Cyber Security Assessment of Enterprise-Wide Architectures

Mathias Ekstedt, Associate Prof.

Industrial Information and Control Systems
KTH Royal Institute of Technology
Agenda

Problem framing
  Management/design challenge
  Security metrics

Cyber Security Modeling Language (CySeMoL)
  What you see and what you get
  Inside the box
Cyber security management is difficult!

Is my control system secure enough?

Interconnected Complex architecture and data flow
Many vendors (incl. off-the-shelf components)

Which parameters decides cyber security?
Any vulnerabilities? And where are they?

And how do vulnerabilities relate?
In practice, cyber security management and design has limited resources

Should I spend my budget on: a training program for my staff, logging functionality, or network scanning?
Security assessment – how do you know..?

(Penetration) test

Theoretical metrics

Compliance

Communication Networks

Workstation for operators

Advanced Workstations

SCADA LAN

Communication Equipment (Pre/End)

Application Servers

Modem

DMZ LAN

Firewall

Historic Webserver

Office LAN

Firewall

System Vendors

Other Control Centers

INTERNET WAN

Geographically distributed process

SCADA Server (Online/Standby)

RTU / PLC

RTU / PLC

RTU / PLC

Automation Systems for Substations
Current decision support

Security audits/penetration tests
+ Measures actual security
- Is only valid for the aspects that are studied
- Only valid for the competence of the auditor(s)
- Is only valid for a single point in time
- Does not capture all types of vulnerabilities
- Is not always viable (e.g. ICS, design phase)

Literature such as ISO/IEC standards
- Cumbersome to interpret and implement
  - All encompassing standards → abstract
  - Detailed standards → unrelated knowledge islands
- Does not necessarily captures security
Cyber security metrics

A validity study of CWE/CVSS-based metrics:

Time from start of attack until successful compromise of that host → TTC (Time To Compromise)

- $t_1 = 1400.3 \text{ s}$
- $t_2 = 3000.2 \text{ s}$
- $TTC = t_2 - t_1$

Vulnerability information combined into different system level metrics
Cyber security metrics validity

A better security estimation model is needed…

The life for our decision-maker in summary...

- Poor understanding of the system architecture configuration and its environment
- Poor understanding of how to achieve security in this complex environment
- Limited resources, time and money, organizational support

> **Requirements and constraints** for this research
Agenda

Problem framing
  Management/design challenge
  Security metrics

Cyber Security Modeling Language (CySeMoL)
  Inside the box
  What you see and what you get
  Inside the box
**Attack and defense graphs**

- **Anti-malware**
  - Access as root to operating system
  - Execute arbitrary code
  - Exploit
  - Establish connection
  - Vulnerability exist

- **Network intrusion detection system**
  - Access as root to operating system
  - Execute arbitrary code
  - Exploit
  - Establish connection
  - Vulnerability exist
Bayesian networks

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
<th>F</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anti-malware</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Network intrusion detection</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>Execute code (TRUE)</td>
<td>0.21</td>
<td>0.32</td>
<td>0.41</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Anti-malware

Network intrusion detection system

Execute arbitrary code

Exploit

Establish connection

Vulnerability exist
Bayesian networks

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
<th>F</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anti-malware</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Network intrusion detection</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Execute code (TRUE)</td>
<td>0.21</td>
<td>0.32</td>
<td>0.41</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Anti-malware

Network intrusion detection system

Execute arbitrary code

Exploit

Establish connection

Vulnerability exist
Attacks and defenses – relation to assets

Network interface

Network intrusion detection system

Network zone

Anti-malware

Operating System

Application service

Establish connection

Access as root

Execute arbitrary code

Exploit

Vulnerability exist
Studies/topics covered by CySeMoL

Attacks/malicious activities:
• Zero-day discovery
• Memory corruption exploitation
• Web application exploitation (XSS, RFI, SQLi, Command injection)
• Social engineering
• Code injection using removable media
• Password guessing (online/offline)
• Denial of service
• Man-in-the-middle
• Discovery of unknown entry-points
• …
Studies/topics covered by CySeMoL

Defenses
- Network intrusion detection systems
  - Both detection and prevention-based
- Host intrusion detection systems
- Web application firewalls
- Anti-malware
- Firewalls
- Security training
- Encryption
- Software development best practice methods
- Network management (e.g., scanning, USB policy, etc)
- …
The Cyber Security Modeling Language (CySeMoL)

Actual architecture → Modeled architecture → Analysis results

Architecture language

Quantified theory

<table>
<thead>
<tr>
<th>Exploit</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anti-malware</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Network intrusion detection</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>
| Execute code (TRUE) | 0.21 | 0.32 | 0.41 | 0.7 | 0 | 0 | 0
CySeMoL screen shot – attack success

Green – low probability
Yellow – medium probability
Red – high probability
CySeMoL screen shot – attack success in detail (same system model but each attack step visualized individually)
Data sources

Parameters, relationships and dependency-structure:
- Literature, e.g. standards or scientific articles.
- Review and prioritization by external experts.

The probabilities:
- **Logical necessities**, e.g.: if the firewalls allow you to connect to A from B and you have access to B, then you can connect to A.
- **Others’ scientific studies**, e.g. time-to-compromise for authentication codes and patch level vs patching procedures.
- **Experts’ judgments**, Own surveys to researchers and security professionals.
- **Own experiments**, lab and cyber defense exercises
Data from expert judgment

Review of variables to include in the scenarios.
+

Probabilities on scenarios:
• Finding unknown entry-points: 4 penetration testers.
• Finding unknown vulnerabilities: 18 researchers.
• Arbitrary code exploits: 22 penetration testers and researchers.
• Intrusion detection: 165 researchers.
• DoS: 50 researchers.
• Web application vulnerability discovery and defenses: 21 researchers and penetration testers
Cooke’s classical method for weighting experts

Find the “true expert” not the average of experts in general. (It is enough if one person knows the truth, if we can only identify that person…)

A knowledge test with a number of questions (~10)
Respondents’ weights are derived from their answers’ on these questions, based on if they are
– calibrated/correct
– informative

This is “best practice”

Roger M Cooke, Experts in uncertainty: opinion and subjective probability in science, 1991
Survey example

Web application vulnerability discovery

QUESTION 10

Consider the scenario described by the conditions in the table below.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>The application was developed using a type-safe API</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The developers have undergone web application security training</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>The application has been improved through black box testing tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The application has been improved through static code analysis tools</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>

Let $X$ be the effort (in hours) required by a professional penetration tester to discover an input validation vulnerability under these conditions. What is the value of $X$ according to your judgement?

- It is very unlikely (5 % chance) that $X$ is below: 16 hours
- It is fifty-fifty (50 % chance) that $X$ is below: 32 hours
- It is very likely (95 % chance) that $X$ is below: 40 hours

Cumulative probability distribution

Hannes Holm, Mathias Ekstedt, Teodor Sommestad, *Effort estimates on web application vulnerability discovery*, Hawaii International Conference on Systems Sciences (HICSS), 2013
Conducted experiments

Signature-based network intrusion detection systems
Network vulnerability scanners
Phishing
Effectiveness of network intrusion detection

How effective is Snort at detecting known attacks?
How effective is Snort at detecting zero day attacks?

173 attacks less novel than the rule set
183 attacks more novel than the rule set

*An aggregation of data for 1990 (1 sample), 1994 (1 sample), 1998 (2 samples), 1999 (2 samples), 2000 (4 samples).

Fig. 1. An overview of the tested exploits categorized according to the disclosure of their vulnerabilities.
Effectiveness of network intrusion detection – known

### TABLE III

Detection rates for known exploits given different operating system environments and services (sample sizes are given within brackets).

<table>
<thead>
<tr>
<th>Service</th>
<th>Detection rate</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Windows</td>
<td>Unix</td>
<td>Multi</td>
</tr>
<tr>
<td>Total</td>
<td>55% (173)</td>
<td>57% (125)</td>
<td>52% (42)</td>
<td>50% (6)</td>
</tr>
<tr>
<td>FTP</td>
<td>92% (26)</td>
<td>95% (22)</td>
<td>67% (3)</td>
<td>100% (1)</td>
</tr>
<tr>
<td>HTTP</td>
<td>37% (38)</td>
<td>40% (35)</td>
<td>0% (3)</td>
<td>-</td>
</tr>
<tr>
<td>Web applications</td>
<td>60% (15)</td>
<td>-</td>
<td>60% (15)</td>
<td>-</td>
</tr>
<tr>
<td>SMB/Samba</td>
<td>82% (11)</td>
<td>83% (6)</td>
<td>80% (5)</td>
<td>-</td>
</tr>
<tr>
<td>SMTP/POP3/IMAP</td>
<td>60% (10)</td>
<td>67% (9)</td>
<td>0% (1)</td>
<td>-</td>
</tr>
<tr>
<td>Other*</td>
<td>47% (73)</td>
<td>47% (53)</td>
<td>47% (15)</td>
<td>40% (5)</td>
</tr>
</tbody>
</table>

*32 different services
Effectiveness of network intrusion detection – zero day

**TABLE I**

**ZERO-DAY DETECTION RATES FOR DIFFERENT OPERATING SYSTEM ENVIRONMENTS AND SERVICES (SAMPLE SIZES ARE GIVEN WITHIN BRACKETS).**

<table>
<thead>
<tr>
<th>Service</th>
<th>Detection rate</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>Windows</td>
<td>Unix</td>
<td>Multi</td>
</tr>
<tr>
<td>Total</td>
<td>19%* (183)</td>
<td>17% (135)</td>
<td>28% (40)</td>
<td>13% (8)</td>
<td></td>
</tr>
<tr>
<td>FTP</td>
<td>85% (13)</td>
<td>90% (10)</td>
<td>67% (3)</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>HTTP</td>
<td>12% (49)</td>
<td>12% (42)</td>
<td>0% (4)</td>
<td>33% (3)</td>
<td></td>
</tr>
<tr>
<td>Web applications</td>
<td>25% (16)</td>
<td>-</td>
<td>25% (16)</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>SMB/Samba</td>
<td>75% (4)</td>
<td>75% (4)</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>SMTP/POP3/IMAP</td>
<td>75% (4)</td>
<td>100% (3)</td>
<td>0% (1)</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Other a</td>
<td>8% (97)</td>
<td>4% (76)</td>
<td>31% (16)</td>
<td>0%</td>
<td></td>
</tr>
</tbody>
</table>

* p = 1.84 \cdot 10^{-10}

a 23 different services
Validity and reliability

CySeMoL has been validated on a component-level through the studies used to create it.

CySeMoL has been validated on a system-level through a Turing-test.
In summary: what CySeMoL can do for you

This is (roughly) what my future system alternatives look like

Probably, I can’t say for sure, but it seems as if scenario 2 is the most secure alternative
More information

Please visit:

www.ics.kth.se/cysemol