A Model for Evaluation of Security Risk in Enterprise Information Systems

Anoop Singhal
National Institute of Standards and Technology

Coauthors: Lingyu Wang Steve Noel and Sushil Jajodia

Concordia University
George Mason University

Outline

- **Motivation and Related Work**
- **Example of an Attack Graph**
- **Quantitative Security and Risk Analysis**
- **Conclusion and Future Work**

Motivation

- 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

Motivation (Cont'd)

- 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/

Motivating Example

- Assume that the cost of recovering from a database breach (*incident*) is $200K
- Suppose the NIST model says that the likelihood of a database breach is 0.2
- This implies that the *risk exposure* is

  $200K * 0.2 = $40K
- This can help you decide about security investments such as firewalls or IDS
- Help you quantify whether additional security investment is justified by a reduction in expected losses

Challenges for Quantitative Analysis

- 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 a network system

Sample Vulnerability

symantec.

Symantec Vulnerability Alert

Microsoft SQL Server 2000 Resolution Service Denial of Service Vulnerability

Synopsis

Bugtraq ID: **5312** CVE: CAN-2002-0650

Published: July 25 2002 Last Update: July 25 2002 GMT

Last Update: Initial analysis.

Remote: Yes Local: No

Availability: Always Authentication: Not Required

Ease: No Exploit Available

Vulnerability Classification: Failure to Handle Exceptional Conditions

Urgency Rating: **6.1**

Threat Breakdown:

Severity	6.7
Impact	4.0
Ease of Exploit	1
Credibility	Vendor Confirmed

Related Work

- Efforts by NIST on standardizing security metric
  - Special publication 500-133 1985, 800-55 2003
  - NVD and CVSS v2

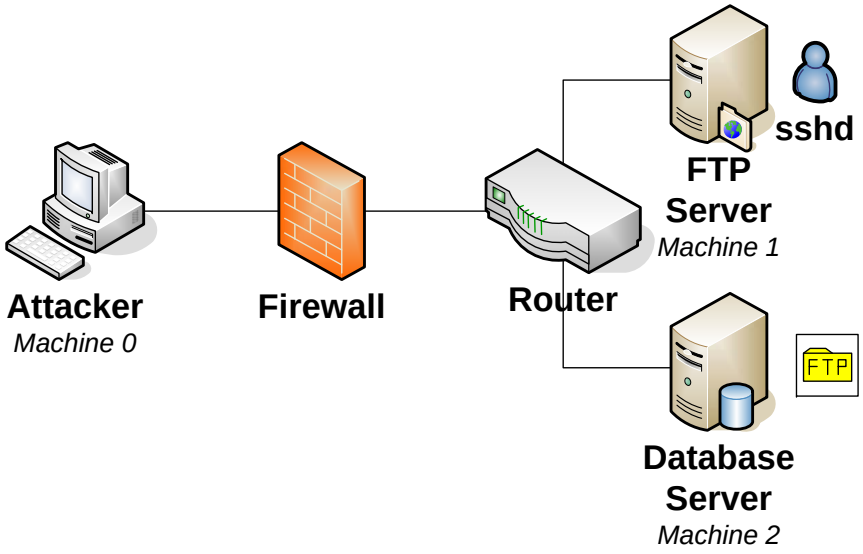
Related Work (Cont'd)

- Attack graph
  - Model checker-based (Ritchey et. al S&P'00, Sheyner et. al S&P'02)
  - Graph-based (Noel et. al ACSAC'03, Singhal et. al DBSEC'06, DBSEC '07)

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

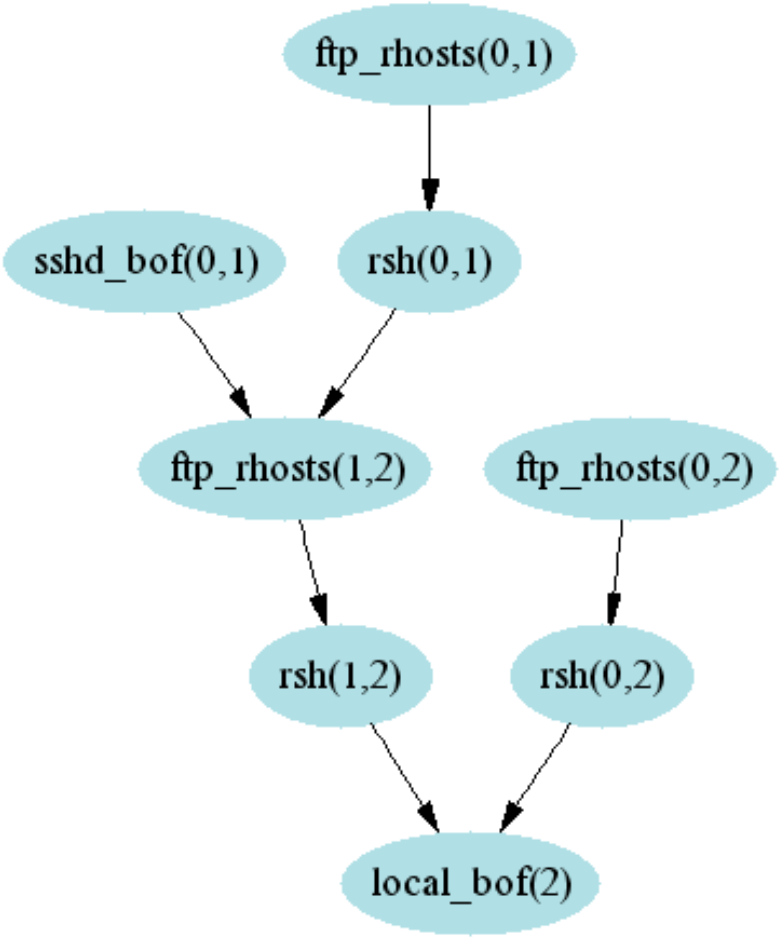
Attack Graph Example



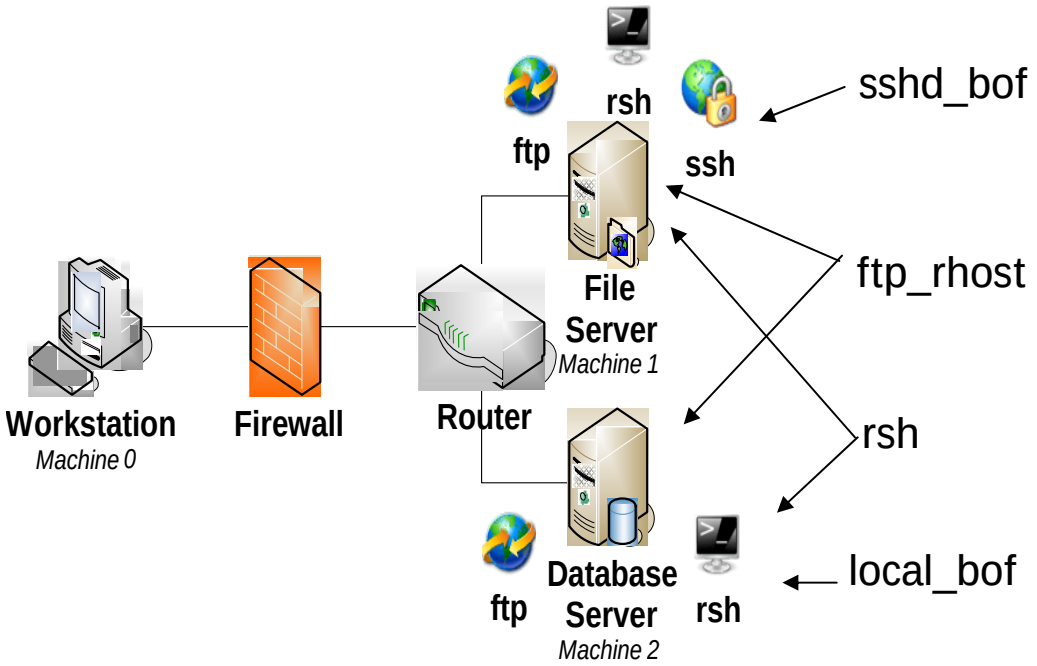
Different Paths for the Attack

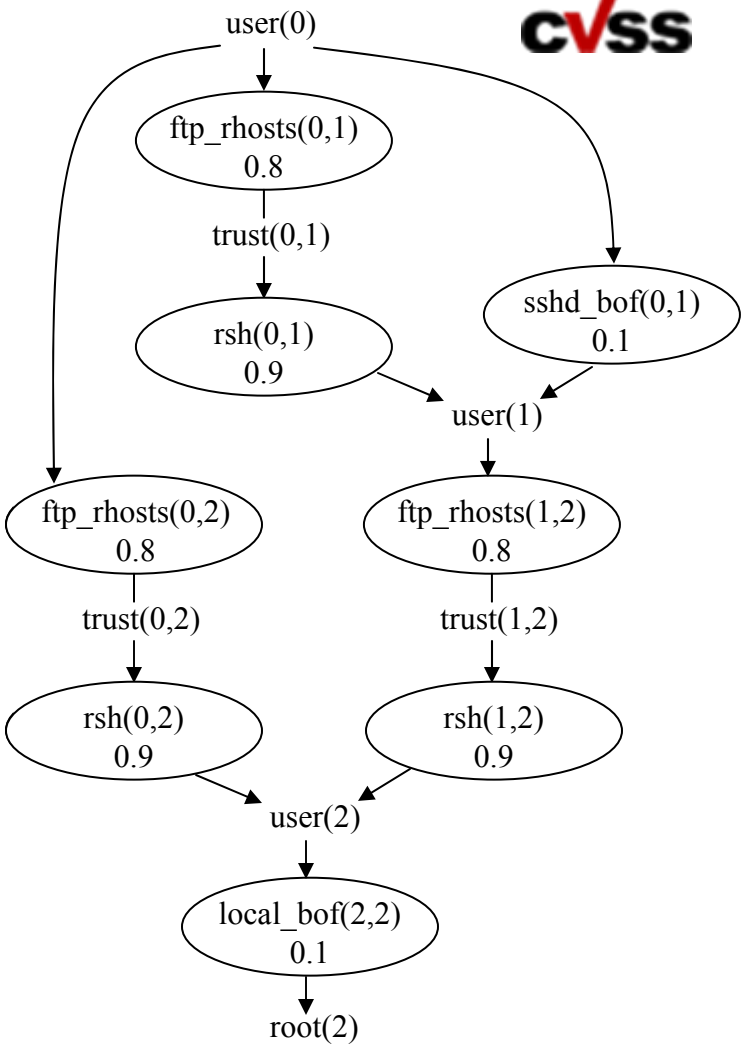
- *sshd_bof(0,1) → ftp_rhosts(1,2) → rsh(1,2) → local_bof(2)*
- *ftp_rhosts(0,1) → rsh(0,1) → ftp_rhosts(1,2) → rsh(1,2) → local_bof(2)*
- *ftp_rhosts(0,2) → rsh(0,2) → local_bof(2)*

Attack Graph from machine 0 to DB Server

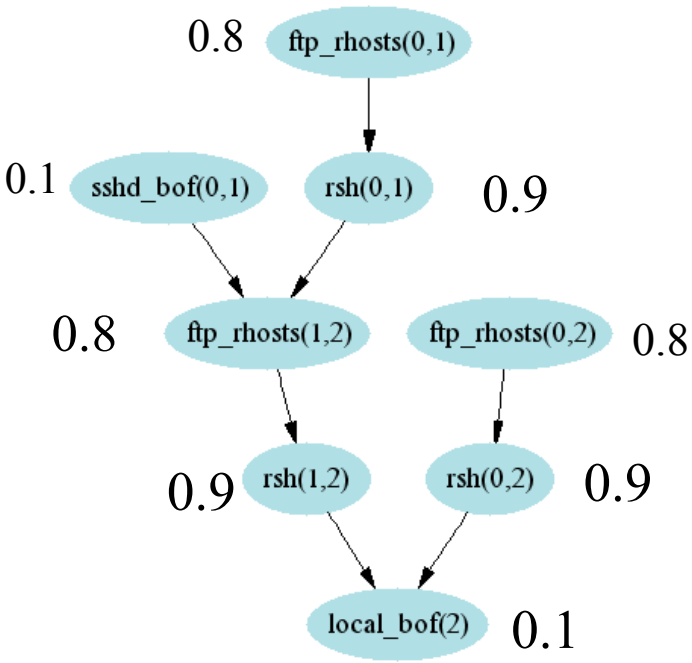


Example


rsh
ftp
ssh
sshd_bof
File
Server
Machine 1
ftp_rhost
Workstation
Machine 0
Firewall
Router
rsh
ftp
Database
Server
Machine 2
rsh
local_bof

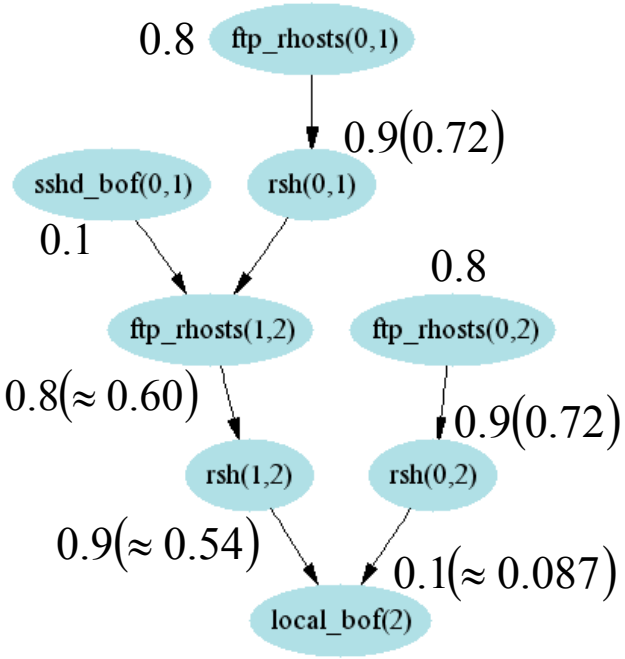

user(0)
ftp_rhosts(0,1)
0.8
trust(0,1)
rsh(0,1)
0.9
sshd_bof(0,1)
0.1
user(1)
ftp_rhosts(0,2)
0.8
ftp_rhosts(1,2)
0.8
trust(0,2)
trust(1,2)
rsh(0,2)
0.9
rsh(1,2)
0.9
user(2)
local_bof(2,2)
0.1
root(2)

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: $p(A \text{ or } B) = p(A) + p(B) - p(A)p(B)$.

Network Hardening

- When we harden the network, this changes the attack graph, along with the way its probabilities are propagated.
- Our options to block traffic from the *Attacker*:
  - Make no change to the network (baseline)
  - Block ftp traffic to prevent *ftp_rhosts(0,1)* and *ftp_rhosts(0,2)*
  - Block rsh traffic to prevent *rsh(0,1)* and *rsh(0,2)*
  - Block ssh traffic to prevent *sshd_bof(0,1)*

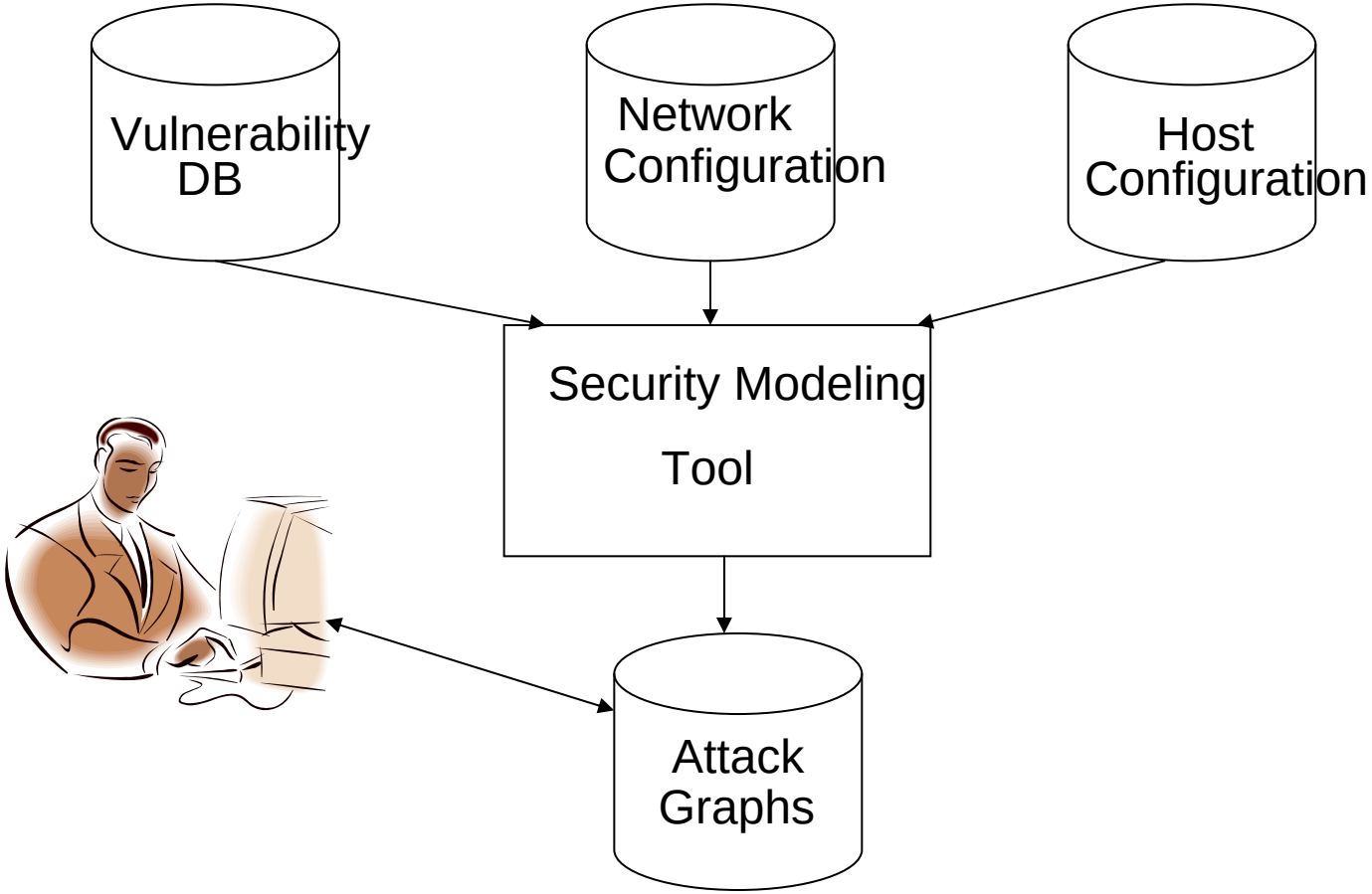
Comparison of Options

- We can make comparisons of relative security among the options
- Make no change p=0.1
- Blocking rsh traffic from *Attacker* leaves a remaining 4-step attack path with total probability $p = 0.1 \cdot 0.8 \cdot 0.9 \cdot 0.1 = 0.0072$
- Blocking ftp traffic, p=0.0072
- But blocking ssh traffic leaves 2 attack paths, with total probability $p \approx 0.0865$, i.e., compromise is *10 times more likely* as compared to blocking rsh or ftp.

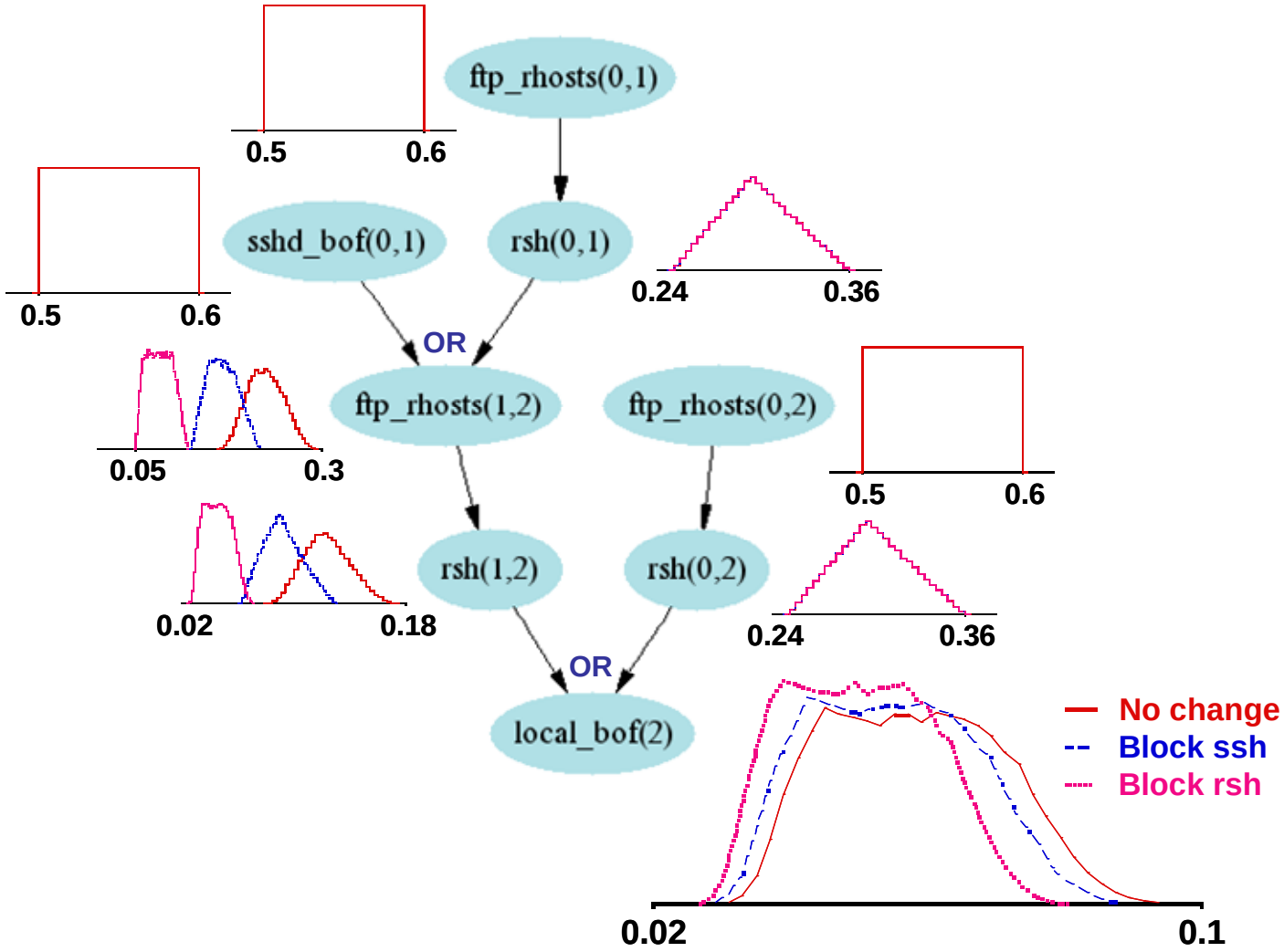
Need for a Modeling Tool

- For a large enterprise network that has hundreds of host machines and several services we need a modeling tool that can
  - Generate the attack graph
  - Use the attack graph for quantitative analysis of the current configuration
  - Help the network administrators to decide what changes to make to improve security

System Architecture



Probability Of DB Compromise for Each Choice



Conclusions

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

Future Work

- Implement a prototype system and validate the output
- Generalize the model to use CVSS Base and Temporal scores for each vulnerability
- Generalize the model when exploits are related (in addition to the causal relationship)