Network Vulnerability Measurement – A Novel Approach

Lee Badger
Tim Grance
Karen Scarfone

Dec. 3, 2009

Measurement Scales and Bold Assertions

Recall: (simplified)

Ratio	A zero point exists where none of the attribute is present
Interval	Magnitudes of differences between values are meaningful
Ordinal	Values have **<**, **>**, and **=** relationships
Nominal	values have no firm numerical ordering, but **=** scale values mean equal attribute values Credit: S.S. Stevens, Wikipedia

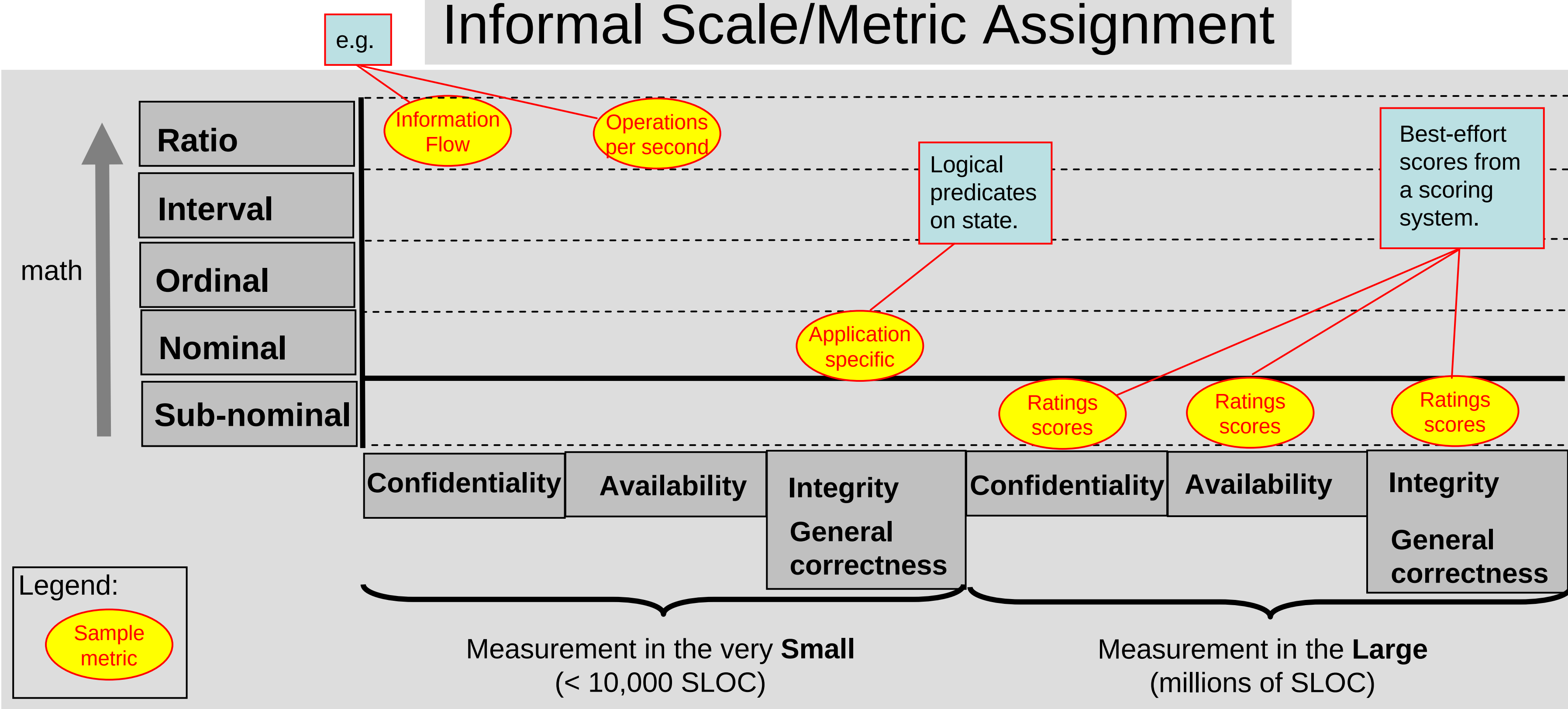
Measurement Scales and Bold Assertions

Recall: (simplified)

Ratio	A zero point exists where none of the attribute is present
Interval	Magnitudes of differences between values are meaningful
Ordinal	Values have **<**, **>**, and **=** relationships
Nominal	values have no firm numerical ordering, but **=** scale values mean equal attribute values

Credit: S.S. Stevens, Wikipedia

Informal Scale/Metric Assignment

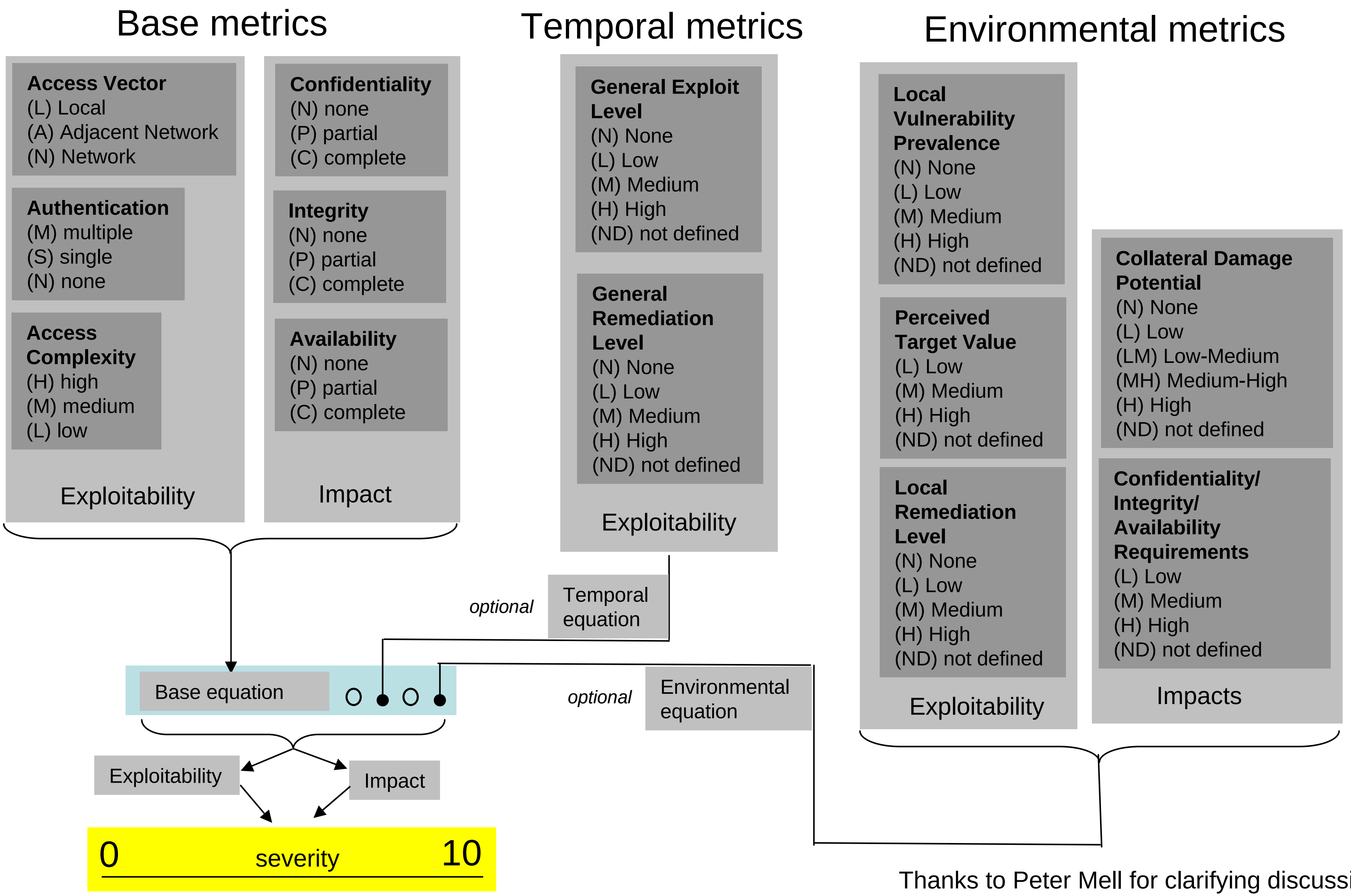


3 NIST Scoring Systems

Available at: http://csrc.nist.gov/publications

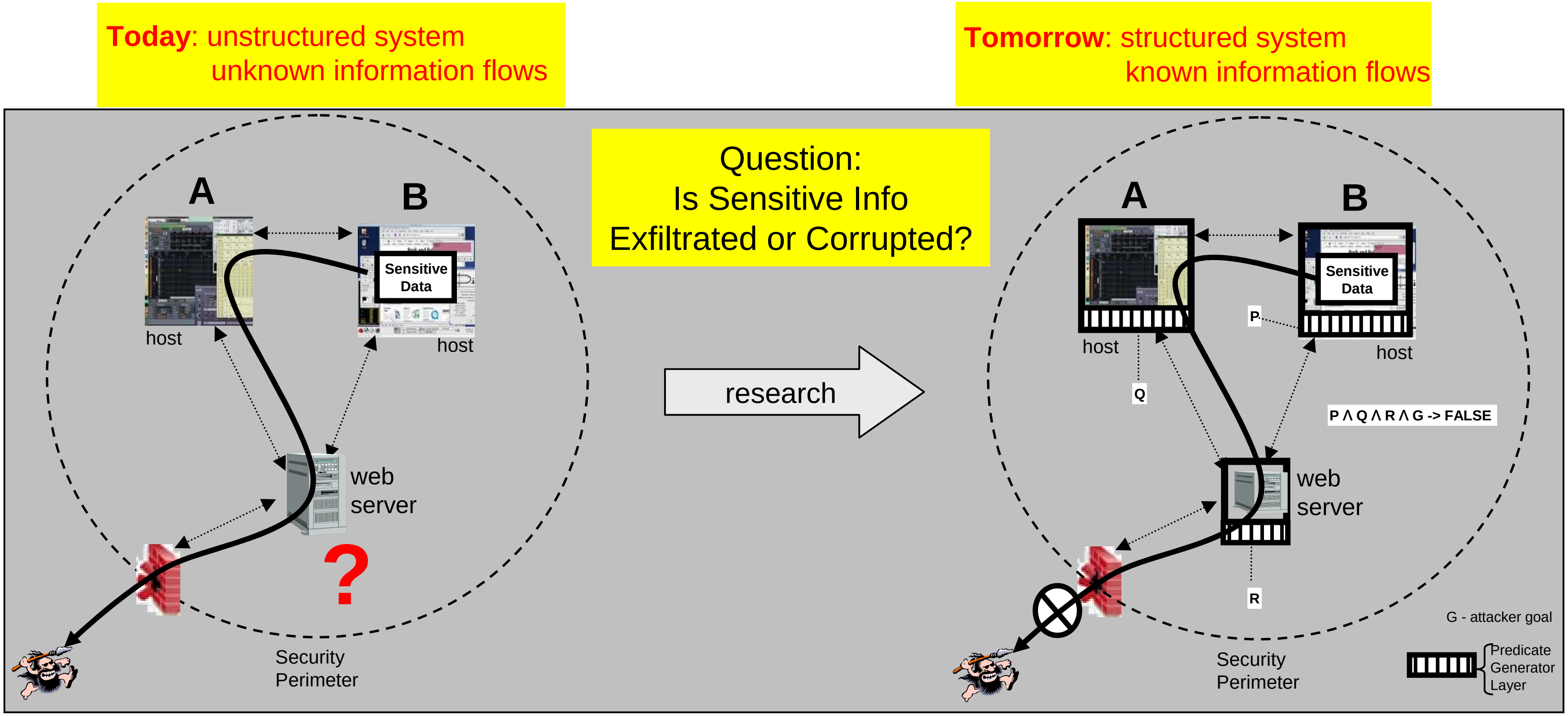
Acronym	Title	NIST #	Comments
CVSS	The Common Vulnerability Scoring System…	IR 7435	Method to express the characteristics and impacts of software flaw vulnerabilities. The scoring basis for the National Vulnerability Database, maintained at NIST (nvd.nist.gov).
CCSS	The Common Configuration Scoring System (DRAFT)	IR 7502	Method to measure the vulnerability of security settings of a system.
CMSS	The Common Misuse Scoring System… (DRAFT)	IR 7517	Method to measure the vulnerability of the **intentional functions** of a system. Measure trust assumptions.

The Common Misuse Scoring System (CMSS)



Thanks to Peter Mell for clarifying discussions.

Metrics Idea in a Nutshell



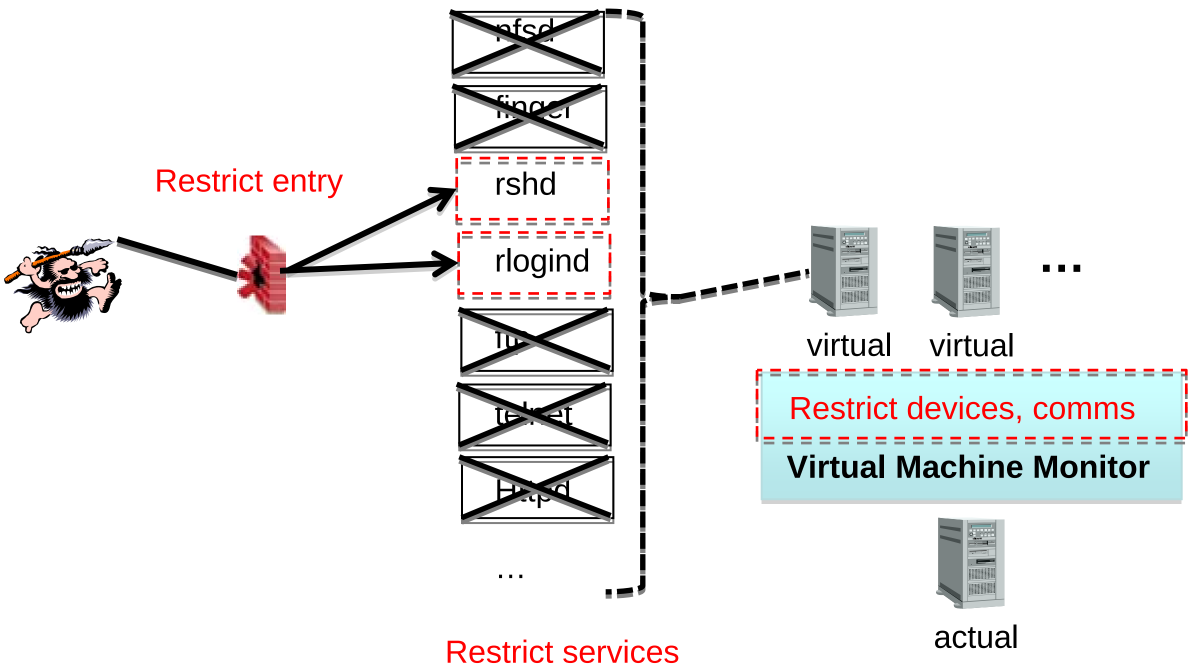
1. Add mediation/observation layers: get restricted topology
2. Analyze source code of new layers to get **constraints**
3. Formalize attacker **goal** as an attack graph
4. Solve **goal** + **constraints**, if possible
5.
   - FALSE means attack not feasible
   - OTHERWISE, get constraints attacker must satisfy.

Augment system to constrain runtime behavior, increase observability

Many hooking techniques are now available:

- System Call Wrappers
- Library Wrappers
- Protocol Wrappers
- Object Wrappers
- Instruction Wrappers
- File System Wrappers
- Device Wrappers
- Translation-based Wrappers

1 I.e., balkanize the system using wrappers, or the sandboxing built into some operating systems



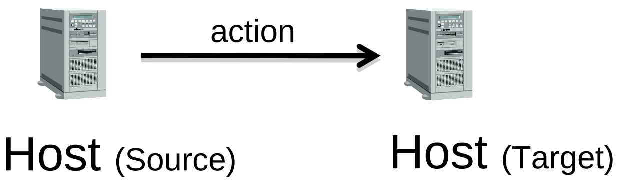
2 Balance with risk of incompatibility

Use Attack Graphs

An **attack graph** is an abstraction of a network (system).

- A node represents network configuration and attacker capabilities held (e.g., root access on host *n*)
- An edge represents an action taken to move to an attacker goal.

action IIS-buffer-overflow **is**
- **intruder preconditions**
  - plvl(S) >= user
  - plvl(T) < root
- **network preconditions**
  - w3svc_T
  - R(S,T,80)
- **intruder effects**
  - plvl(T) := root
- **network effects**
  - ! w3svc_T



1. Nodes **X** services **X** known-vulnerabilities ➔ many possible scenarios

2. Analysis limited to **known** vulnerabilities (e.g., CVE records)

We wish to handle **unknown** vulnerabilities, too…

Credit: from "Tools for Generating and Analyzing Attack Graphs", O. Sheyner and J. Wing, Springer-Verlag 2004.

Traditional Attack Graphs vs Our Approach

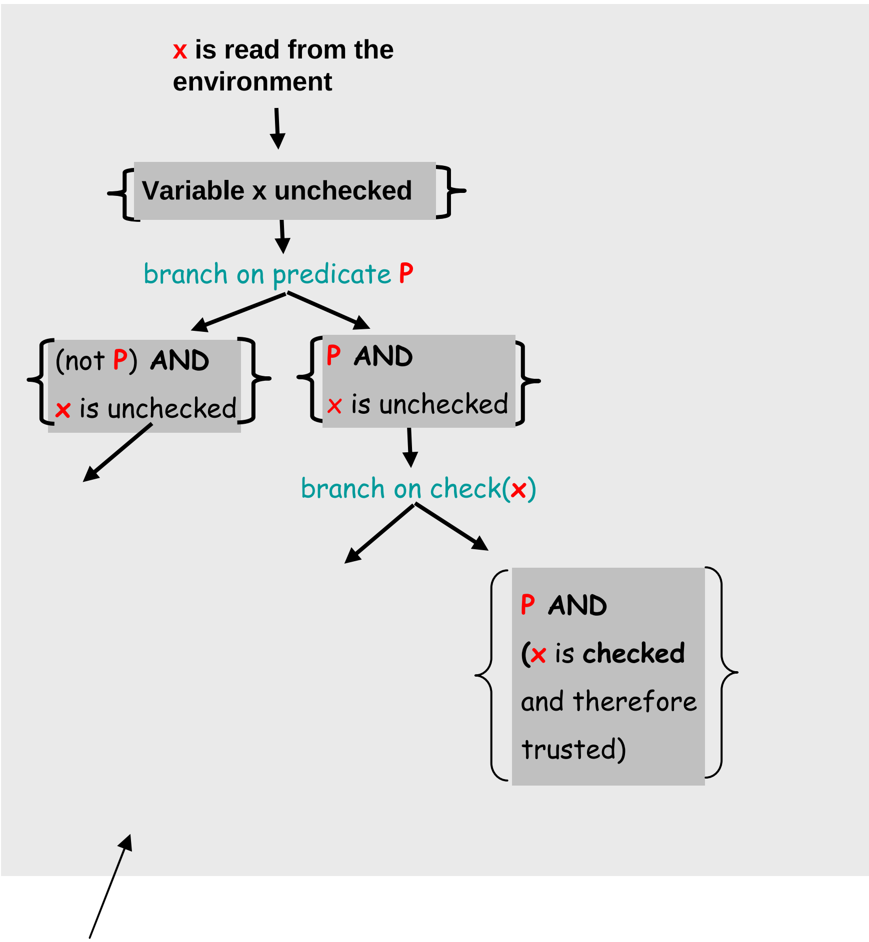
Traditional	Our Approach
Approximate Results For Current (more Complex) Systems	(Hopefully) More Precise Results For (more Restricted) Systems.
State: Connectivity; Host Vulnerabilities; Attacker privileges/goals	State: Connectivity; Source code, for selected services; Attacker privileges/goals
5 hosts X 8 exploits → 5,948 nodes	3 hosts X code X (1 or a few) services
Monotonicity assumption ↓	↓
5 hosts X 8 exploits → 229 nodes	SAT problem: size still unknown

P. Amman, D. Sijesekara, S. Kaushik, "Scalable, Graph-based Network Vulnerability Analysis," CCS'02, Nov., Washington DC.

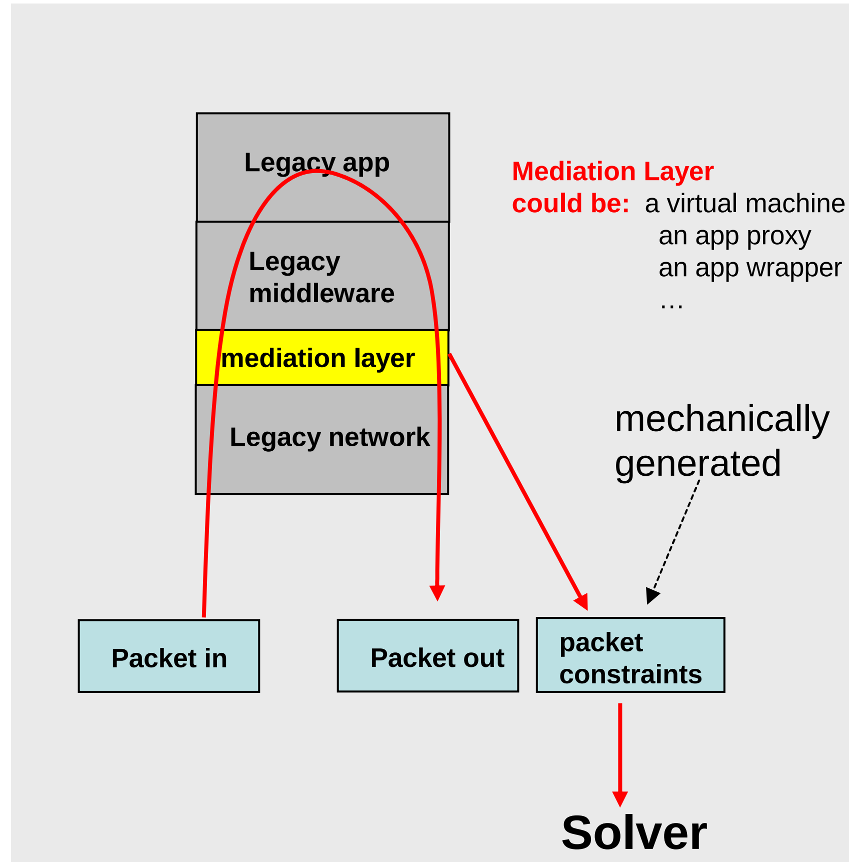
O. Sheyner, J. Haines, S. Jha, R. Lippman, J. Wing, "Automated Generation and Analysis of Attack Graphs," IEEE S&P, Oakland 2002.

Symbolic Execution: Brief Synopsis

Concept



Use

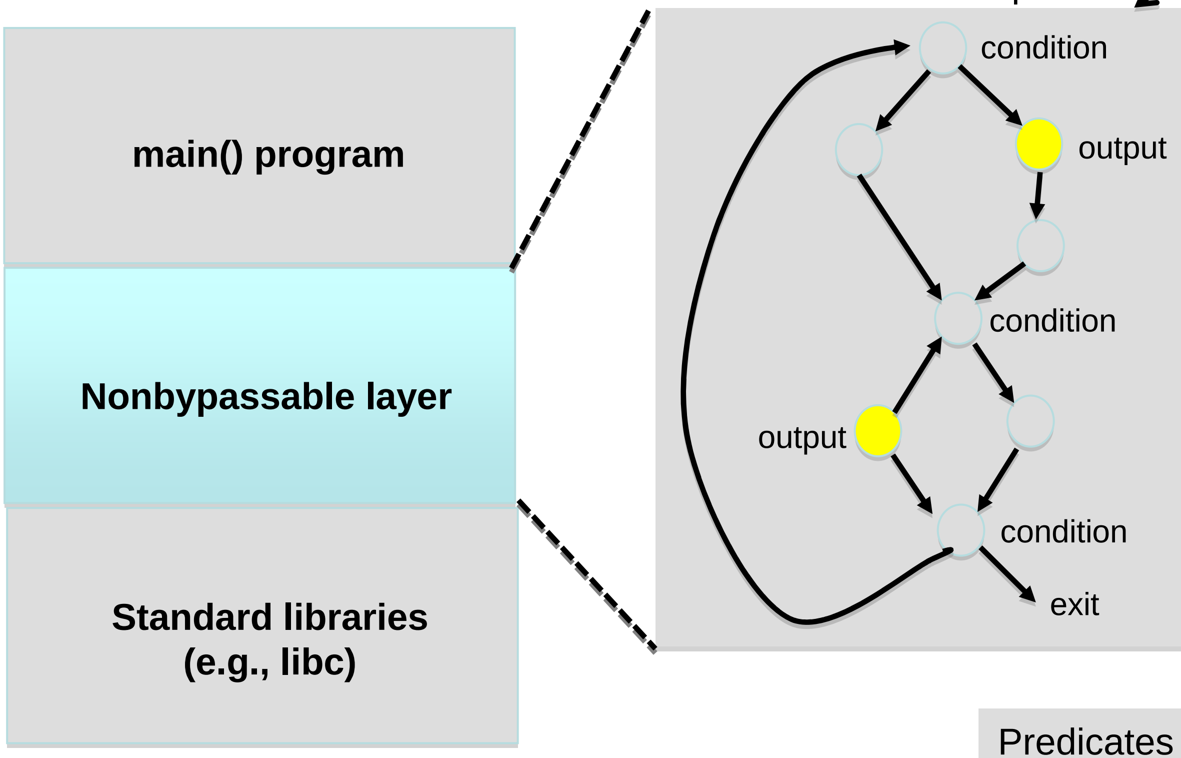


Credit: this legacy idea is in the Stanford Saturn system: see http://saturn.stanford.edu, and others.

Focus Analysis using Slices

M. Weiser, "Program Slicing", IEEE TSE, 1984.

Hammock Graph



Slice layer with respect to selected output statements (e.g., sendmsg())

Instead of generating all statements in the slice, generate boolean expressions at output statements.

Predicates on: values per o_i, ordering, relations on o_i, bindings to external events (e.g., authentication).

Abstract system trace: o_1, o_2, o_3, …..

Specify upstream outputs to be "trustable" by downstream inputs.

Nuts, Bolts, first Steps

Experimenting with the LLVM compiler infrastructure (www.llvm.org).

And with the LLVM-based CLANG (C-family) compiler (clang.llvm.org).

Static Single Assignment gives use/def chains helpful for slicing and symbolic analysis.

Pass management framework makes it pretty easy to develop the analysis as a compiler analysis/transformation pass.

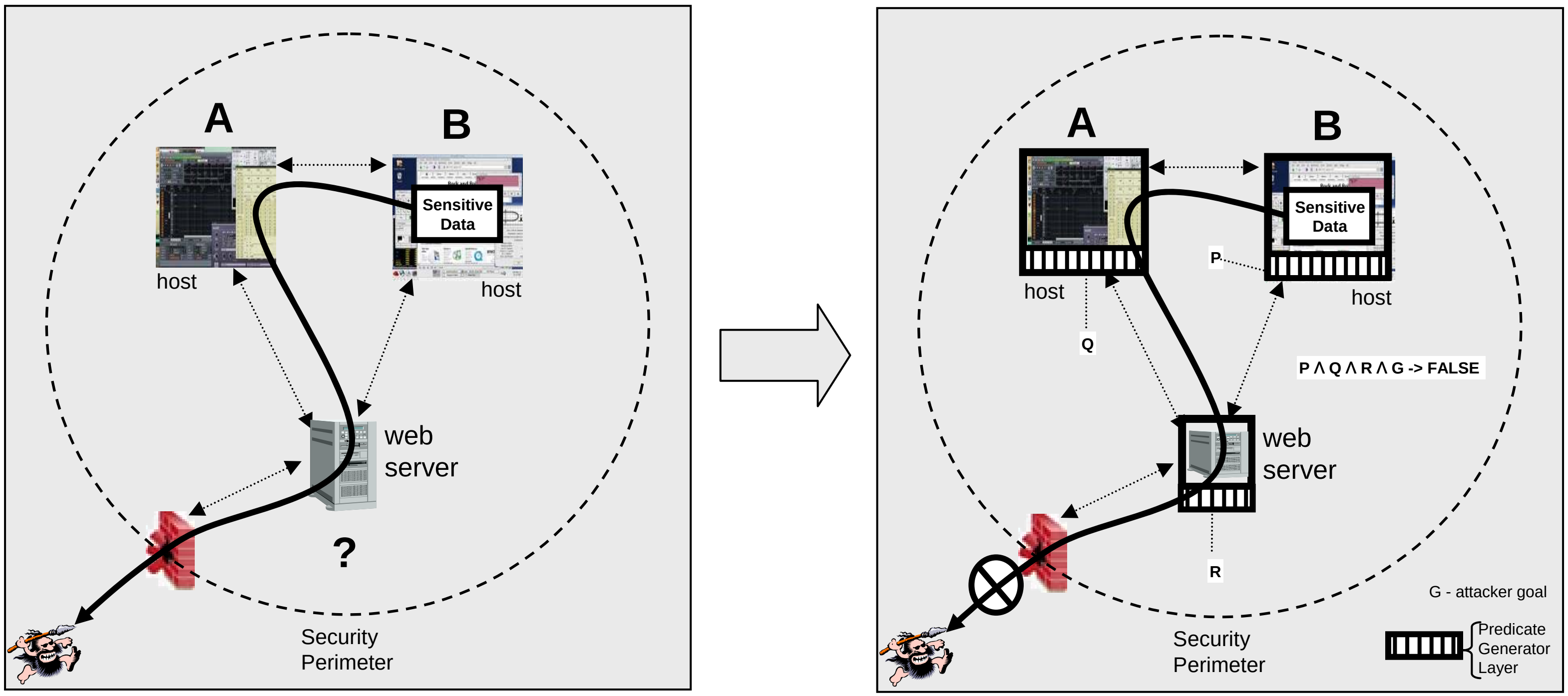
First service chosen: rsh:

Rshd:	393 sloc	"easy"
Libutil:	5,365	Less "easy"
Libpam:	5,383	Less "easy"
Libc:	175,367	Less "easy"

Backup

Informal Example

Before After



Possible Scenario:

1) attacker triggers buffer overflow in IIS, gaining control of IIS
2) captured IIS sends malicious JPEG to host A, gaining control of A
3) host A sends “rcp” command to host B
4) host B “trusts” host A and returns sensitive file
5) host A sends file to the captured IIS
6) captured IIS tunnels file through firewall to attacker

Analysis

1) attacker’s goal is to retrieve the data, i.e., “there exists a sequence of write(src,dest) operations such that write(sensitive-data, d1), write(d1, d2), … write(dn, attacker)” must be satisfiable for the attacker to succeed
2) P is: write(sensitive-data, x) is in the trace only if x is authenticated
3) Q is: if a controlled endpoint reads a complex object, its authentication is subsequently “none”
4) R is: an object passed via HTTP is tagged by its complexity score

Candidate Inputs and Outputs for Measurement

Inputs:

Asset Inventory	List of resources needing protection.
Network Topology	A topological model of the target system showing boundary controllers and where new layers can be transparently inserted to restrict attack paths.
Attacker Victory Conditions	A first-order predicate calculus statement defining attacker victory.
Assumed Attacker Starting Positions	External network access only vs intruder code launched from USB devices vs rogue laptops.

Outputs:

Attacker's Required Constraint Set	Conjunctive normal form boolean expression, Possibly with a proof of unsatisfiability (it's FALSE). Conjunctive normal form constraint set:- it can be Large (e.g., STP has solved a expressions with **2 million** variables for software analysis.
Analysis limitations	Set of simplifying assumptions.

Note: STP is Simple Theorem Prover; see Vijay Ganesh and David Dill, "A Decision Procedure for Bit-Vectors and Arrays"