA assessments
- Serves as a collection point for data derived from variety of sources
- Supports mappings and groupings that can be used to connect and traverse catalog data

Understanding the data model makes it easier to use the TARA catalog tool
Vector Groups and Taxonomies

**Vector Group** – Named collection of attack vectors

**Taxonomy** – Hierarchically structured collection of vector groups

```
<table>
<thead>
<tr>
<th>Select</th>
<th>VG ID</th>
<th>Children</th>
<th>Vector Group</th>
<th>Description</th>
<th>Type</th>
<th>Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A000422</td>
<td>10</td>
<td>ATT&amp;CK</td>
<td>Adversarial Tactics, Techniques, and Common Knowledge (ATT&amp;CK™) is a framework for describing post-compromise adversary behavior within an enterprise network.</td>
<td>Root</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>A000387</td>
<td>16</td>
<td>CAPEC</td>
<td>Common Attack Pattern Enumeration and Classification (CAPEC™) provides a publicly available catalog of common attack patterns.</td>
<td>Root</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>A000384</td>
<td></td>
<td>CM Practices</td>
<td>Groups of Countermeasures (CMs)</td>
<td>Root</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td>A000493</td>
<td>3</td>
<td>ICS/SCADA System</td>
<td>Organizational taxonomy representing ICS/SCADA Systems</td>
<td>Root</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A000471</td>
<td>4</td>
<td>IP System</td>
<td>Organizational taxonomy representing IP-based, distributed systems</td>
<td>Root</td>
<td></td>
</tr>
</tbody>
</table>
```

“Root” indicates Taxonomy
Vector Group Example: Software (Top)
Vector Group Example: Software (Bottom)

<table>
<thead>
<tr>
<th>AV ID</th>
<th>Name</th>
<th>Confidential</th>
<th>Integrity</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>T000049</td>
<td>Buffer Overflow</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000024</td>
<td>Malicious Software Update</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000041</td>
<td>Exploit race conditions and/or deadlock conditions in software</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000026</td>
<td>Accessing Functionality Not Properly Constrained by ACLs</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000027</td>
<td>Accessing, modifying or executing executable files</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000028</td>
<td>Manipulating User-Controlled Variables</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000038</td>
<td>Manipulation of resources loaded by a software application</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000055</td>
<td>Target Programs with Elevated Privileges</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000027</td>
<td>Manipulating Input to File System Calls</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000152</td>
<td>Read Sensitive Strings Within an Executable</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000188</td>
<td>Unauthorized / unrestricted copying</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000192</td>
<td>Counterfeit web sites used to distribute malicious software updates</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000181</td>
<td>Malicious software implantation through 3rd party bundling</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000182</td>
<td>Software defects hidden/obscured by code complexity</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>T000208</td>
<td>User exploits vulnerability to gain unauthorized or privileged access</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Links to catalog attack vectors associated with the Software vector group

Attack vectors listed are in no particular order
Taxonomy Example: *IP System*

![Diagram of IP System taxonomy example]

- **IP System**
  - Computer
    - BIOS
  - IP Network
  - SDLC
    - Security
      - Capability
  - Server
    - Database
    - Web Server
  - Client
    - Desktop
    - Web App
    - Web Service
- **Software Vector Group**
  - API
  - Malware
  - OS
  - VM
  - Web 2.0
  - XML

© 2017 The MITRE Corporation. All Rights Reserved. 
Approved for Public Release
Attack Vectors (AVs)

A sequence of steps performed by an adversary in the course of conducting a cyber attack

- **Sources of Attack Vector data**
  - Common Attack Pattern Enumeration and Classification (CAPEC)
  - Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™)
  - Common Weakness Enumeration (CWE)
  - Common Vulnerabilities and Exposures (CVE)
  - ICS-CERT Advisories

- All attack vector data derived from public domain sources
### Attack Vector Example: Stuxnet (Top)

<table>
<thead>
<tr>
<th>Threat ID:</th>
<th>Threat Vector Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0988321</td>
<td>Malware infection of ICS/SCADA equipment</td>
</tr>
</tbody>
</table>

#### Description:
Stuxnet includes a highly specialized malware payload that is designed to target Siemens supervisory control and data acquisition (SCADA) systems that are configured to control and monitor specific industrial processes. Stuxnet infects PLCs by subverting the Step-7 software application used to reprogram those devices.

#### References:
Attack Vector Example: Stuxnet (Bottom)

Links to catalog countermeasures

Links to catalog vector groups
Countermeasures (CMs)

“Actions, devices, procedures, or techniques that meet or oppose (i.e., counters) a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken.”
Source: CNSS 4009

- **Sources of Countermeasure data**
  - Common Attack Pattern Enumeration and Classification (CAPEC)
  - Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™)
  - Common Weakness Enumeration (CWE)
  - Common Vulnerabilities and Exposures (CVE)
  - ICS-CERT Advisories
  - DoD and NIST publications
  - Industry recognized security best practices

All countermeasure data derived from public domain sources
Countermeasure Example: Patch Management (Top)
## Countermeasure Example: Patch Management (Bottom)

![Image of MAE Tools interface showing threat vectors and patch management]

**Links to associated catalog Attack Vectors**

<table>
<thead>
<tr>
<th>Threat Vector ID</th>
<th>Name</th>
<th>Prevent</th>
<th>Detect</th>
<th>Respond</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>T000259</td>
<td>Router DoS using TCP protocol messaging</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000260</td>
<td>Router DoS using malformed IP packets</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000261</td>
<td>Cisco IOS Software TCP Denial of Service Vulnerability</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000262</td>
<td>Bypass router login</td>
<td>N/A</td>
<td>Medium</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000263</td>
<td>Unauthorized access via router CLI</td>
<td>N/A</td>
<td>Medium</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000264</td>
<td>Router authentication bypass</td>
<td>N/A</td>
<td>Medium</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000265</td>
<td>Router DoS using spoofed router access</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000272</td>
<td>Unauthorized router telnet access</td>
<td>N/A</td>
<td>Medium</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000275</td>
<td>Router DoS using crafted IP packets</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000276</td>
<td>Router DoS using crafted HTTP protocol messaging</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000277</td>
<td>Router DoS using ICMP protocol messaging</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000278</td>
<td>Router DoS using malformed ARP messaging</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000310</td>
<td>Router DoS using OSPF vulnerability</td>
<td>Medium</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000312</td>
<td>Software assurance practices</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000314</td>
<td>Supply Chain practices</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Unclassified</td>
</tr>
<tr>
<td>T000167</td>
<td>IDS/IPS not configured to detect adversary reconnaissance or penetration attempts</td>
<td>N/A</td>
<td>N/A</td>
<td>Medium</td>
<td>Unclassified</td>
</tr>
</tbody>
</table>
Countermeasure Categories

The countermeasure taxonomy provides a list of countermeasure categories:

- Each category contains 20-40 related countermeasures

<table>
<thead>
<tr>
<th>VG ID</th>
<th>Children</th>
<th>Vector Group</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A000422</td>
<td>10</td>
<td>ATT&amp;CK</td>
<td>Adversarial Tactics, Techniques, and Common Knowledge (ATT&amp;CK™) is a framework for describing post-compromise adversary behavior within an enterprise network.</td>
<td>Root</td>
</tr>
<tr>
<td>A000387</td>
<td>16</td>
<td>CAPEC</td>
<td>Common Attack Pattern Enumeration and Classification (CAPEC™) provides a publicly available catalog of common attack patterns.</td>
<td>Root</td>
</tr>
<tr>
<td>A000384</td>
<td></td>
<td>CM Practices</td>
<td>Groups of Countermeasures (CMs)</td>
<td>Root</td>
</tr>
<tr>
<td>A000493</td>
<td>2</td>
<td>ICS/SCADA System</td>
<td>Organizational taxonomy representing ICS/SCADA Systems</td>
<td>Root</td>
</tr>
<tr>
<td>A000495</td>
<td>2</td>
<td>Indicators</td>
<td>Organizational taxonomy of Indicators of Compromise (IOCs)</td>
<td>Root</td>
</tr>
<tr>
<td>A000471</td>
<td>4</td>
<td>IP System</td>
<td>Organizational taxonomy representing IP-based, distributed systems</td>
<td>Root</td>
</tr>
</tbody>
</table>

### AV ID - Name

- T000281 - Resiliency practices
- T000311 - Network security practices
- T000312 - Software assurance practices
- T000313 - Architecture/design practices
- T000314 - Supply Chain practices
- T000315 - Insider threat practices
- T000320 - Top 20 countermeasures
Countermeasure Mappings

- Represents the effect a countermeasure has on an attack vector
  - Range of countermeasure effects
    - Detect (denoted by a ‘D’)
      - The countermeasure makes it possible to determine if the attack has occurred, is occurring, or potentially could occur
        - Examples: Intrusion Detection Systems (IDS), continuous monitoring, etc.
    - Prevent (denoted by a ‘P’)
      - The countermeasure partially or completely eliminates conditions that make the attack possible
        - Examples: network segmentation, cyber threat awareness training, etc.
    - Respond (denoted by a ‘R’)
      - The countermeasure reduces the likelihood that the attack will occur or that its impact will be significant
        - Examples: System restoration from backup, maintaining a cyber playbook, forensic analysis of compromised systems, etc.
Mitigation Mappings Table

A mitigation mapping table conveys the effects that a range of countermeasures has over a range of attack vectors

- Attack vectors represented as columns in the mapping table
- Countermeasures represented as rows in the mapping table
- Matrix cells can be used to identify what effect \{Prevent, Respond, Detect\} a countermeasure has on an attack vector

<table>
<thead>
<tr>
<th>Countermeasures</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>R</td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C6</td>
<td></td>
<td></td>
<td>P</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C7</td>
<td></td>
<td></td>
<td>P</td>
<td></td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C8</td>
<td></td>
<td></td>
<td>R</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C9</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Coverage gap
- Superfluous countermeasure
- Countermeasure C4 has a responsive effect on Attack vectors A3 and A6

Mitigation Mappings Table
Effect Confidence

- Assesses the certainty that a given effect will be realized
  - High (denoted by ‘H’)
    - Engineering verification confirms the effect, i.e., demonstration, inspection, testing, or analysis
  - Moderate (denoted by ‘M’)
    - Mapping based on Subject Matter Expert (SME) judgment
  - Low (denoted by ‘L’)
    - Plausible effect that has not yet been confirmed or substantiated
## Example Mitigation Mappings Table

<table>
<thead>
<tr>
<th>Countermeasure (CM)</th>
<th>Effect (by Attack Vector ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CM ID</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>C000103</td>
<td>Match buffer size to data input size</td>
</tr>
<tr>
<td>C000293</td>
<td>Disable file and printer sharing</td>
</tr>
<tr>
<td>C000134</td>
<td>Select programming languages that minimize potential software defects</td>
</tr>
<tr>
<td>C000238</td>
<td>Enforce software quality standards and guidelines that improve software quality</td>
</tr>
<tr>
<td>C000117</td>
<td>Apply principle of least privilege</td>
</tr>
<tr>
<td>C000135</td>
<td>Avoid use of dangerous memory functions and operations</td>
</tr>
<tr>
<td>C000039</td>
<td>Convert input data into the data format in which it is used</td>
</tr>
<tr>
<td>C000059</td>
<td>Enable use of the HTTP Referrer header field</td>
</tr>
<tr>
<td>C000093</td>
<td>Merge data streams prior to validation</td>
</tr>
<tr>
<td>C000096</td>
<td>Use vetted runtime libraries</td>
</tr>
<tr>
<td>C000123</td>
<td>Design software to fail securely</td>
</tr>
<tr>
<td>C000136</td>
<td>Utilize processor-based protection capabilities</td>
</tr>
<tr>
<td>C000045</td>
<td>Utilize high quality session IDs</td>
</tr>
<tr>
<td>C000047</td>
<td>Encrypt session cookies</td>
</tr>
<tr>
<td>C000051</td>
<td>Use digital signatures/checksums to authenticate source of changes</td>
</tr>
<tr>
<td>C000089</td>
<td>Validate the range of numeric input</td>
</tr>
<tr>
<td>C000095</td>
<td>Convert input to canonical form before validating</td>
</tr>
<tr>
<td>C000101</td>
<td>Verify buffer sizes</td>
</tr>
<tr>
<td>C000102</td>
<td>Verify message size data</td>
</tr>
<tr>
<td>C000137</td>
<td>Use unsigned variables to represent whole numbers</td>
</tr>
<tr>
<td>C000094</td>
<td>Validate data exchanges across language boundaries</td>
</tr>
<tr>
<td>C000132</td>
<td>Use sandboxing to isolate running software</td>
</tr>
<tr>
<td>C000146</td>
<td>Apply transport-level mechanisms such as TLS and or VPNs to protect sensitive content</td>
</tr>
</tbody>
</table>

### Mapping Table

**Effects (P, R, D) x Confidence (H, M, L):**

{PH, PM, PL, RH, RM, RL, DH, DM, DL}
Tools Demo

Catalog Search Tools

Catalog Update Tools

Threat Assessment & Remediation Analysis (TARA)
Sources of Catalog Data
Common Attack Pattern Enumeration and Classification (CAPEC)

- **MITRE open source repository of cyber attack patterns**
  - Includes postulated attacks and real world security incidents
  - DHS-hosted, Community-contributed, MITRE-moderated
  - Updated quarterly

- **CAPEC includes over 450 attack patterns**
  - Attack patterns contributed by the security research community at large, subject to MITRE review for quality and completeness
  - Patterns conform to XML schema and include fields that characterize the sophistication and resources required
    - CAPEC patterns provide analysis of underlying design weaknesses, which is key to follow-on mitigation engineering activities
CAPEC Taxonomy: Mechanisms of Attack

http://capec.mitre.org/
Example CAPEC Attack Pattern

CAPEC-100: Overflow Buffers

Attacker Pattern ID: 100
Abstraction: Standard

Presentation Filter: Basic

Summary
Buffer Overflow attacks target improper or missing bounds checking on buffer operations, typically triggered by input injected by an attacker. As a consequence, an attacker is able to write past the boundaries of allocated buffer regions in memory, causing a program crash or potentially redirection of execution as per the attackers' choice.

Attack Prerequisites
- Targeted software performs buffer operations.
- Targeted software inadequately performs bounds-checking on buffer operations.
- Attacker has the capability to influence the input to buffer operations.

Solutions and Mitigations
Use a language or compiler that performs automatic bounds checking.
Use secure functions not vulnerable to buffer overflow.
If you have to use dangerous functions, make sure that you do boundary checking.
Compiler-based canary mechanisms such as StackGuard, ProPolice and the Microsoft Visual Studio /GS flag. Unless this provides automatic bounds checking, it is not a complete solution.
Use OS-level preventative functionality. Not a complete solution.
Utilize static source code analysis tools to identify potential buffer overflow weaknesses in the software.

https://capec.mitre.org/data/definitions/100.html
Adversary Tactics, Techniques, and Common Knowledge (ATT&CK)

- Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™) is a model and framework for describing the actions an adversary may take while operating within an enterprise network
  - Can be used to characterize post-Exploit adversary behavior
    - Focuses on Control, Execute, and Maintain steps within the cyber attack lifecycle
  - Can be used to help prioritize network defense against advanced persistent threat (APT) threat actors operating within the network
  - TTPs provide technical descriptions, indicators, targeted platforms, sensor data, detection analytics, and potential mitigations

# ATT&CK Taxonomy: Post Exploit Adversary TTPs

<table>
<thead>
<tr>
<th>Persistence</th>
<th>Privilege Escalation</th>
<th>Defense Evasion</th>
<th>Credential Access</th>
<th>Host Enumeration</th>
<th>Lateral Movement</th>
<th>Execution</th>
<th>C2</th>
<th>Exfiltration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legitimate Credentials</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accessibility Features</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AddMonitor</td>
<td>Binary Padding</td>
<td>Credential Dumping</td>
<td>Account enumeration</td>
<td>Application deployment software</td>
<td>Command Line</td>
<td>Commonly used port</td>
<td>Automated or scripted exfiltration</td>
<td></td>
</tr>
<tr>
<td>DLL Search Order Hijack</td>
<td>DLL Side Loading</td>
<td>Credentials in Files</td>
<td>File system enumeration</td>
<td>Exploitation of Vulnerability</td>
<td>File Access</td>
<td>Commm through removable media</td>
<td>Data compressed</td>
<td></td>
</tr>
<tr>
<td>Edit Default File Handlers</td>
<td>Disabling Security Tools</td>
<td>Network Sniffing</td>
<td>Group permission enumeration</td>
<td>Logon scripts</td>
<td>PowerShell</td>
<td>Data encrypted</td>
<td>Data size limits</td>
<td></td>
</tr>
<tr>
<td>New Service</td>
<td>Tools File System Logical Offsets</td>
<td>User Interaction</td>
<td>Peer connections</td>
<td>Pass the hash</td>
<td>Process Hollowing</td>
<td>Data staged</td>
<td>Data staged</td>
<td></td>
</tr>
<tr>
<td>Path Interception</td>
<td>Process Hollowing</td>
<td>Credential manipulation</td>
<td>Peer connections</td>
<td>Pass the ticket</td>
<td>Registry</td>
<td>Exfil over C2 channel</td>
<td>Exfil over alternate channel to C2 network</td>
<td></td>
</tr>
<tr>
<td>Scheduled Task</td>
<td></td>
<td></td>
<td>Local network connection enumeration</td>
<td>Remote Desktop</td>
<td>Rundll32</td>
<td>Custom application layer protocol</td>
<td>Exfil over other network medium</td>
<td></td>
</tr>
<tr>
<td>Service File Permission Weakness</td>
<td></td>
<td></td>
<td>Local networking enumeration</td>
<td>Service Manipulation</td>
<td>Service Task</td>
<td>Custom encryption cipher</td>
<td>Exil over physical medium</td>
<td></td>
</tr>
<tr>
<td>Shortcut Modification</td>
<td></td>
<td></td>
<td>Operating system enumeration</td>
<td>Third Party Software</td>
<td></td>
<td>Data obfuscation</td>
<td>Exil over physical medium</td>
<td></td>
</tr>
<tr>
<td>Web shell</td>
<td></td>
<td></td>
<td>Windows management instrumentation</td>
<td>Windows remote management</td>
<td></td>
<td>Fallback channels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIOS</td>
<td>Bypass UAC</td>
<td></td>
<td>Operating system enumeration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hypervisor Rootkit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLL Injection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exploitation of Vulnerability</td>
<td>Indicator blocking on host</td>
<td></td>
<td>Owner/User enumeration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indicators removal from tools</td>
<td></td>
<td></td>
<td>Process enumeration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indicator removal from host</td>
<td></td>
<td></td>
<td>Security software enumeration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Masquerading NTFS Extended Attributes</td>
<td></td>
<td></td>
<td>Service enumeration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTFS Extended Attributes</td>
<td></td>
<td></td>
<td>Window enumeration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obfuscated Payload</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rundll32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scripting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Packing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestomping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Additional Tips**

- **BIOS**: Bypass UAC, Bypass UAC
- **Hypervisor**: Rootkit, Rootkit
- **Logon Scripts**: Exploitation of Vulnerability, Indicator blocking on host
- **Master Boot Record**: Indicator removal from tools, Indicator removal from host
- **Mod. Exist’g Service**: Masquerading NTFS Extended Attributes
- **Registry Run Keys**: Obfuscated Payload, Rundll32
- **Serv. Reg. Perm. Weakness**: Scripting, Software Packing
- **Windows Mgmt Instr. Event Subsc.**: Timestomping
- **Winlogon Helper DLL**: Remote Services

**Links**

- [http://attack.mitre.org](http://attack.mitre.org)
An Example ATT&CK Technique

Exploitation of Vulnerability

Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Exploiting software vulnerabilities may allow adversaries to run a command or binary on a remote system for lateral movement, escalate a current process to a higher privilege level, or bypass security mechanisms. Exploits may also allow an adversary access to privileged accounts and credentials. One example of this is MS14-068, which can be used to forge Kerberos tickets using domain user permissions.¹²

Examples

- **FIN6** has used tools to exploit Windows vulnerabilities in order to escalate privileges.
  The tools targeted CVE-2013-3660, CVE-2011-2005, and CVE-2010-4398, all of which could allow local users to access kernel-level privileges.³

https://attack.mitre.org/wiki/Technique/T1068
Common Weakness Enumeration (CWE)

- MITRE open source repository of software weaknesses
  - Over 800 weaknesses currently identified
  - Updated quarterly

Derivation of Attack Vectors

- Cross-reference CWE and CAPEC to identify a range of attack patterns for a given set of software weaknesses
  - Example: Top 25 SANS/CWE weaknesses

http://cwe.mitre.org/
Common Vulnerabilities and Exposures (CVE)

- Open source repository of software vulnerabilities
  - Over 79000 CVEs reported across commercial software products
  - Weekly release cycle

- Derivation of Attack Vectors
  - Cross reference CVE with CAPEC to identify patterns that can exploit a given software vulnerability
  - Can be used to correlate vulnerabilities with specific technologies
    - Example: SNMP related attack vectors added to TARA catalog based on CVE vulnerabilities reported for SNMP agents

http://cve.mitre.org/
ICS-CERT Advisories

Advisories provide information about current security issues, vulnerabilities, and exploits, organized by vendor.

Each advisory identifies the affected product(s), impact, vulnerability, and mitigation.

https://ics-cert.us-cert.gov/
The TARA Assessment Methodology
Threat Assessment & Remediation Analysis (TARA)

- MITRE-developed methodology to identify and assess cyber threats and select countermeasures effective at mitigating those threats
  - Leverages catalog of Attack Vectors (AVs), Countermeasures (CMs), and associated mappings
    - Use of catalog ensures that findings are consistent across assessments
  - Uses scoring models to quantitatively assess AVs and CMs
    - AVs ranked by risk, providing a basis for effective triage
    - CMs ranked by cost-effectiveness, providing a basis for identifying optimal solutions
  - Delivers recommendations
    - Allows programs to make informed choices on how best to improve a system’s security posture and resilience
TARA Methodology Workflows

Workflow – Sequence of connected activities that produce useful work
Phases of a TARA Assessment

Objective is to identify and assess cyber threats and select countermeasures effective at mitigating those threats

- **Define Scope of Assessment**
  - The evaluation target(s)
  - The range of threats to be assessed
  - The adversary
  - The phase of the system acquisition lifecycle
  - Verify assessment scope with sponsor

- **Cyber Threat Susceptibility Analysis (CTSA)**
  - Model the target
  - Perform catalog search to identify candidate AVs
  - Eliminate implausible AVs
  - Define a scoring model to rank plausible AVs
  - Construct the Susceptibility Matrix

- **Cyber Risk Remediation Analysis (CRRA)**
  - Select AVs to mitigate
  - Use mitigation mappings to identify candidate countermeasures (CMs)
  - Eliminate implausible CMs
  - Define a scoring model to rank CMs
  - Select the best CM solution set
  - Develop well-formed recommendations
TARA Assessment Products

Suspceptibility Matrix

*Provides a ranked list of cyber threats, mapped to components of the evaluation target*

<table>
<thead>
<tr>
<th>Attack Vectors</th>
<th>Risk Score</th>
<th>Shopping cart</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV ID</td>
<td>AV Name</td>
<td>Browser</td>
</tr>
<tr>
<td>T000049</td>
<td>Buffer Overflow</td>
<td>High</td>
</tr>
<tr>
<td>T000014</td>
<td>Accessing, Intercepting, and Modifying HTTP Cookies</td>
<td>Moderate</td>
</tr>
<tr>
<td>T000050</td>
<td>Forced Integer Overflow</td>
<td>Moderate</td>
</tr>
<tr>
<td>T000071</td>
<td>SOAP Array Overflow</td>
<td>Moderate</td>
</tr>
<tr>
<td>T000052</td>
<td>Inducing buffer overflow to disable input validation</td>
<td>Low</td>
</tr>
<tr>
<td>T000170</td>
<td>Attack through shared data</td>
<td>Low</td>
</tr>
</tbody>
</table>

*Answers the questions: Where and how is my system most susceptible?*

Solution Effectiveness Table

*Provides a ranked list of countermeasures, mapped to cyber threats, and identifies the preventative or mitigating effect each countermeasure provides*

<table>
<thead>
<tr>
<th>Countermeasure (CM)</th>
<th>Scoring</th>
<th>Effect (by Attack Vector ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM ID</td>
<td>Name</td>
<td>U/C Ratio</td>
</tr>
<tr>
<td>C000134</td>
<td>Select programming languages that minimize software defects</td>
<td>75</td>
</tr>
<tr>
<td>C000117</td>
<td>Apply principle of least privilege</td>
<td>67</td>
</tr>
<tr>
<td>C000093</td>
<td>Merge data streams prior to validation</td>
<td>50</td>
</tr>
<tr>
<td>C000096</td>
<td>Use vetted runtime libraries</td>
<td>50</td>
</tr>
<tr>
<td>C000047</td>
<td>Encrypt session cookies</td>
<td>33</td>
</tr>
<tr>
<td>C000051</td>
<td>Use digital signatures/checksums</td>
<td>33</td>
</tr>
<tr>
<td>C000132</td>
<td>Use sandboxing to isolate running software</td>
<td>25</td>
</tr>
</tbody>
</table>

*Answers the questions: How are my threats mitigated and where are the gaps?*
Threat-informed Systems Analysis for Acquisition Programs

Hypothetical attack vectors selected based on analysis of conceptual system architecture (functional baseline)

Theoretical attack vectors selected based on analysis of preliminary system design (allocated baseline)

Potential and validated attack vectors based on detailed system design (production baseline)

Influence Requirements and Architecture

Influence Design

Influence Deployment
System Life Cycle Processes

The Systems Engineering “Vee” Model

Technical Processes

- Business or Mission Analysis
- System Rqmts Definition
- Stakeholder Needs & Rqmts Definition
- Architecture Definition
- Design Definition
- Implementation
- Project Test & Integration
- System Analysis
- Project Definition
- Integration
- Verification
- Validation
- Transition
- Operation
- Maintenance
- Disposal

Agreement Processes
- Acquisition
- Supply

Organizational Project-Enabling Processes
- Life Cycle Model Mgmt
- Infrastructure Mgmt
- Portfolio Mgmt
- Human Resource Mgmt
- Quality Mgmt
- Knowledge Mgmt

Technical Management Processes
- Project Planning
- Project Assess & Control
- Decision Mgmt
- Risk Mgmt
- Configuration Mgmt
- Information Mgmt
- Measurement
- Quality Assurance

Applications of TARA in the SSE Framework

- Security architecture analysis / threat model development
- Countermeasure selection (trade)
- Cyber risk assessments
- SCRM assessments
What’s Next?

- **TARA has been used to conduct cyber risk assessments for DoD acquisition programs since 2010**
  - Changes in the methodology can lead to different kinds of assessments and assessment artifacts
  - Changes to the underlying data model and/or technical content make possible assessments on different kinds of systems

- **Decision support for cyber incident analysis and response is a form of risk assessment conducted in an operational context**

- **Adaptation of TARA to support operational risk assessments**
  - Changes in how catalog data is selected and evaluated
  - Catalog content specific to nuclear reactor safety and control systems
  - Taxonomies that facilitate navigation within large sets of data
Backup Slides
Assessing Countermeasure Effects

The following table provides guidance for assessing the effect a countermeasure has on a given attack vector:

<table>
<thead>
<tr>
<th>Countermeasure Effect</th>
<th>Tends to be…</th>
</tr>
</thead>
<tbody>
<tr>
<td>prevent</td>
<td>detect</td>
</tr>
<tr>
<td>The countermeasure disrupts the attack's sequence of activities</td>
<td>X</td>
</tr>
<tr>
<td>The countermeasure eliminates condition(s) necessary for the attack to occur</td>
<td>X</td>
</tr>
<tr>
<td>The countermeasure facilitates detection of conditions leading to an attack</td>
<td>X</td>
</tr>
<tr>
<td>The countermeasure reduces the likelihood of the attack being successful</td>
<td>X</td>
</tr>
<tr>
<td>The countermeasure minimizes the extent of damage or disruption</td>
<td>X</td>
</tr>
<tr>
<td>The countermeasure facilitates rapid recovery/reconstitution after the attack occurs</td>
<td>X</td>
</tr>
<tr>
<td>The countermeasure facilitates forensic analysis and/or attribution following an attack</td>
<td>X</td>
</tr>
</tbody>
</table>