Lecture 14: Risk and Vulnerability in ICS
Outline

Vulnerability Identification/Scanning

Risk Classification and Ranking
Recall: Statistics of ICS Incidents

80% impacting ICS are “unintentional”

- Only 35% from outsider
- Insider + unintentional is a big concern

Embedded devices and network appliances were targeted 34%

- Windows-based ICS and enterprise hosts 66%

These numbers would help to understand risks that should be prioritized

https://scadahacker.com/
Recall: Flowchart of Assessing Risks to ICS

1. System characterization
2. Identify threat events, sources
3. Identify vulnerabilities
4. Classify risk
5. Determine risk
6. Measure likelihood and impact
7. Vulnerability ranking
8. Mitigation plan
9. Control recommendation
Recall: Scanning Industrial Networks

Device Scanners

Vulnerability Scanners

Traffic Scanner
Recall: Steps to be taken for System Characterization

Use \texttt{arp-scan} to identify network-connected hosts

Confirm identified hosts are authorized for the network. If not, physically inspect and take actions. Update system architecture with newly discovered info

Collect \texttt{host info} for each connected device, including hardware and OS info
  - Can be obtained via \texttt{systeminfo}

Collect \texttt{app info} for each device including vendor, name, patches, etc.
  - Can be obtained via \texttt{wmic}

Consolidate this info into database with appropriate classified policies
Vulnerability Identification

Vulnerability is not only unpatched software but also use of unnecessary services/apps

- Cannot be fully detected by scanning for presence (or absence) of software

Vulnerability can exist in form of:

- Improper authentication
- Poor credential management
- Improper access control
- Inconsistent documentation
Vulnerability Identification

Assessment phase depends on scanning tool

Involves review of relevant apps, host, config files

Physical aspect of ICS is inspected

Security controls are reviewed

Objective is to identify backdoors (holes) that may exist in the network perimeter
# Common ICS Vulnerabilities

<table>
<thead>
<tr>
<th>Category</th>
<th>Potential Vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Physical Security</td>
</tr>
<tr>
<td></td>
<td>Configuration Errors or Management</td>
</tr>
<tr>
<td></td>
<td>Port Security</td>
</tr>
<tr>
<td></td>
<td>Use of Vulnerable INP</td>
</tr>
<tr>
<td></td>
<td>Lack of IDS Capabilities</td>
</tr>
<tr>
<td>Config</td>
<td>Poor Account Management/Password Policies</td>
</tr>
<tr>
<td></td>
<td>Lack of Patch Management</td>
</tr>
<tr>
<td></td>
<td>Ineffective Whitelisting</td>
</tr>
<tr>
<td>Platform</td>
<td>Insecure Embedded Apps/Untrusted 3rd Party Apps</td>
</tr>
<tr>
<td></td>
<td>Lack of System Hardening</td>
</tr>
</tbody>
</table>
# Common ICS Vulnerabilities

<table>
<thead>
<tr>
<th>Category</th>
<th>Potential Vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICS Apps</td>
<td>Code Quality</td>
</tr>
<tr>
<td></td>
<td>Lack of Authentication</td>
</tr>
<tr>
<td></td>
<td>Vulnerable INP</td>
</tr>
<tr>
<td>Embedded Devices</td>
<td>Config Errors</td>
</tr>
<tr>
<td></td>
<td>Vulnerable INP</td>
</tr>
<tr>
<td></td>
<td>Insufficient Access Control</td>
</tr>
<tr>
<td>Policy</td>
<td>Security Awareness</td>
</tr>
<tr>
<td></td>
<td>Social Engineering</td>
</tr>
<tr>
<td></td>
<td>Physical Security</td>
</tr>
<tr>
<td></td>
<td>Access Control</td>
</tr>
</tbody>
</table>
Steps of Vulnerability Identification

Devices with little or no security feature are identified

- So they can be placed in special security zones and secured separately

Networks are reviewed to detect possible communication hijacking (MitM) opportunities

- Every component connected to ICS is assessed to discover improper features (i.e., no patch)

Good practice to work with suppliers

- So that they can also keep their vulnerabilities updated
Vulnerability Scanning

Some automated tools as we discussed earlier

Manual tests for critical host:

- Collecting info using command-line tools
- Comparing info of OS, apps and services against known vulnerabilities
- Two popular vulnerability database:
  - National Vulnerability Database (NVD) by NIST [https://nvd.nist.gov/]
  - Open-source Vulnerability Database (OSVDB) [https://cve.mitre.org/data/refs/refmap/source-OSVDB.html]

WhiteSource Vulnerability Database: [https://www.whitesourcesoftware.com/vulnerability-database/]
Example of Manual Vulnerability Scanning

1. Use “wmic” to list all installed apps running on Windows server

2. SCADA app software is shown as “XYZ” with vendor name “ABC” and version “2.3”

3. Using OSVDB with “ABC” keyword several results are returned

4. Compare your system to see if you have that vulnerability mentioned

5. Install the patches if available + needed
Authenticated Scan

Performs *white box* assessment

- **Authenticating** remotely on the device and performing variety of internal audits
  - Including network statistics

Provides **accurate reflection** of security posture of target

- Not just what is visible to attacker

More friendly on target and does not typically inject hostile traffic info network
## White Box vs. Black Box

<table>
<thead>
<tr>
<th>White</th>
<th>Black</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intent is to identify security vulnerabilities that could leak to an exploit</td>
<td>Represents system in a way that attacker sees it</td>
</tr>
<tr>
<td>Requires asset owner to disclosure significant info for test purposes</td>
<td>Protect intellectual property</td>
</tr>
<tr>
<td>Provides most comprehensive look at vulnerabilities and risk</td>
<td>Does not provide complete exposure to risk</td>
</tr>
</tbody>
</table>
Types of Vulnerability Scanning

Active mechanisms

- Place some packet on network
- “Aggressiveness” of the scan and impact on the target can be controlled

Passive scanner

- Snapshot view of vulnerabilities on target
- Able to enumerate network and detect new devices are added
  - Well suited for ICS due to static nature of network topology and regular traffic patterns

Host-based scanner

- Must be installed in host
  - Not really acceptable within ICS zone
Important Tips for Vulnerability Scanning

Should never be used on online ICS without prior testing and approval from directly responsible for operation of ICS

A system has no vulnerabilities does not mean that it has been configured in a secure manner

- Neither we can say that is fully secure
Configuration Auditing

Compliance auditing

- Compares current config of host against set of acceptable settings
- The settings may be determined by organization’s policy, regulatory standard, etc.

*Nessus* vulnerability scanner provides this audit

- Can be performed on config of OS, apps, antivirus, database, network infrastructure, etc.

US DoE funded project to develop set of security config guidelines for ICS components

- Can be used in Nessus
Vulnerability Scoring/Prioritization

Common Vulnerability Scoring System (CVSS) globally accepted industry standard for determining severity of system vulnerability

- **Base Metric**: mandatory component used to present characteristics of vulnerability
  - Constant with time and across different environments
  - Provided by the party responsible for disclosing the vulnerability
- **Temporal Metric**: Change over time but not across different environments
- **Environmental Metric**: reflects environmental characteristics
Example fields for Metrics

- **Base metrics**
  - Access vector
  - Confidentiality impact
  - Access complex
  - Integrity impact
  - Authentication
  - Availability impact

- **Temporal metrics**
  - Exploitability
  - Remediation level
  - Report confidence

- **Environment metrics**
  - Collateral damage potential
  - Target distribution
  - Security requirements
Definition of Keywords

An asset is what we’re trying to protect

A threat is what we’re trying to protect against

A vulnerability is a weakness or gap in our protection efforts

Risk is the intersection of assets, threats, and vulnerabilities

Risk is a function of threats exploiting vulnerabilities to obtain, damage or destroy assets. Thus, threats (actual, conceptual, or inherent) may exist, but if there are no vulnerabilities then there is little/no risk. Similarly, you can have a vulnerability, but if you have no threat, then you have little/no risk.

- Asset + Threat + Vulnerability = Risk
Risk Classification and Ranking

Compare the threats and vulnerabilities identified

- Important to make effective security program that addresses not only operational security but also business operations

Last step before taking actions (applying policies, etc.)

- Take into account the consequence to operations that would occur, if cyber event occurs

For instance gas pipelines that controlled by ICS;

- If a real battle fought, much harder for victory
- But cyber war
Estimate Consequences and Likelihood

Microsoft model DREAD (Damage Potential, Reproducibility, Exploitability, Affected User, Discoverability):

- Provides qualitative method of assigning value to each classification
- Consequence is not dependent on time
- Consider how easy to obtain knowledge (malware code) to exploit vulnerability
  - If no proof of concept has ever been developed, less likely to be exploited
- The skill level of attacker for that exploit
  - A script kiddie could perform this attack?
## DREAD Model

<table>
<thead>
<tr>
<th>Rating</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
<th>Indirectly Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Can subvert security</td>
<td>Leaking sensitive info</td>
<td>Leaking trivial info</td>
<td>Consequences</td>
</tr>
<tr>
<td></td>
<td>Get full trust authorization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Upload content</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Can be reproduced every time, does not require a timing window</td>
<td>Can be reproduced only with a timing window/particular</td>
<td>Very difficult to reproduce</td>
<td>Likelihood</td>
</tr>
<tr>
<td></td>
<td>No authentication required</td>
<td>situation Authorization required</td>
<td>Requires admin rights</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Novice programmer could make the attack in short time</td>
<td>Skilled programmer could make the attack</td>
<td>Attack requires extremely skilled person</td>
<td>Likelihood</td>
</tr>
<tr>
<td></td>
<td>Simple toolset</td>
<td>Exploit and tools are publicly available</td>
<td>and in-depth knowledge</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Custom exploit/tools</td>
<td></td>
</tr>
</tbody>
</table>
## DREAD Model

<table>
<thead>
<tr>
<th>Rating</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
<th>Indirectly Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>All users</td>
<td>Some users</td>
<td>Very small percentage of users (anonymous users)</td>
<td>Consequences</td>
</tr>
<tr>
<td></td>
<td>Default config</td>
<td>Non-default config</td>
<td>Obscure feature</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Key assets</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Published info explains the attack</td>
<td>Vulnerability is in seldom-used part of product</td>
<td>Bug is obscure</td>
<td>Likelihood</td>
</tr>
<tr>
<td></td>
<td>Vulnerability is in the most commonly used feature</td>
<td>Only a few users should come across would take some time to see malicious use</td>
<td>Requires source code Administrative access</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Very noticeable</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Risk Reduction and Mitigation

Should not be a one time tactical investment

- Instead a long-term strategic investment