Rules of the Road for Useful Security Metrics

Metricon 6.0, August 7th 2012

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

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?”

Challenges in Security Metrics

- Typical issues addressed in the literature
  - How can a database server be secured from intruders?
  - How do I stop an ongoing intrusion?
- Notice that they all have a qualitative nature
- Better questions to ask:
  - 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?
- For this we need a system security modeling and analysis tool

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

[1] Common Vulnerability Scoring System (CVSS-SIG) v2, http://www.first.org/cvss/

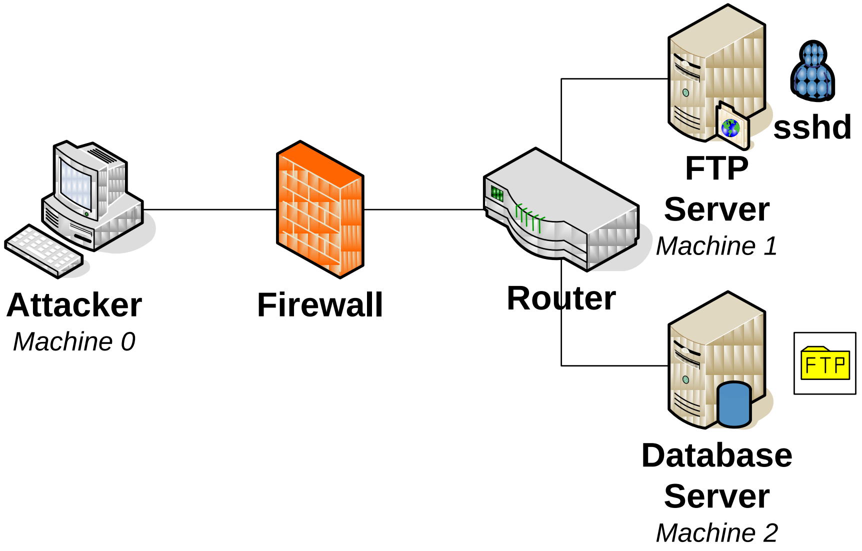
Challenges in Security Metrics

- Counting the number of vulnerabilities is not enough
  - Vulnerabilities have different importance
  - The scoring of a vulnerability is a challenge
    - Context of the Application
    - Configuration of the Application
- How to *compose* vulnerabilities for the overall security of an enterprise network system

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

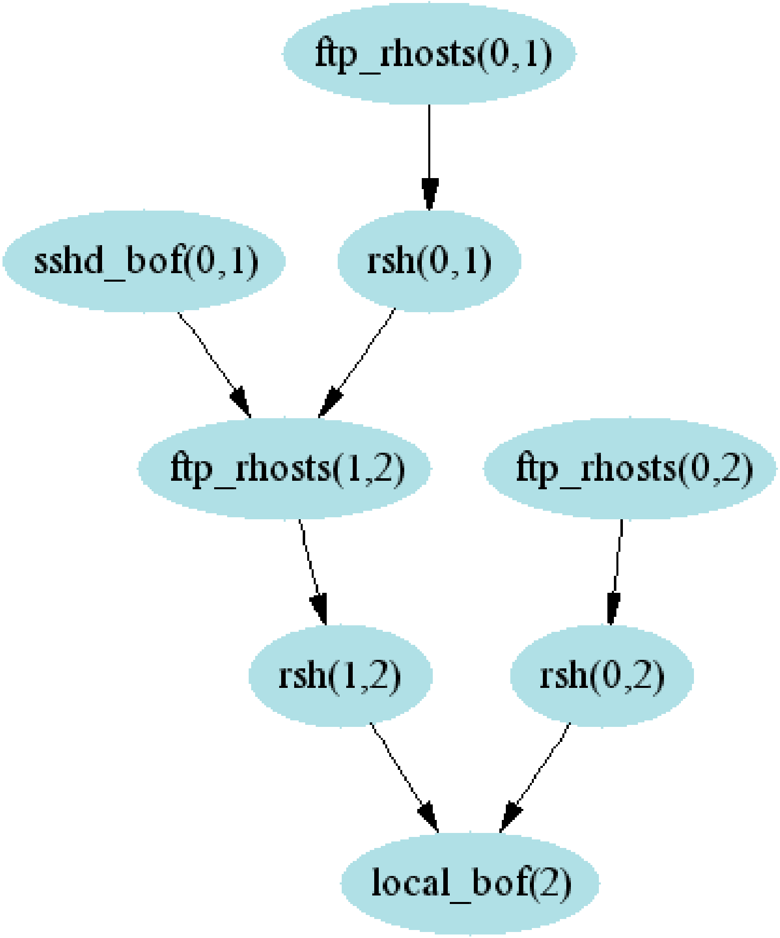
Attack Graph Example



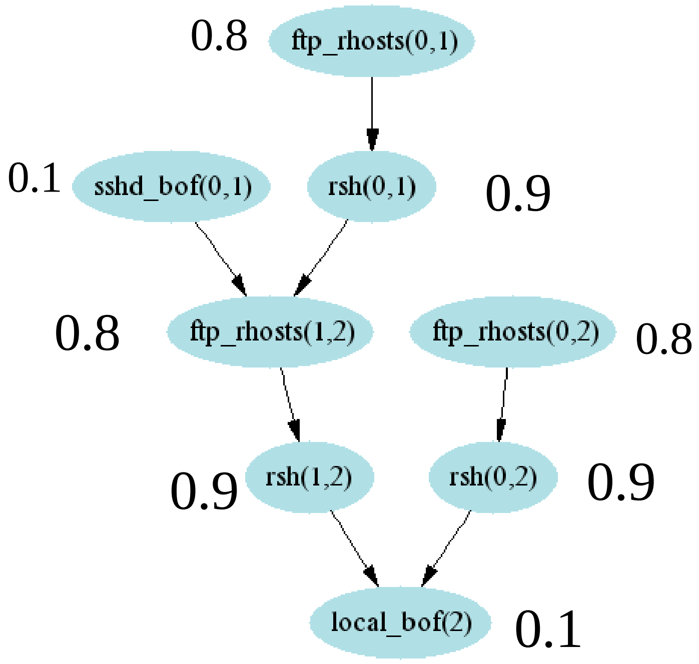
Different Paths for the Attack

- *sshd_bof(0,1)* → *ftp_rhosts(1,2)* → *rsh(1,2)* → *local_bof(2)*
- *ftp_rhosts(0,1)* → *rsh(0,1)* → *ftp_rhosts(1,2)* → *rsh(1,2)* → *local_bof(2)*
- *ftp_rhosts(0,2)* → *rsh(0,2)* → *local_bof(2)*

Attack Graph from machine 0 to DB Server

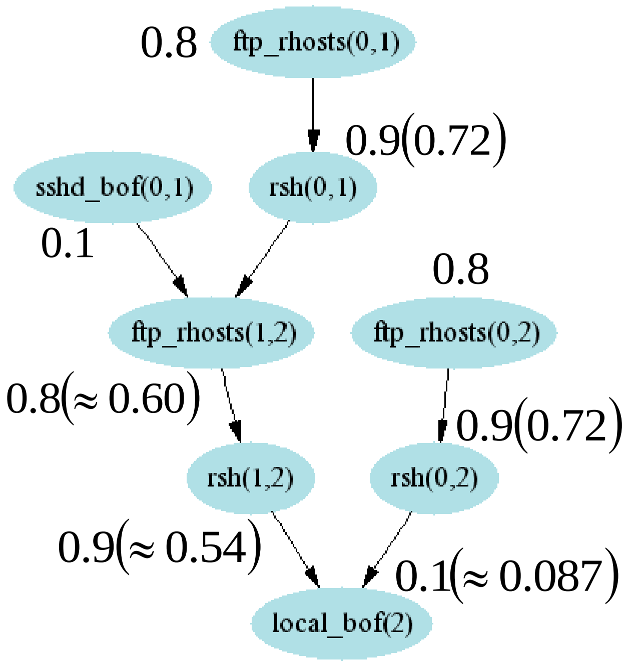


Attack Graph with Probabilities



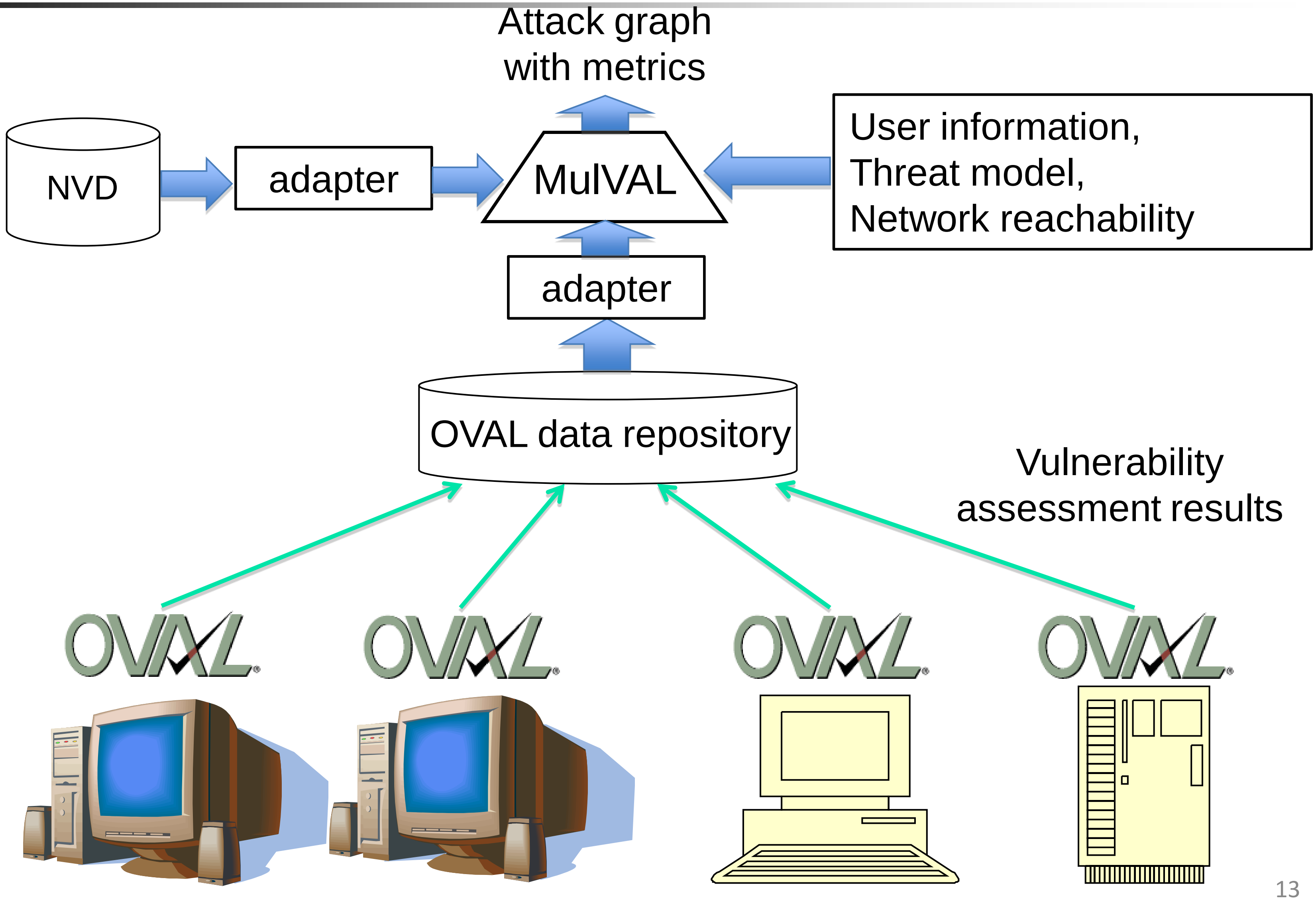
- Numbers are estimated probabilities of occurrence for individual exploits, based on their relative difficulty.
- The *ftp_rhosts* and *rsh* exploits take advantage of normal services in a clever way and do not require much attacker skill
- A bit more skill is required for *ftp_rhosts* in crafting a .rhost file.
- *sshd_bof* and *local_bof* are buffer-overflow attacks, which require more expertise.

Probabilities Propagated Through Attack Graph



- When one exploit must follow another in a path, this means **both** are needed to eventually reach the goal, so their probabilities are multiplied: $p(A \text{ and } B) = p(A)p(B)$
- When a choice of paths is possible, **either** is sufficient for reaching the goal: $p(A \text{ or } B) = p(A) + p(B) - p(A)p(B)$.

MulVAL attack-graph tool-chain



Attack graph with metrics

NVD

adapter

MulVAL

User information,
Threat model,
Network reachability

adapter

OVAL data repository

Vulnerability assessment results

Conclusions

- Based on attack graphs, we have proposed a model for security risk analysis of information systems
  - Composing individual scores to more meaningful cumulative metric for overall system security
- The metric meets intuitive requirements