EVA
Evolutionary Vulnerability Analyzer
A Framework for Network Analysis and Risk Assessment
• Introduction
• Attack Graphs
  – Model
  – Creation
• Analysis of Attack Graphs
  – Evolutionary Method
  – Modes of Analysis
• Experimental Results
• Problem: Vulnerability scanners limited
  – Only evaluates individual machines
  – Cannot show how vulnerabilities relate
• Example: “Foothold” situation
  – Attacker compromises machine A
  – Machine A has private communication channel with machine B
  – Attacker uses machine A to attack machine B
Solution: Attack graphs

- Visual representation of exploits paths
• Benefits of analyzing attack graphs
  – Find a set of hardening measures
  – Perform “what if” evaluations
  – Assist with network design
  – Guide forensics evaluation
  – Detect multi-stage attacks from IDS alerts
Attack Graphs

Model
• Nodes of the graph
  – Initial nodes represent the present state of the network
  – Interior and terminal nodes represent states the attacker has achieved

• Edges of the graph
  – Attacks executed by attacker
  – Represented visually as a diamond “node”
• Exploit path is sequence from initial nodes to a terminal node
• Discovers exploit paths through attack template “requires/provides” syntax
  – Templates have preconditions (requirements) and postconditions (consequences)
  – Postcondition of one attack may be a precondition for another attack
  – Path is sequence of such relationships
**SSH Attack Template**

- **Preconditions**
  - Target has **SSH** vuln
  - Priv source >= user
  - Priv target < root
  - Source can connect to target on port **22**

- **Postcondition**
  - Attacker has priv root on target

---

**IIS Attack Template**

- **Preconditions**
  - Target has **IIS** vuln
  - Priv source >= user
  - Priv target < root
  - Source can connect to target on port **80**

- **Postcondition**
  - Attacker has priv root on target
Abstract exploit templates eliminate most redundancy

Currently models
- Privilege escalation
- Password guessing
- Information leaks
- Altering firewall and router rules

R2R Attack Template

Preconditions
- Target has R2R vuln
- Priv source >= user
- Priv target < root
- Source can connect to target on port r2r

Postcondition
- Attacker has priv root on target
Attack Graphs

Generation
• **Input data**
  - List of vulnerabilities present on all machines
  - Model of firewall and router rules

• **Attacker model**
  - Assumes a single attacker for each graph
  - Initial privileges attacker has on all machines
  - Additional “attacker” machines
  - Can model insider and outsider scenarios
● Preprocessing
  - Convert all vulnerabilities and port numbers to abstract model
  - Cluster identical machines
    • Must have same vulnerabilities AND connectivity
    • Less work for the generator

● Generation
  - Use expert system to discover all possible exploit paths
• Outputs graph as data file and visualized graph
• Visual complexity can rise quickly

Attack graph for network with 15 hosts:
Analysis of Attack Graphs

Evolutionary Method

Dr. Melissa Danforth
Department of Computer Science
California State University, Bakersfield
• Goal: Prevent attacker from achieving certain resources (“goal nodes”) in graph

• Evolutionary Method
  – Computationally infeasible to brute force
  – Start with random solutions
    • Solution varies with analysis mode
  – Use genetic algorithm to refine solutions
    • Guided search of solution space
  – Flexible and allows multiple solutions
Example: Find a set of patches

- Initial solutions are random subset of patches
- Applies patches to graph and sees how well the patches disconnect the goal nodes
- Assign a fitness metric
- Select solutions with best fitness
- “Breed” them to create next generation
- Repeat
• Assessing fitness is most CPU intensive task
• Must apply each hardening measure and cascade its effects throughout the graph
• Over 60% of the single-threaded application CPU time was spent in this function
• Switched this task to multi-threaded function
  – Each has its own copy of the attack graph
  – Memory is cheap, time is not (usually)
• Fitness metric measures benefit of solution and cost of solution
  – Affected by mode of analysis and policy
• Policy model allows defaults specified by mode to be overridden
  – Can override both costs and benefits for specific cases or general cases
  – Can have a different policy for different modes of analysis
Analysis of Attack Graphs

Modes of Analysis
• Find set of hardening measures
  – Prevent attacker from reaching resources by patching machines, applying new firewall or router rules and/or placing IDS sensors
  – Can also be run in “patch only” mode
  – Solution is a proposed set of measures
  – Fitness metric based on cost for measures in set and how well they disconnect the attacker from the goal nodes
• Strategic Planning
  – Assess unknown risks by asking “what if”
  – Affects the generation of the attack graph
  – Alter the vulnerability list or firewall/router rules to reflect the scenario
  – Generate an attack graph for the scenario
  – Analyze resulting graph using any other mode
• Network Design – Simple mode
  – Administrator designs several different sets of firewall and/or router rules for the network
  – Attack graph is generated for each design
  – Risk metric is calculated based on how well connected the goal nodes are to the graph
  – Design with lowest risk metric is selected

• Simple mode is not very interesting
  – Just a variation on strategic planning
• Network Design – Evolutionary Mode
  - Administrator gives a single prototype design
  - Evolutionary analysis seeks improvements
  - Solutions alter firewall/router rules or place IDS sensors
  - Fitness metric based on how well goal nodes are disconnected or watched
  - Outputs several designs that minimize both risk and cost
• Forensic Evaluation and IDS Alerts
  – Match forensic evidence and/or IDS alerts to nodes in graph
  – Detect exploit paths in use by attacker
  – Forensic evaluation – Guides analyst by highlighting other resources the attacker may have compromised
  – IDS alerts – Integrate with intrusion response or activate additional monitoring
Experimental Results

CSU Bakersfield
Computer Science Department
Instructional Laboratory Network

Dr. Melissa Danforth
Department of Computer Science
California State University, Bakersfield
• **Base Configuration Scenario**
  - Attacker is an outsider

• **Strategic Planning Scenarios**
  - Student visits a malicious website with a vulnerable version of Firefox
  - A malicious student attacks the network from one of the instructional lab machines
  - An instructor brings in a compromised laptop and plugs it into the LAN
• Base Configuration Original Graph
• Base Configuration Patched Graph
• Vulnerable Browser Original Graph
• Vulnerable Browser Patched Graph
• Malicious Student Original Graph
• Malicious Student Patched Graph
- Rogue Laptop Original Graph
● Rogue Laptop Patched Graph
• Rogue Laptop Redesigned Network Graph
• Scalability Testing
  - Generated networks with 5 to 2500 machines
  - Largest network took 1.5 hours to analyze on a quad-core Xeon 2.33GHz system
  - Smallest network took approximately 1 second
  - Larger networks have more complex attack graphs, so they take longer to analyze even with clustering and abstract exploit templates
Future Work
• Automate remaining “by hand” processes
  – Importing firewall and router rules
  – Translating Nessus plugin IDs to abstract exploit class names
• Allow multiple attacks in attacker model
• Implement IDS correlation mode
• Improve visualization of the graphs
• Create a cohesive GUI to tie all parts together
Questions?

Students on this project:
Jonathan Berling
Fred McHale
John Millikin
Nick Toothman