Toward Scalable Security Models and Analysis

Dong-Seong (Dan) Kim

University of Canterbury
Christchurch, New Zealand
Email: dongseong.kim@canterbury.ac.nz
Outline

• Introduction
• Problems
• Our proposed ideas
  – HARMs
  – Simplified HARM in construction
  – Simplified HARM in evaluation
• Summary
New Zealand
NATO Emerging Security Challenges Division Science for Peace and Security Programme (SPS)

NATO Country (USA)

NPD: Prof. K. Trivedi, Duke Uni.
Co-Director: Assoc. Prof. D. Huang, ASU

NATO Partner Country (Morocco)

PPD: Prof. A. Haqiq,
Hassan 1st University

Major non-NATO allies Country

Dr. Dong-Seong Kim
(U of Canterbury, New Zealand)
Cyber Security Analysis and Assurance using Cloud-Based Security Measurement System

- Vulnerabilities Database (NVD, CVE, SecurityFocus, etc)
- Connectivity (Topology)
- Attacks (threats)
- Detection/Mitigation
- etc

Cloud-Based Security and monitoring and Measurement (CBSM) system

Enterprise network systems /Cloud systems

Attack and Defense Model (aka. Representation Models (ARM))

Security Analysis results
Security Assessment

How secure is my network?

NIDS: network intrusion detection system

How to assess security?
Security Assessment (cont.)

• To assess security, one requires **3Ms**:  
  1. Security **Measures**  
     • To **collected** required information.
  2. Security **Metrics**  
     • To **represent** the analysis of security.
     • To **capture** security using simulation, analytic models, or hybrid models.

Lifecycle of ARM?
Attack Representation Model (ARM)*

life cycles

*aka., Attack and Defense Models

*an example?
An example network and AG

**Security objective:** to harden the network w.r.t target condition root(2)

- **Vulnerabilities:**
  - ftp_rhosts
  - rsh
  - sshd_bof
  - local_bof

User 1 trusts User 0

- **User 1**
- **User 2**

**Goal:** User 0 acquires User 2’s root

An example network and AG

Security objective: to harden the network w.r.t target condition root(2)

Vulnerabilities:
- ftp_rhosts
- rsh
- sshd_bof
- local_bof
- web_bof

Vulnerabilities:
- ftp_rhost
- rsh
- local_bof

How scalable and adaptable this security model?

Goal: User0 acquires User2’s root

Two issues on ARMs

• **Scalability issues**
  – The generation/evaluation of full attack models (all possible attack scenarios) exhibit a state-space explosion.

• **Dynamic adjustment issues**
  – A change in the network system causes reconstruction (in worst case) of the ARMs.

Dealing with Scalability

1. Using Hierarchical ARMs (HARMs)
   – Modelling hosts and vulnerabilities in two different layers (i.e., 2-level hierarchy).
   – Simulation result

2. Construct ARMs based on Important components
   – Improve the construction complexity using less components.

3. Security Analysis based on Important components
   – Using important hosts and vulnerabilities for security analysis.
Our proposed idea

Use of two-level *Hierarchical* ARMs (HARMs)

Budapest = Buda (higher area) + pest (lower area)

Danube river

Represent the *network path information* in the upper level and *vulnerability exploitation information* in the lower level

Note that this can be extended to multi-level Hierarchical model.

Example of HARMs

Vulnerabilities:
• ftp_rhosts
• rsh
• sshd_BoF
• local_BoF

Vulnerabilities:
• ftp_rhosts
• rsh
• local_BoF
Performance Evaluation via Simulation

• Assume an external attacker and a target
• Consider
  – performances in construction and evaluation phase for an AG and an HARM (AG-AG type)
    • Time, number of computations
  – various network topologies
    • Fully connected, ring and star
  – variable number of vulnerabilities
    • Hosts are assigned with varying number of vulnerabilities
  – different vulnerability types
    • Vulnerabilities to gain different level of privileges (e.g., user/root)
Performance Evaluation via Simulations (cont.)

- Simulation 1 – fully connected topology, bounded attack path length

Fixed No. of Vulnerabilities = 10
(1 remote-to-root, 9 remote-to-other)
Performance Evaluation via Simulations

- Simulation 2 – various topologies, attack path length unbounded

- HARM performs better than AG in all topology types.

- Increase #hosts.

Evaluation
Fixed No. of Vulnerabilities = 10
(1 remote-to-root, 9 remote-to-other)
Performance Evaluation via Simulations

- Simulation 3 – various number of vulnerabilities (L2R only), attack path length unbounded

Increase #vul. 

HARM performs better than AG in all topology types.

Evaluation (Fixed No. of Hosts = 3)

Evaluation (Fixed No. of Hosts = 1200)
Dealing with Scalability

1. Using Hierarchical ARMs (HARMs)
   – Modelling hosts and vulnerabilities in two different layers (i.e., 2-level hierarchy).
   – Simulation result

2. Construct ARMs based on Important components
   – Improve the construction complexity using less components.

3. Security Analysis based on Important components
   – Using important hosts and vulnerabilities for security analysis.
Construct ARMs using Important Components

• When analysing network security, there are only a subset of network components that have a critical role in an event of an attack.

• All network components are considered when the ARMs are constructed.

• To improve the performance of both construction and evaluation phases of ARMs, we consider to use only important hosts and vulnerabilities.
Recap – ARM life cycles

Preprocessing
- Reachability
- Other if necc.
- Vulnerability

Construction (Generation)
- Build (Update) ARM
- Compute important components

Representation
- Visualisation/Storage

Evaluation
- Security metrics

Modification
- Applying security best practices
- Change(s) in the network
- Network
- Update

Reachability information
Vulnerability information
Updated information
An example network

The goal is to compromise H5.

Attackers located outside the internal network
An example network and its vulnerabilities

Vulnerabilities of $H_1 - H_4$

<table>
<thead>
<tr>
<th>ID</th>
<th>CVE ID</th>
<th>CVSS BS</th>
<th>Impact</th>
<th>Exploitability</th>
<th>CI</th>
<th>Access Level</th>
<th>Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>CVE-2005-1794</td>
<td>6.4</td>
<td>4.9</td>
<td>10.0</td>
<td>P</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_2$</td>
<td>CVE-2011-0661</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>C</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_3$</td>
<td>CVE-2010-0231</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>C</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_4$</td>
<td>CVE-2011-2552</td>
<td>7.8</td>
<td>6.9</td>
<td>10.0</td>
<td>N</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_5$</td>
<td>CVE-1999-0520</td>
<td>6.4</td>
<td>4.9</td>
<td>10.0</td>
<td>P</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_6$</td>
<td>CVE-2010-2729</td>
<td>9.3</td>
<td>10.0</td>
<td>8.6</td>
<td>C</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_7$</td>
<td>CVE-1999-0505</td>
<td>7.2</td>
<td>10.0</td>
<td>3.9</td>
<td>C</td>
<td>Admin</td>
<td>None</td>
</tr>
<tr>
<td>$v_8$</td>
<td>CVE-2002-1117</td>
<td>5.0</td>
<td>2.9</td>
<td>10.0</td>
<td>P</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_9$</td>
<td>CVE-2003-0386</td>
<td>4.3</td>
<td>2.9</td>
<td>8.6</td>
<td>P</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{10}$</td>
<td>CVE-2010-0025</td>
<td>5.0</td>
<td>2.9</td>
<td>10.0</td>
<td>P</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{11}$</td>
<td>CVE-1999-0497</td>
<td>0.0</td>
<td>0.0</td>
<td>10.0</td>
<td>N</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

Vulnerabilities of $H_5$

<table>
<thead>
<tr>
<th>ID</th>
<th>CVE ID</th>
<th>CVSS BS</th>
<th>Impact</th>
<th>Exploitability</th>
<th>CI</th>
<th>Access Level</th>
<th>Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{12}$</td>
<td>CVE-2011-1789</td>
<td>5.0</td>
<td>2.9</td>
<td>10.0</td>
<td>N</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{13}$</td>
<td>CVE-2011-1786</td>
<td>5.0</td>
<td>2.9</td>
<td>10.0</td>
<td>N</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{14}$</td>
<td>CVE-2011-1785</td>
<td>7.8</td>
<td>6.9</td>
<td>10.0</td>
<td>N</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{15}$</td>
<td>CVE-2011-0355</td>
<td>7.8</td>
<td>6.9</td>
<td>10.0</td>
<td>N</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{16}$</td>
<td>CVE-2010-4573</td>
<td>9.3</td>
<td>10.0</td>
<td>8.6</td>
<td>C</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{17}$</td>
<td>CVE-2010-3609</td>
<td>5.0</td>
<td>2.9</td>
<td>10.0</td>
<td>N</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{18}$</td>
<td>CVE-2010-1142</td>
<td>8.5</td>
<td>10.0</td>
<td>6.8</td>
<td>C</td>
<td>None</td>
<td>Single System</td>
</tr>
<tr>
<td>$v_{19}$</td>
<td>CVE-2010-1141</td>
<td>8.5</td>
<td>10.0</td>
<td>6.8</td>
<td>C</td>
<td>None</td>
<td>Single System</td>
</tr>
<tr>
<td>$v_{20}$</td>
<td>CVE-2009-3733</td>
<td>5.0</td>
<td>2.9</td>
<td>10.0</td>
<td>P</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{21}$</td>
<td>CVE-2008-4281</td>
<td>9.3</td>
<td>10.0</td>
<td>8.6</td>
<td>C</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>$v_{22}$</td>
<td>CVE-2008-2097</td>
<td>9.0</td>
<td>10.0</td>
<td>8.0</td>
<td>C</td>
<td>Admin</td>
<td>Single System</td>
</tr>
</tbody>
</table>

CI: confidentiality impact

using vulnerability scanners such as NESSUS
Ranking hosts

• Ranking Hosts w.r.t NCMs

<table>
<thead>
<tr>
<th></th>
<th>Degree</th>
<th>Closeness</th>
<th>Betweenness</th>
<th>Rank Sum</th>
<th>Final Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>H₄</td>
<td>3/4</td>
<td>4/5</td>
<td>10/12</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>H₁</td>
<td>3/4</td>
<td>4/5</td>
<td>8/12</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>H₂</td>
<td>3/4</td>
<td>4/5</td>
<td>8/12</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>H₃</td>
<td>2/4</td>
<td>4/7</td>
<td>4/12</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>H₅</td>
<td>1/4</td>
<td>4/12</td>
<td>4/12</td>
<td>14</td>
<td>5</td>
</tr>
</tbody>
</table>

— Degree (node popularity), Closeness (related distance), Betweenness (usage of a node between paths)

• combine all NCMs to formulate the final rank
  — Each rank acted as a score to give the final rank (i.e., scores are used to re-rank nodes)
  — Rankings from each NCM are used as the importance score
Ranking vulnerabilities on hosts

• Ranking Vulnerabilities

Vulnerabilities are ranked based on their CVSS BSs (common vulnerability score system base score)

Important vulnerabilities are selected based on the threshold value (e.g., higher than the average CVSS BSs)

<table>
<thead>
<tr>
<th>Vulnerabilities of $H_1 - H_4$</th>
<th>$\nu_2$</th>
<th>$\nu_3$</th>
<th>$\nu_6$</th>
<th>$\nu_4$</th>
<th>$\nu_7$</th>
<th>$\nu_1$</th>
<th>$\nu_5$</th>
<th>$\nu_8$</th>
<th>$\nu_{10}$</th>
<th>$\nu_9$</th>
<th>$\nu_{11}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVSS BS</td>
<td>10.0</td>
<td>10.0</td>
<td>9.3</td>
<td>7.8</td>
<td>7.2</td>
<td>6.4</td>
<td>6.4</td>
<td>5.0</td>
<td>5.0</td>
<td>4.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vulnerabilities of $H_5$</th>
<th>$\nu_{16}$</th>
<th>$\nu_{21}$</th>
<th>$\nu_{22}$</th>
<th>$\nu_{18}$</th>
<th>$\nu_{19}$</th>
<th>$\nu_{14}$</th>
<th>$\nu_{15}$</th>
<th>$\nu_{12}$</th>
<th>$\nu_{13}$</th>
<th>$\nu_{17}$</th>
<th>$\nu_{20}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVSS BS</td>
<td>9.3</td>
<td>9.3</td>
<td>9.0</td>
<td>8.5</td>
<td>8.5</td>
<td>7.8</td>
<td>7.8</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
Revisit: the example network

Attacker located outside the internal network

Attacker located outside the internal network
A HARM for the example net

• Naïve method: AG-AT HARM – upper level
A simplified HARM

- Using only important hosts: AG-AT HARM in the upper level
A HARM for the example net

- Naïve method: AG-AT HARM in the lower level

![Diagram of HARM example net]
A simplified HARM

• Using Important vulnerabilities: AG-AT HARM

Using 5 vulnerabilities

Compromise *Host

\[ \text{AND}_1 \]

\[ V_7 \]

\[ \text{OR}_1 \]

\[ V_2 \]
\[ V_3 \]
\[ V_4 \]
\[ V_6 \]

None

Using 7 vulnerabilities

Compromise *Target

\[ \text{AND}_1 \]

\[ V_{22} \]

\[ \text{AND}_2 \]

\[ \text{OR}_1 \]

\[ \text{OR}_2 \]

\[ V_{16} \]
\[ V_{21} \]
\[ V_{14} \]
\[ V_{15} \]
\[ V_{18} \]
\[ V_{19} \]

None

Lower level (AT)
Performance Simulation

- Results – host based security analysis

The construction time linearly improves as the number of important hosts modelled reduce. For evaluation, there is a steady improvement until the host number reaches 500. From then, it rapidly improves the performance.
Performance Simulation (cont.)

- Results – vulnerability based security analysis

The construction time linearly improves as the number of important vuls modelled reduce. The variation of vulnerability numbers has minimum effect.
Conclusion

• Constructing ARMs using only important hosts and vulnerability can improve the performance in construction and evaluation.

  – Nearly equivalent security analysis is performed, with 87% improved construction time and 99.5% improved evaluation time in the simulation.
Dealing with Scalability

1. Using Hierarchical ARMs (HARMs)
   – Modelling hosts and vulnerabilities in two different layers (i.e., 2-level hierarchy).
   – (semi-)automated generation
   – Simulation result

2. Construct ARMs based on Important components
   – Improve the construction complexity using less components.

3. Security Analysis based on Important components
   – Using important hosts and vulnerabilities for security analysis.
Attack Representation Model (ARM) life cycles

1. Use only important hosts
2. Use only important vulnerabilities in hosts

1. Scalable?
2. Equivalent security solution c.f. exhaustive search?
Network coverage

- Consider an attack scenario that covers only a subset of the network (e.g., an attacker located inside the network).

How to define the subnet covered by the attack scenario?
Attacker to Victim Centrality (AVC)

• Typical NCMs in the upper level do not consider the location of the attacker and the target (victim).

• We define a location-based (Attacker to Victim) centrality (AVC) measurement based on distance measurements.

Measure distance between A and T
Attacker to Victim Neighbour Centrality (AVNC)

- If there are components with the same ranking, then the AVC may not identify important components correctly.

AVC measures

- Same rank
- The closer to T, the higher the rank

Therefore, we assign more weights to nodes that are closer to highly important components (value could be weighted)
Attack scenario:
Attacker compromise $x_i$ DMZ hosts, through Internal hosts, then finally obtain data from the designated DB
Security Analysis via Simulation (cont.)

- Results – multiple subnets, external attacker

The host based importance based measures perform better than vulnerability based importance measures.

There are components with same importance rankings. Patching vulnerabilities for these components may not reduce the risk value, so there are fluctuations in the graph.
Attack scenario:
Attacker compromise $x_i$ Internal hosts, obtain data from the designated DB
The location based centrality measure AVNC performs most closely to the exhaustive search.

Security Analysis via Simulation (cont.)

• Results – multiple subnets, internal attacker

The location based centrality measure AVNC performs most closely to the exhaustive search.
Limitations

• **Combinations** of rankings
  – Overlaps between NCMs indicate improvements can be achieved by combining their rankings
  – Combining with vulnerability rankings

• **Multiple** target host locations
  – Changes in rankings

• Attack on **less** important hosts and vulnerabilities
  – **High** cost attacks
  – Advanced **persistent** threat (APT)
Conclusion

• Evaluating HARMs using only important hosts and vulnerability can improve the performance of evaluation.

• Nearly equivalent security analysis is achievable, with improved evaluation time (from exponential down to polynomial.)
Final summary

1. Performance Analysis via Simulations
   - Reachability information
   - Logic Reduction Techniques
   - Build/Update ARMs
   - 2. Importance based ARMs Construction

2. Importance based Security Analysis
   - Visualization/Storage
   - Security Analysis
   - 3. Importance based Security Analysis

3. Network
   - Change(s) in the network
   - Updated information
   - Cloud, DID
   - MTDs (via SDN)
   - Applying security best practices

Preprocessing
- Reachability
- Other if necc.
- Vulnerability

Construction (Generation)
- Reachability information
- Build/Update ARMs

Representation
- ARMs
- Visualisation/Storage
- Updated information
- ARMs

Evaluation
- Security metrics

Modification
- Network
- Update
- Cloud, DID
- MTDs (via SDN)
Scalable Security Models

Hagley Park, Christchurch, New Zealand

Dong-Seong Kim
donseong.kim@canterbury.ac.nz
University of Canterbury
Related publications

- Arpan Roy, Dong Seong Kim, Kishor S. Trivedi: Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees. DSN 2012
- Jin Hong, Dong Seong Kim, "Performance analysis of scalable attack representation models" In Proc. of the 28th IFIP TC-11 International Information Security and Privacy Conference (SEC 2013)
- Jin Hong, Dong Seong Kim, Scalable Security Analysis in Hierarchical Attack Representation Model using Centrality Measures, in Proc. of RSDA 2013 in conjunction with DSN 2013.
- Jin Hong, Dong Seong Kim, Scalable Attack Representation Model using Logic Reduction Techniques, in Proc. of TrustCom 2013.