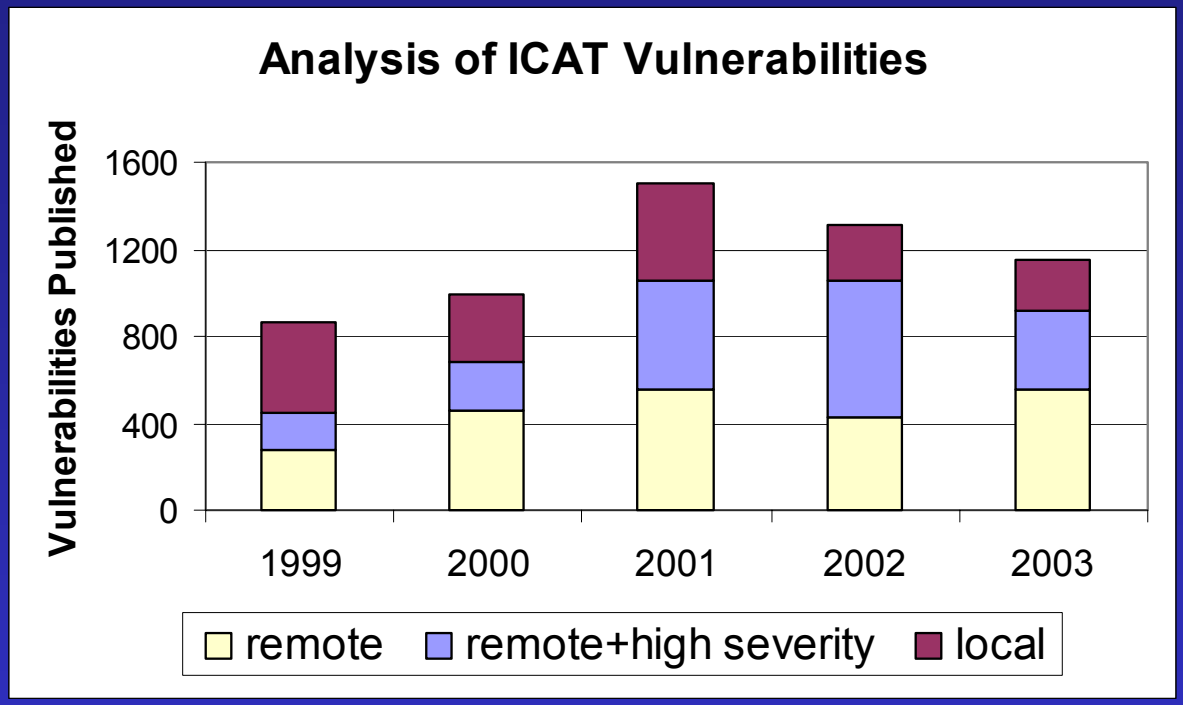
Foundation for Intrusion Prevention

Shai Rubin,
Ian D. Alderman, David W. Parter, and Mary K. Vernon

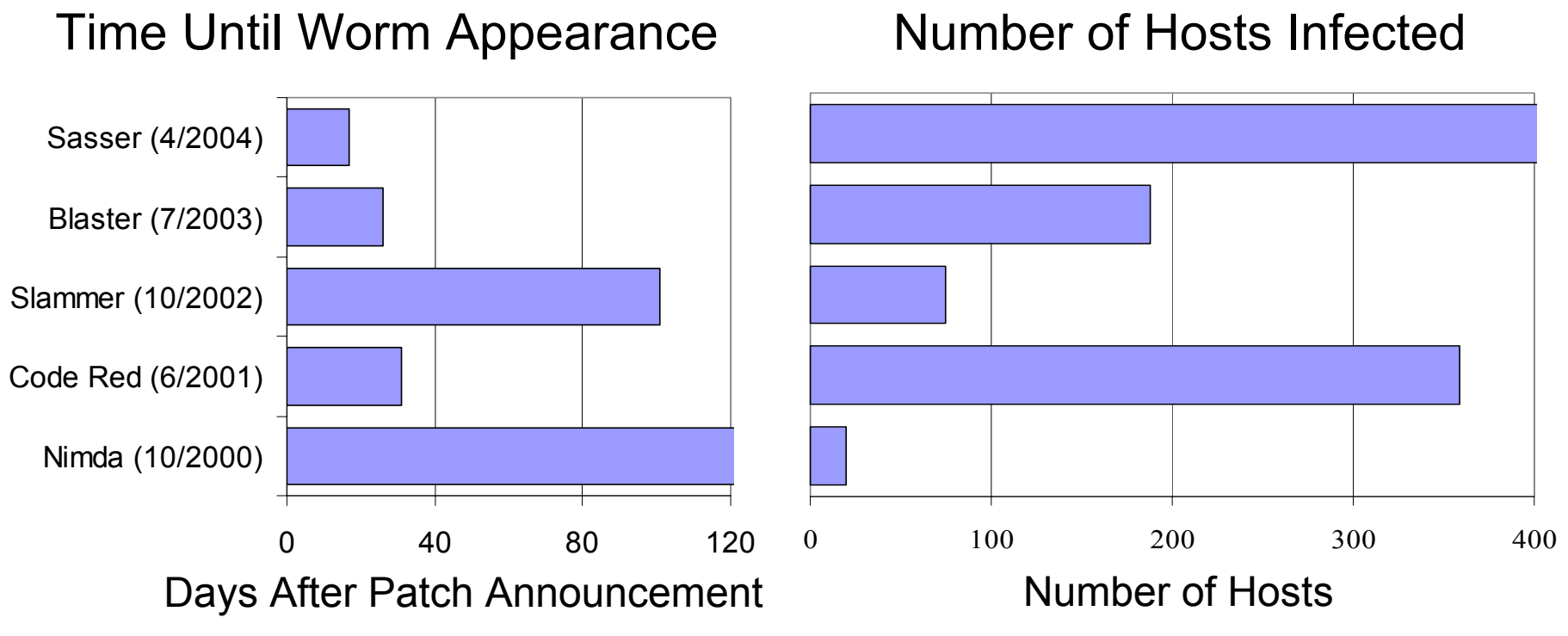
University of Wisconsin, Madison, USA

Rate of New Vulnerabilities



- Remote vulnerability: exploit done over the network
- New vulnerabilities (2002, 2003):
  - ~80% of vulnerabilities are remote
  - ~40% of vulnerabilities are remote + high severity
  - ~2 new remote vulnerabilities per day

Threat from Known Vulnerabilities



- Intrusion Prevention: repair before attack.
  - Identify all vulnerable hosts
  - Understand the severity of the threat to your system
  - Reliably repair all vulnerable hosts

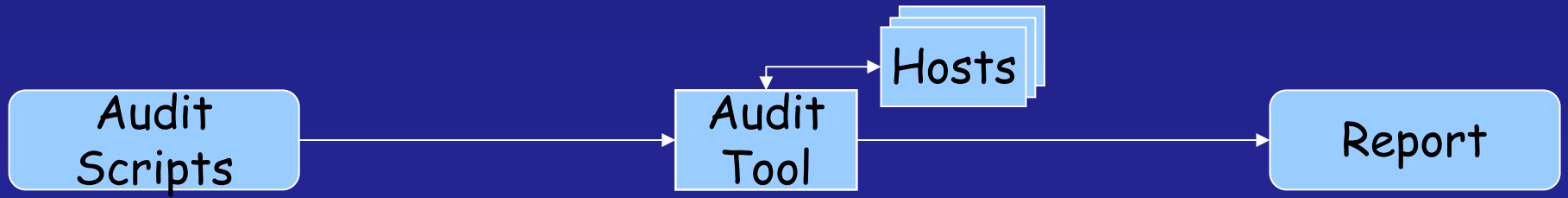
In this talk

Intrusion prevention

- Quickly identify all vulnerable hosts
- Estimate severity of exposure to your system
- Reliably repair all vulnerable hosts

- Deficiencies of current identification process
- Envisioned intrusion prevention infrastructure
- Pilot study

Today's Network Audit Process



- **Slow deployment**: scripts are written manually
- **Many false positives**: some vulnerabilities cannot be exclusively determined over the network
- Ambiguous severity ratings: report does not quantify the severity of the threat to your site
- Ambiguous vulnerability specification: hard to tell what the vulnerability is from the script or description

Guestbook Vulnerability

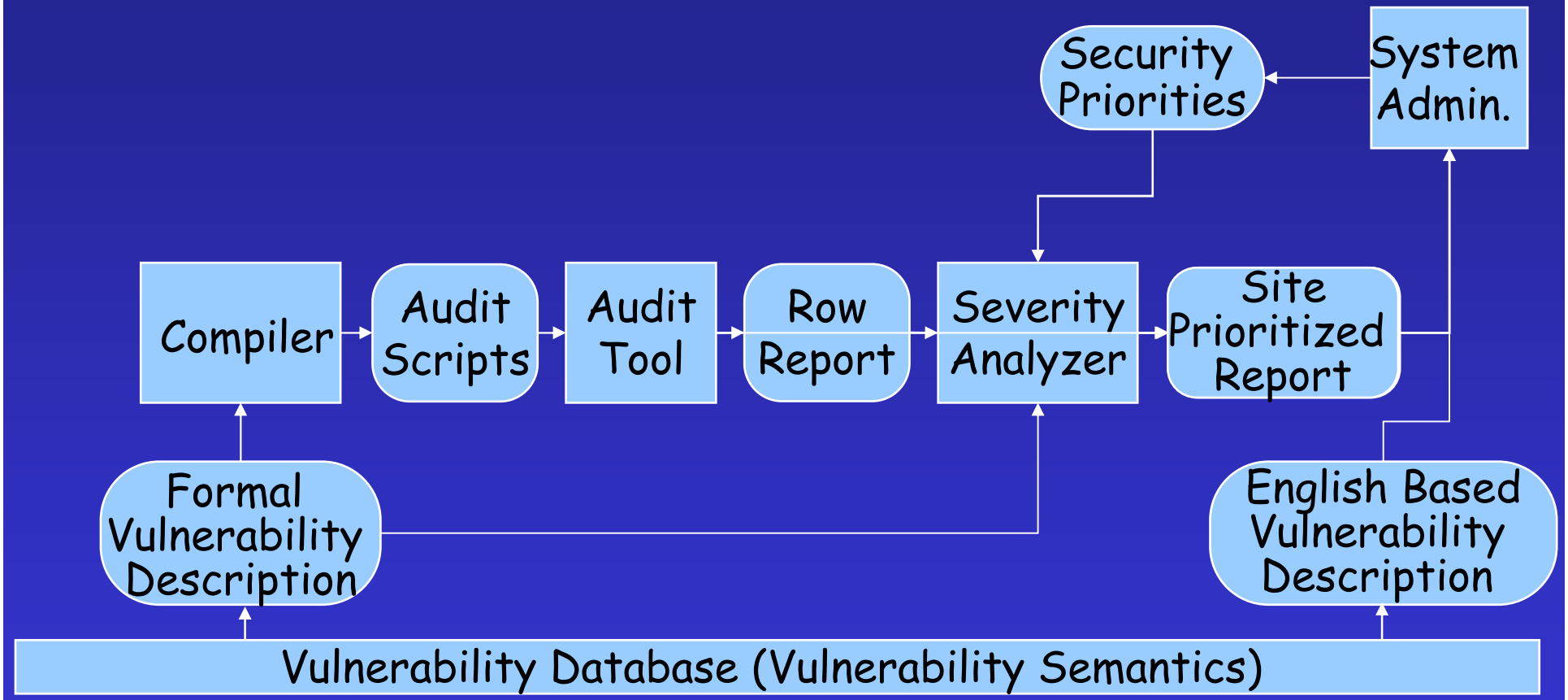
From the ICAT database (**CAN-1999-1053**):

guestbook.pl cleanses user-inserted SSI commands by removing text between "<!--" and "-->" separators, which allows remote attackers to execute arbitrary commands when guestbook.pl is run on Apache 1.3.9 and possibly other versions, since Apache allows other closing sequences besides "-->".

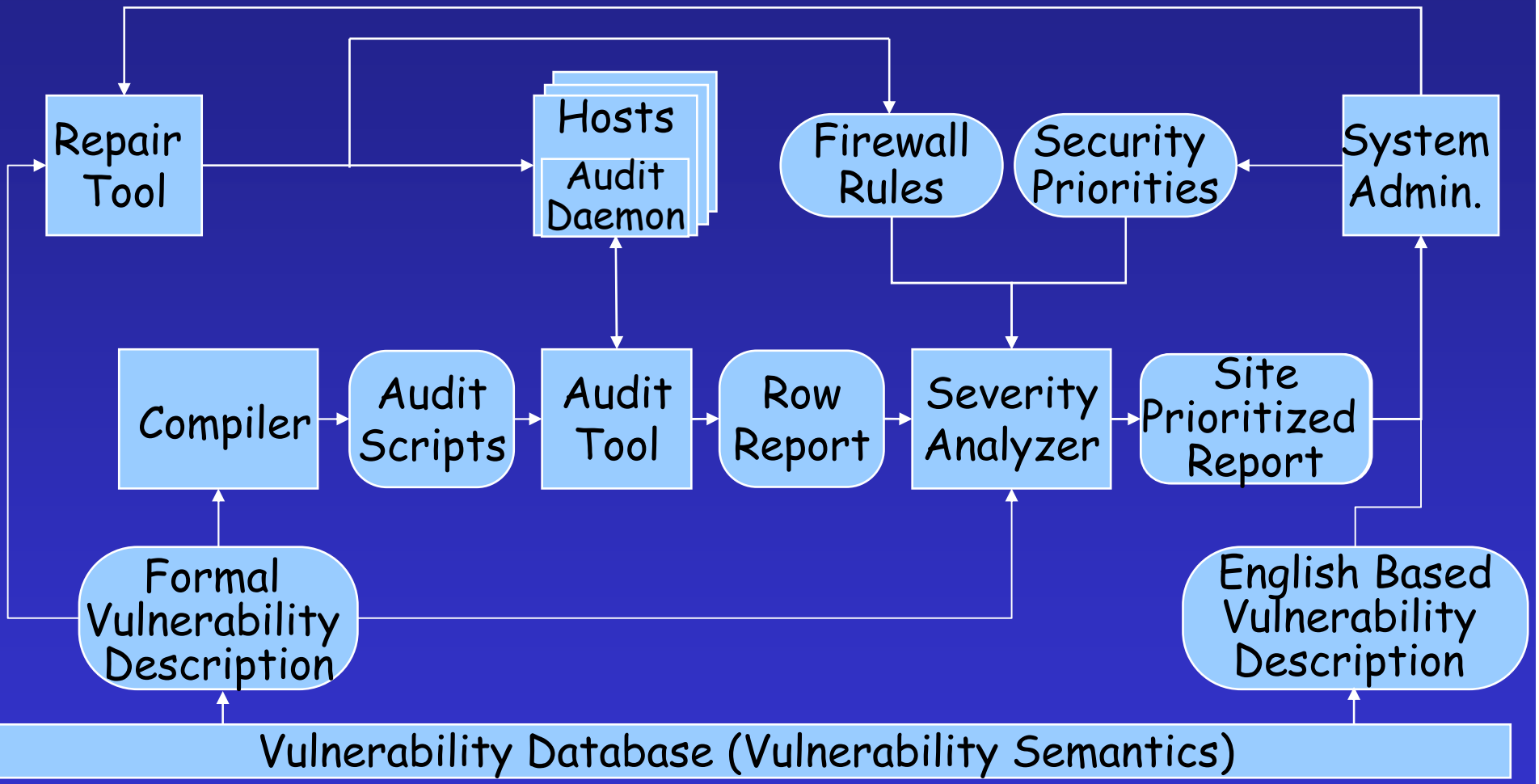
Severity: High

- Misleading: actually the problem is that the script does not cleans SSI
- Incomplete: XBitHack must be set (SSI enabled)
- Provides unnecessary details: "<!--"
- Ambiguous specification of vulnerable host:
  - vulnerable if XBitHack is set (SSI enabled)
  - more vulnerable if guestbook.pl is installed

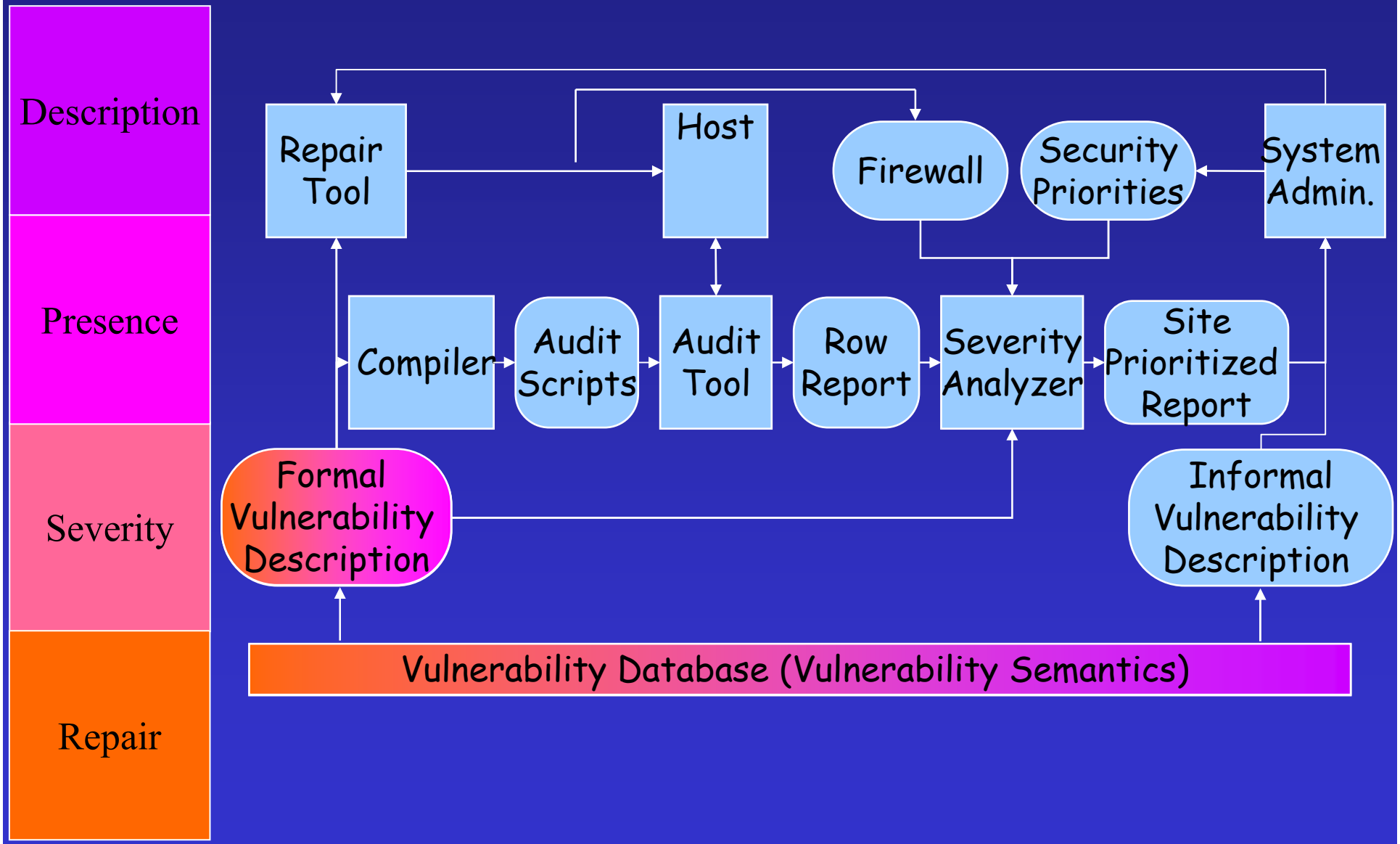
Intrusion Prevention, Vision



Intrusion Prevention, Vision



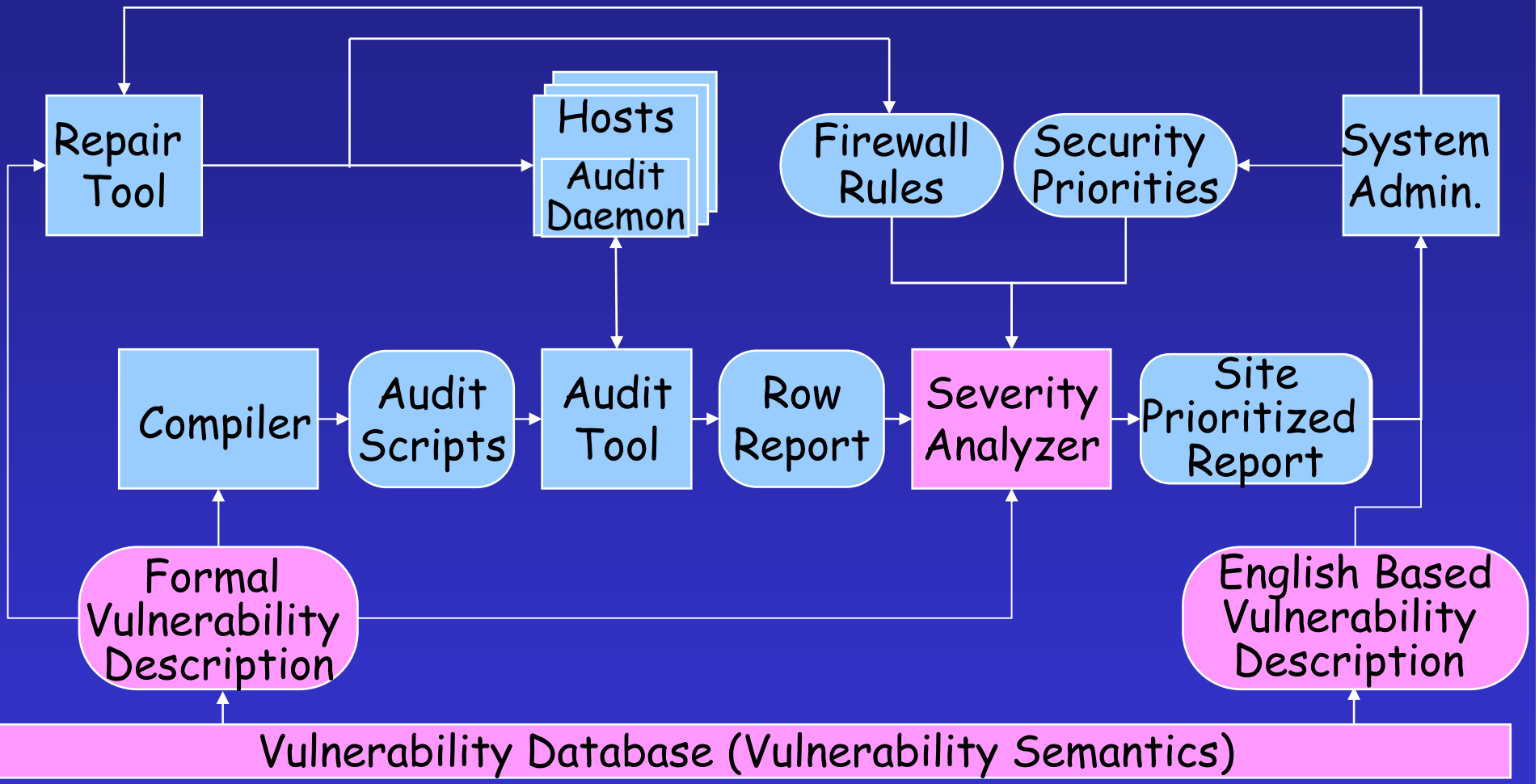
Vulnerability Semantics



Guestbook Semantics

Description	Name	*Apache GuestBook (CAN-1999-1053)*
	OS	*ANY*
	Vulnerable unit	*Apache version 1.3.9, guestbook.pl.*
	Configuration	*Server Sides Include (SSI) on.*
	Protocol,Port	*RFC: 2616/HTTP,80+'any'*
Presence	Condition Set	P_1: *serviceRunning()* P_2: *package=Apache* P_3: *content(config_file, [Includes|XBitHack])*
	Verification Hints	$H_{3,UNIX}$: *config_file= /etc/httpd/conf/httpd.conf'*
Severity	Exposed Resources	*if (version=1.3.9) then <SA,CIA>* *else UNKNOWN*
	Expected Time to Exposure (days)	*if (access(guestbook.pl) or* *content(guestbook.pl, 'html=1')) then 0;* *else TimeUntil(guestbook.pl installed)=30*
	Expected Time to Attack (days)	*7*

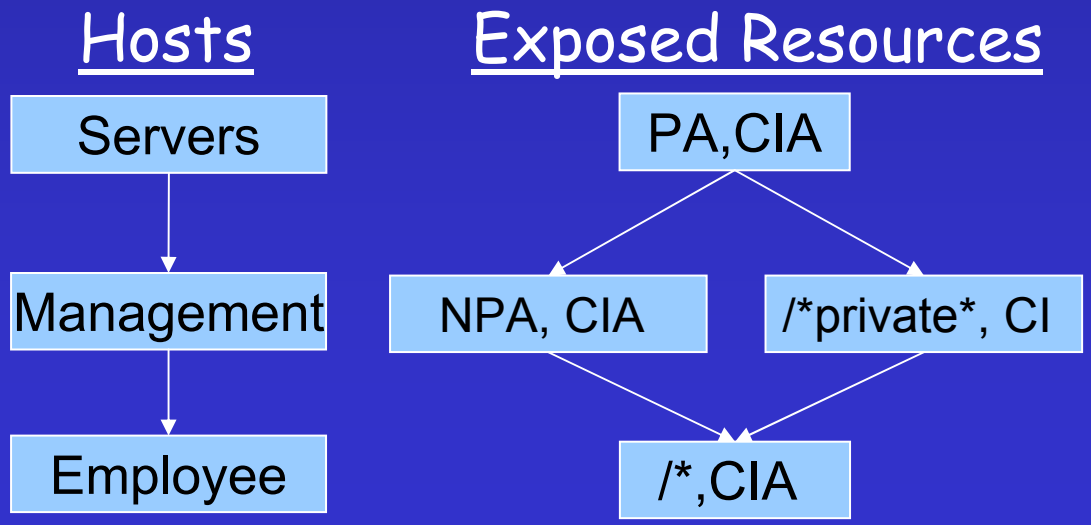
Intrusion Prevention, Vision



Severity Analysis

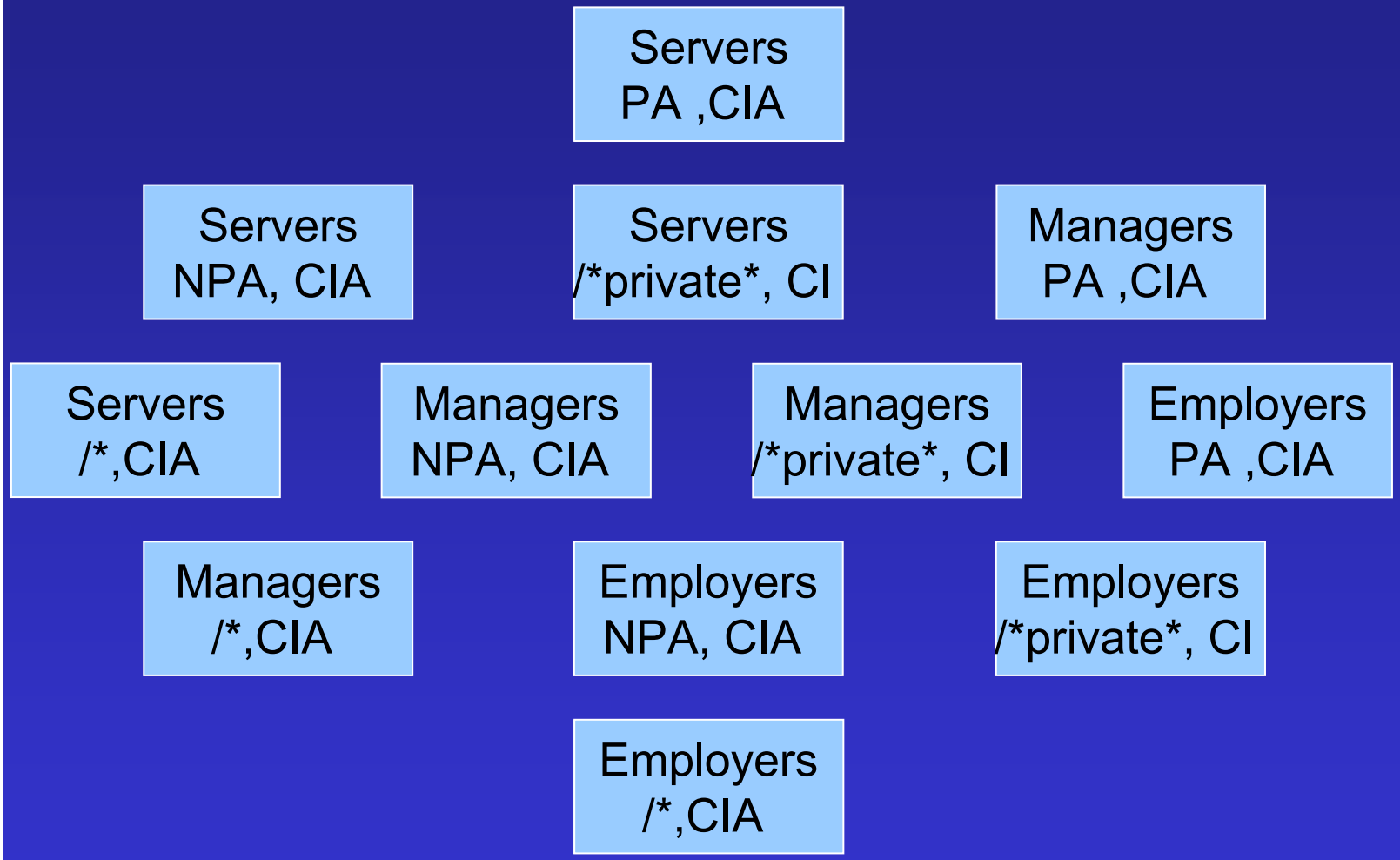
- Severity is based on: host type, exposes resource type, difficulty to exploit the vulnerability, site policy

1. qualitatively rank sets of hosts according to the site security priorities
2. qualitatively rank sets of exposed resources according to the site security priorities



PA = Privileged Account
NPA= Non Privileged Account
C = Confidentiality
I = Integrity
A = Availability

Quantifying Severity



3. relatively quantifying the two rankings

Pilot Study

- Goal: evaluate the feasibility and impact of the Intrusion Prevention Infrastructure
  - Evaluate ease and effectiveness of finding/repairing vulnerabilities
  - define 300+ vulnerabilities using our presence and severity semantics
  - Evaluate use of severity semantics and analysis to quantify the relative value of site hosts and resources

- Site: network with ~1500 hosts,
  strong configuration management,
  dedicated security administrator

Pilot Study: Methodology

- simulated accurate audit:
  - modeled each Nessus vulnerability using the presence and severity semantics
  - manually removed all false positives from Nessus report
- simulated severity analysis:
  - System administrator at site defined a severity model for each type of resource on each type of hosts
- Two phase experiment:
  - site security "baseline": 5 audits, one per month
  - only the 5th audit results are given to the site admin.
  - 6th audit one month after results were revealed

Site Prioritized Report (5th month)

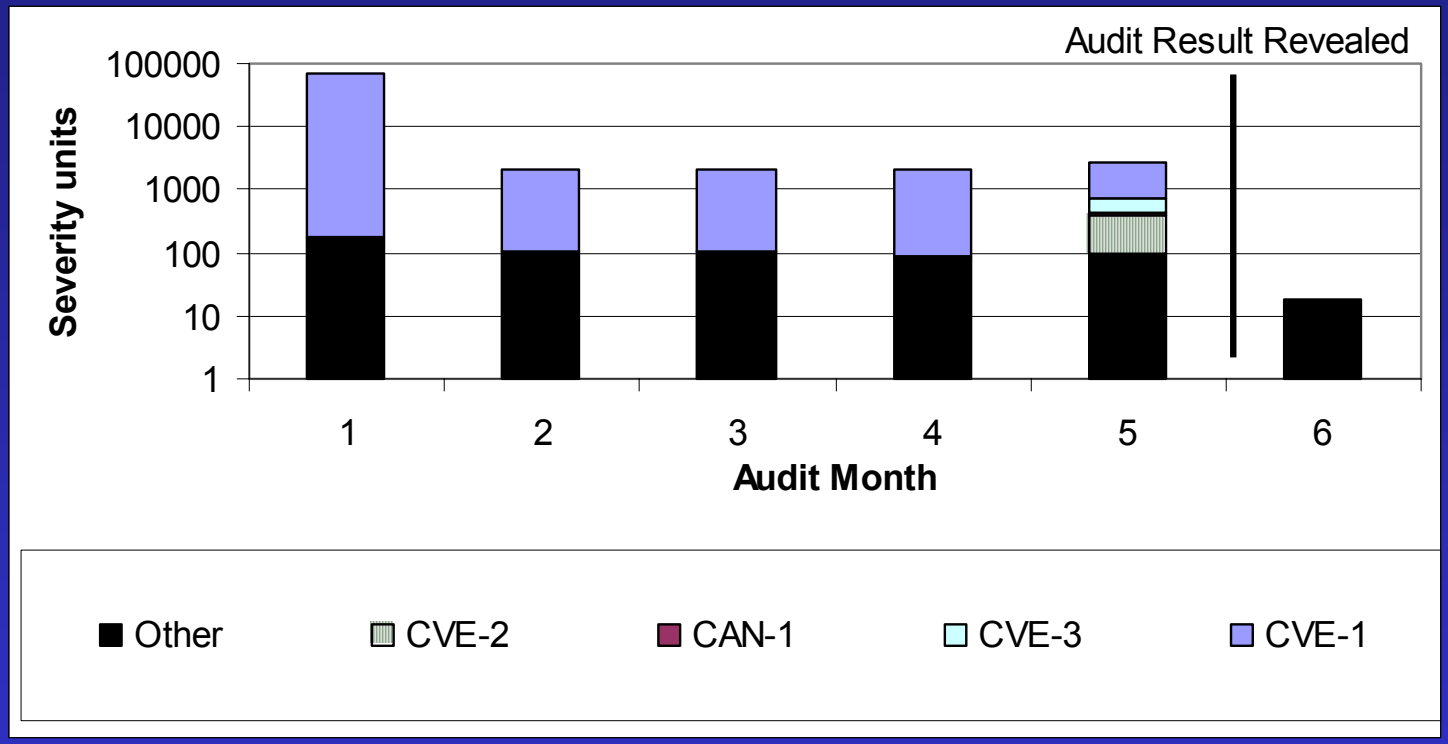
			Severity Analysis			Repair			
Name	Ser-vice	Exposed Resources	Ser-vers	Work Stations	Seve-rity	U	R	B	Comments
CVE-1	ssh	< *CIA,PA* >	2	8	2800	√	√	√	buffer overflow
CVE-2	ftp	< *CIA,SA* >	7	11	410	√	-	√	buffer overflow
CVE-3	ftp	< *CIA,NPA* >	3	-	400	√	-	√	buffer overflow
CVE-5	ftp	< *CIA,PA* >	-	3	300	√	-	√	buffer overflow
CAN-1	finger	< *CIA,PA* >	-	1	100	√	-	√	easy password

PA - Privileged Account,
NPA - Non Privileged Account
SA - Service Account

U - Upgrade
R - Reconfig
B - Block

- Advantages:
  - Concise
  - Fine grained severity estimation
  - Severity estimation combines inherent vulnerability severity with site dependent priorities

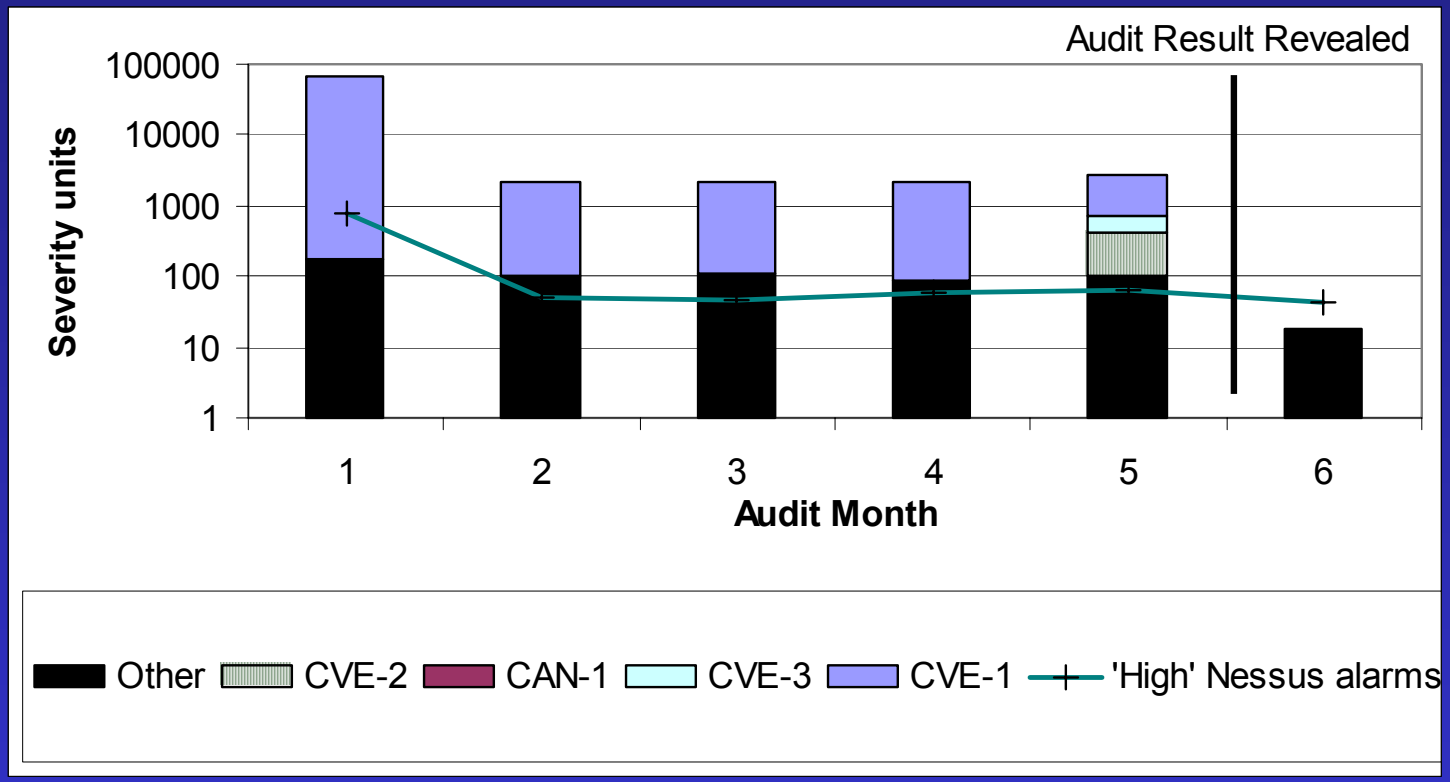
Quantifying Severity



Accurate audit:

- pinpointed severe vulnerabilities (highly alert site)
- improved security practices (no new vulnerabilities in 6)

Customized vs. non-Customized Report



Proposed severity model better represents the administrator's security priorities

What to Take Home

- Intrusion prevention = vulnerable host identification + severity estimation + reliable repair
- Precise vulnerability semantics is necessary to facilitate these three tasks
- Frequent audits find vulnerabilities that the admin missed
- Severity analysis and prioritized report help the admin to understand the severity of the threat to their system
- Not in this talk (but in the paper) severity semantics, and difficulty semantics. Not in this work: repair process

Questions?