Ontologies for Modeling Enterprise Level Security Metrics

Anoop Singhal
Computer Security Division
National Institute of Standards and Technology
Enterprise Network Security Management

- Networks are getting large and complex
- Vulnerabilities in software are constantly discovered
- Network Security Management is a challenging task
- Even a small network can have numerous attack paths
Trends for Published Vulnerabilities
Current Status of Enterprise Network Security Management

- Currently, security management is more of an art and not a science
- System administrators operate by instinct and learned experience
- There is no objective way of measuring the security risk in a network
- “If I change this network configuration setting will my network become more or less secure?”
Why Security Metrics

- Difficult questions to answer:
  - How secure is the database server in a given network configuration?
  - How much security does a new configuration provide?
  - How can I plan on security investments so it provides a certain amount of security?
  - Which countermeasures or controls provide the greatest risk reduction

- For this we need a model or an ontology for Enterprise Level Security
If you cannot measure (or model) it, you cannot improve it.

---Lord Kelvin
Challenges in Security Metrics

- Metric for individual vulnerability exists
  - Impact, exploitability, temporal, environmental, etc.
  - E.g., the Common Vulnerability Scoring System (CVSS) v2 released on June 20, 2007\(^1\)

- However, how to compose individual measures for the overall security of a network?
  - Our work focuses on this issue

---

What is an Attack Graph

- A model for
  - How an attacker can combine vulnerabilities to stage an attack such as a data breach
  - Dependencies among vulnerabilities
  - Present all possible attack paths in a compact graphical structure
What is an Ontology

- It is a set of entities and relations
- It can be created for any collection of related concepts
- One application of ontology is to organize expert knowledge (e.g. automobiles, electronic items, human diseases and so on)
Ontology for Managing Enterprise Level Security

- Precise definitions of computer security concepts and their relationships
- The ontology should have *knowledge* about threats, assets and security mechanisms
- A secondary goal is to make the ontology portable
What is OWL

- Web Ontology Language
- *Classes* describe concepts
- *Sub-classes represent concepts that are more specific*
- *Instances* are members of this class
- *Properties* can define relationships among classes
- *Properties* can also defines different attributes of a class
Example of OWL

- Security Mechanism is a class
- A Detective Mechanism is a sub class
- A Preventive Mechanism is also a sub class
- IDS is an instance of a Detective Mechanism
- A Firewall is an instance of a Preventive Mechanism
- Asset is an example of another class
- A Security Mechanism *protects* an asset
- An asset *has a value*
An Ontology for Security Metrics

- Threat
- Vulnerabilities
- Countermeasures
- Assets
- Risk
- Security Objectives
- Business Goals
- Use Cases
Security Objectives
- confidentiality
- integrity
- availability
- authorization
- authentication
- trust

Security Mechanisms
- hostMechanism
- networkMechanism
- serviceMechanism
- preventiveMechanism
- detectiveMechanism
- correctiveMechanisms
- cost

Threat of Attacks
reduces
- frequency

Risk
probability
- damageValue

Vulnerability
isVulnerableTo
- score
- lifeCycle: {unknown, detected, patched}

Hardware
Time
- timeOfDay

Software
Configurations

Assets
depends
Resources
contains

Use Cases
belongsTo

QoS
refinedTo
systemWideMetrics

Business Goals
systemWideMetrics
- time
- qualityMetrics

QoS Metrics

HasMonetaryValue

Value

secure
Properties of the Asset Class

- `<rdf:Property rdf:ID="value">`  
  - `<rdfs:domain rdf:resources="Asset"/>`  
  - `<rdfs:range rdf:resources=&xsd:integer/>`  
  - `</rdf:Property>`
- `<rdf:Property rdf:ID="depends">`  
  - `<rdfs:domain rdf:resources="Asset"/>`  
  - `<rdfs:range rdf:resources="Asset"/>`  
  - `</rdf:Property>`
- `<rdf:Property rdf:ID="contains">`  
  - `<rdfs:domain rdf:resources="Asset"/>`  
  - `<rdfs:range rdf:resources="Asset"/>`  
  - `<rdf:Property rdf:ID="isVulnerableTo">`  
  - `<rdfs:domain rdf:resources="Asset"/>`  
  - `<rdfs:range rdf:resources="Vulnerability"/>`  
  - `</rdf:Property>`
- `<rdf:Property rdf:ID="belongsTo">`  
  - `<rdfs:domain rdf:resources="Asset"/>`  
  - `<rdfs:range rdf:resources="Resource"/>`  
  - `<rdf:Property rdf:ID="monitaryValue">`  
  - `<rdfs:domain rdf:resources="Assets"/>`  
  - `<rdfs:range rdf:resources="Value"/>`  
  - `</rdf:Property>`
- `<rdf:Property rdf:ID="supportUsage">`  
  - `<rdfs:domain rdf:resources="Assets"/>`  
  - `<rdfs:range rdf:resources="Use Cases"/>`  
  - `</rdf:Property>`
- Stands for *Common Vulnerability Scoring System*
- An open framework for communicating characteristics and impacts of IT vulnerabilities
- Consists three metric groups: *Base*, *Temporal*, and *Environmental*
CVSS (Cont’d)

- **Base metric**: constant over time and with user environments
- **Temporal metric**: change over time but constant with user environment
- **Environmental metric**: unique to user environment
CVSS (Cont’d)

CVSS metric groups

- Each metric group has sub-matrixes
- Each metric group has a score associated with it
- Score is in the range 0 to 10
Access Vector

This metric measures how the vulnerability is exploited.

- Local
- Adjacent Network
- Network
Access Complexity

This metric measures the complexity of the attack required to exploit the vulnerability

- High: Specialized access conditions exist
- Medium: The access conditions are somewhat specialized
- Low: Specialized access conditions do not exist
Authentication

This metric measures the number of times an attacker must authenticate to a target to exploit a vulnerability

- Multiple: The attacker needs to authenticate two or more times
- Single: One instance of authentication is required
- None: No authentication is required
Confidentiality Impact

This metric measures the impact on confidentiality due to the exploit.

- None: No Impact
- Partial: There is a considerable information disclosure
- Complete: There is total information disclosure

Similar things for the Integrity Impact and Availability Impact
Base Score

Base Score = Function(Impact, Exploitability)

Impact = 10.41 * (1-(1-ConImp)*(1-IntImp)*(1-AvailImpact))

Exploitability = 20*AccessV*AccessComp*Authentication
Base Score Example CVE-2002-0392

- Apache Chunked Encoding Memory Corruption

<table>
<thead>
<tr>
<th>BASE METRIC</th>
<th>EVALUATION</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Vector</td>
<td>[Network]</td>
<td>(1.00)</td>
</tr>
<tr>
<td>Access Complex.</td>
<td>[Low]</td>
<td>(0.71)</td>
</tr>
<tr>
<td>Authentication</td>
<td>[None]</td>
<td>(0.704)</td>
</tr>
<tr>
<td>Availability Impact</td>
<td>[Complete]</td>
<td>(0.66)</td>
</tr>
</tbody>
</table>

Impact = 6.9
Exploitability = 10.0
BaseScore = (7.8)
Example Queries

- Find all Assets with value > 100K that have vulnerabilities that are published but not patched
- Which security mechanism will prevent a certain attack and how much does it cost
- Suppose a vulnerability is discovered in a certain version of a shared library, give me all products that use this shared library and are affected by it.
Conclusions

- Presented an Ontology for Modeling Enterprise Level Security
- Implemented it using OWL
- It can be used to generate reports about enterprise level security