Non-invasive Smart Grid Threat Analyzer using Formal Methods

Ehab Al-Shaer

Cyber Defense & Network Assurability (CyberDNA) Center

Department of Software and Information Systems

University of North Carolina Charlotte

59th Meeting of the IFIP 10.4 Working Group on Dependable Computing and Fault Tolerance

January 13-17, 2010

Research Background

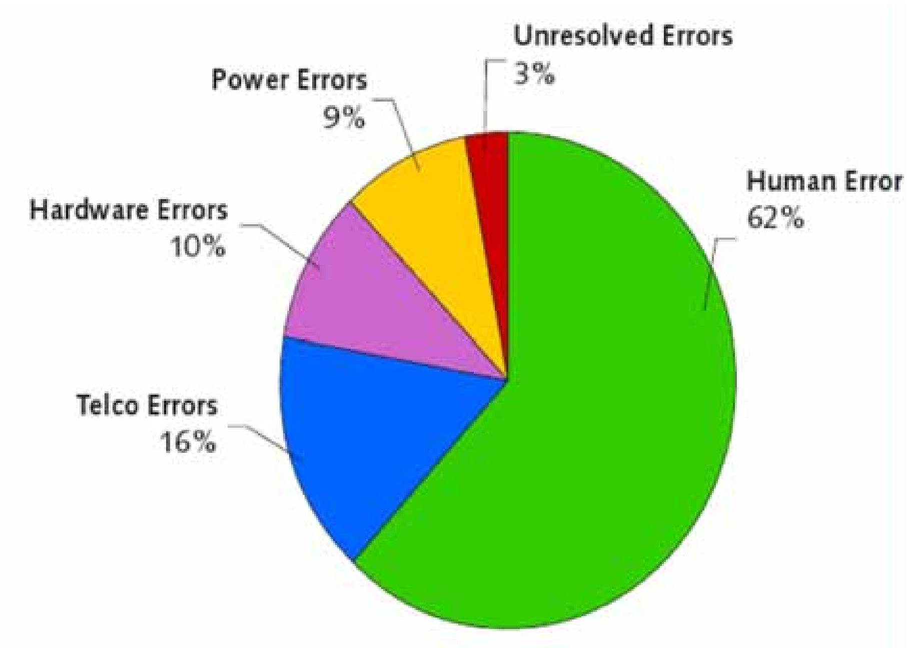
- Research Areas: Using Formal methods
  - Automated Security Configuration Verification, Optimization and Evaluation
  - Proactive Defense (moving target defense)
  - Critical Infrastructure Protection (for Fault & Security) (e.g., Smart Grid, TeleHealth Systems)

- Activities
  - Chair of ACM CCS 2009, 2010
  - Founder and Chair of NSF/ACM SafeConfig, (www.safeconfig.org)
  - NITRD Cyber Security Summit, Aug 2009
  - ARO Moving Target, Oct 2010

Automated Security Configuration– Research Summary

- **Bottom-up approach (compliance, diagnosis and repair)**
  - Firewall Policy Advisor, IM 2003 and INFOCOM04
  - Security Policy Advisor, ICNP2005
    - Conflict Detection (for firewall and IPSec)
      - Intra-firewall analysis
      - inter-device analyses
    - Consistency Checking
  - Proactive Firewall, [INFOCOM 2006, 2007,2009]
  - ConfigChecker, ICNP 2009
  - Community-based Collaborative Diagnosis, DSN 2009
  - SensorChecker (reachability and coverage verification), 2010
  - WikiSeal, 2011
- **Top-Down (Synthesis and Testing)**
  - High-level Firewall Definition Language(FLIP), SACMAT 2007
  - INSPEC Autoamted Firewall Testing, POLICY 2007 and JSAC 2009
  - ConfigBuilder (INFOCOM 2010)
  - ConfigSlider, 2011
  - ConfigLEGO 2011

State of Network Configuration Management



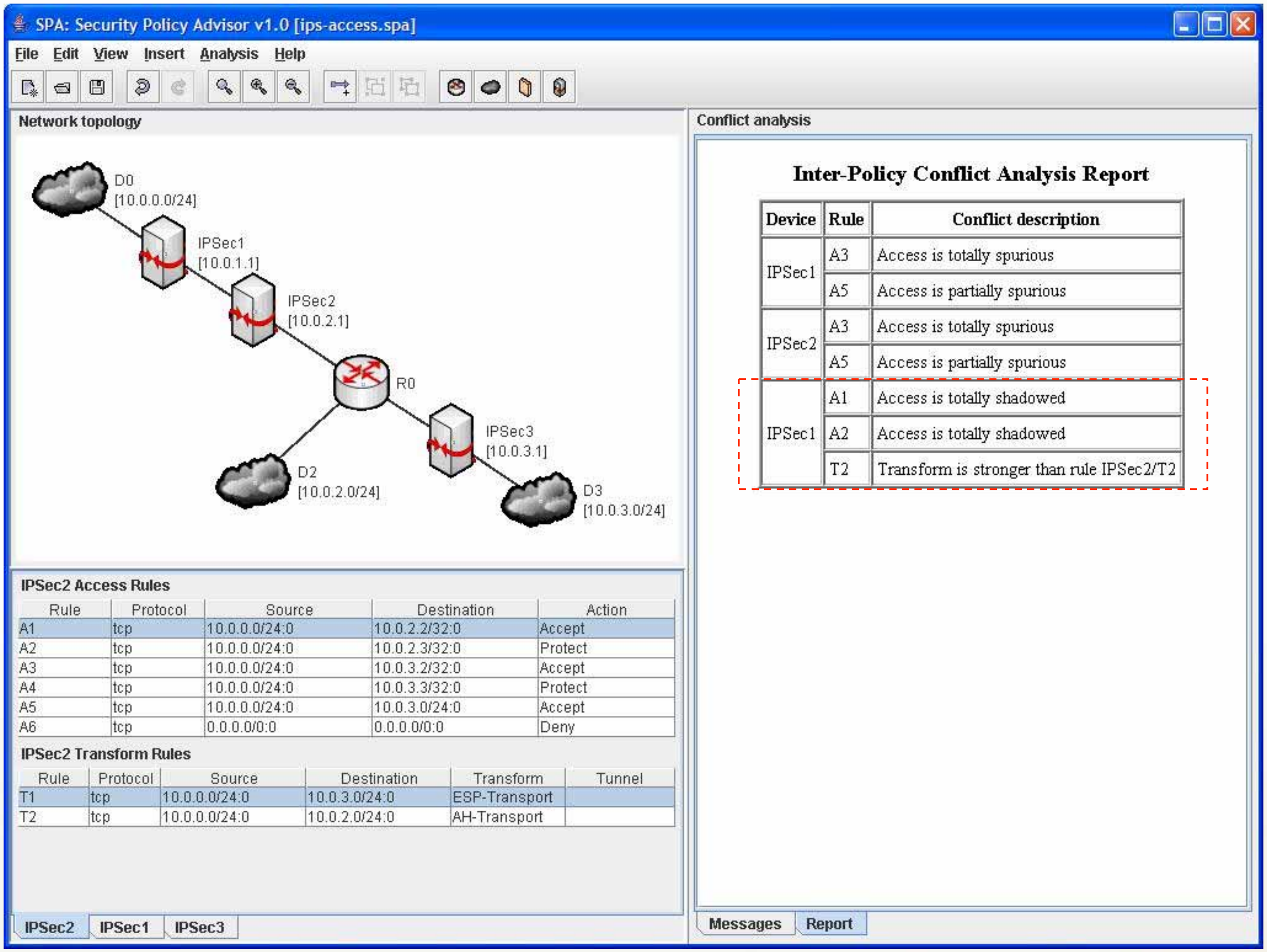
"Eighty percent of IT budgets is used to maintain the status quo.",
Kerravala, Zeus. **"As the Value of Enterprise Networks Escalates, So Does the Need for Configuration Management."** ***The Yankee Group*** **January 2004 [2].**

"Most of network outages are caused by operators errors rather than equipment failure.",
Z. Kerravala. Configuration Management Delivers Business Resiliency. The Yankee Group, November 2002.

- "It is estimated that configuration errors enable 65% of cyber attacks and cause 62% of infrastructure downtime", Network World, July 2006.
- *Recent surveys show Configuration errors are a large portion of operator errors which are in turn the largest contributor to failures and repair time [1].*
- *"Management of **ACLs** was the most critical missing or limited feature, Arbor Networks' Worldwide Infrastructure Security Report, Sept 2007.*

[1] D. Oppenheimer, A. Ganapathi, and D. A. Patterson. Why Internet services fail and what can be done about these? In *USENIX USITS*, Oct. 2003.

Security Policy Advisor



Companies and Institutions Using Security Policy Advisor

UNC CHARLOTTE
College of Computing and Informatics

Companies:

Lisle Technology Partners, USA; Phontech, Norway; Naval Surface Warfare Center, Panama City, USA; Cisco Systems, USA; At&T, USA; Gateshead Council, UK; Danet Group, Germany; TNT Express Worldwide, UK Ltd, United Kingdom; Checkpoint, USA; FireWall-1, The Netherlands; DataConsult, Lebanon; Rosebank Consulting, GB; Mayer Consulting, USA; Panduit Corp, USA; UPMC Paris 5 University, France; Royal institute of Science, Sweden; GE, US; Aligo, USA; Motorola, Inc., USA; Landmark communications, inc., us; uekae.tubitak.gov, Turkey; Duke Energy, USA; The Midland Co, USA; NITW,INDIA; Deloitte & Touche LLP, US; National Taiwan University, Taiwan; Eircom.net. Irland; GE CF, USA; AIT, Thailand; Celestica, Thailand; and Others not listed

Universities/Institutions:

ISRC, Queensland University of Technology, Australia; Imperial College and UCL, London, UK; Columbia University, USA; Georgia Institute of Technology ;NCSU, USA; USC, USA; University of Pittsburgh, PA; University of Waterloo, Canada; University Student in Cyprus International University, Cyprus; University of Rochester, US; UQAM, University of Quebec in Montreal, Canada; Saarland University, Germany; Technical University of Berlin, Computer Science Departement, Germany; UCSB, US; Edith Cowan University, Australia; Universitat Oberta de Catalunya, Spain; ISG, Tunisia; York U, Toronto, Canada; Universidade Federal do Rio Grande do Sul, Brazil; UCL, Belgium; Kent State University, USA; UFRGS, Brazil; University of Stuttgart, IKR, Germany;

Smart Grid vs. Internet Security

- More **Complex** : integration/interdependency of multiple Cyber and Physical networks with different security requirements
  - AMI
  - SCADA
  - Distributed Automation
  - Internet
  - Home
- **More Heterogeneous → potential misconfiguration**
- More potential of **new vulnerabilities/threats**
  - New services
  - cross-network inter-dependency (cyber and physical)
- **More Critical Services** →high threat impact
- **More Closed Network**→ less flexibilities/redundancies

Definitions (based on NIST SP 800-60)

- **Vulnerability** is a *flaw or weakness* in the design or implementation of an information system (including security procedures and security controls associated with the system) that could be intentionally or unintentionally exploited to adversely affect an organization's operations (including missions, functions, and public confidence), assets, or individuals through a *loss of confidentiality, integrity, or availability.*
- **Threat** is any circumstance or event with the potential to intentionally or unintentionally *exploit a specific vulnerability* in an information system resulting in a *loss of confidentiality, integrity, or availability.*
- **Mapping Vulnerability to Threats:**
  - Countermeasure, security configuration, capabilities (e.g., insider), ..etc

AMI Heterogeneous Configuration

- AMI Nodes
  - Smart Meter(SM), Intelligent Collector (IC), Headend system (H), Back-end services, HAN
- AMI Communication Topology
  a. IC from/to Headend (H)
  b. Smart Meter (SM) from/to IC
  c. [SM+IC] to H
  d. Meter to Meter, and IC to IC
- AMI Connectivity/Protocols
  - Unicast (and broacast for unique cases) – no multicast
  - H-IC: Unicast reliable (TCP-based) with congestion control
  - SM-IC: Unicast (LonTalks/LonWorks/NES) reliable but with no congestion control
  - Monitoring and reporting: UDP
- AMI Communication Media:
  - Internet, wifi, cell network, power cable, etc

AMI Heterogeneous Configuration (Cont.)

- AMI Accessibility
  - Authentication
    - Hop-by-hop authentication: SM-IC (LonTalks), IC-H (SSH), cell crypto (UMTS, GPRS), HAN-HS (SSL).
    - IPSec tunnels across public wire/wireless network
  - Access control
    - Between domain boundaries
    - Filters in IC
    - Firewalls in network boundaries
    - Firewall with DMZ for defense in depth in the enterprise network

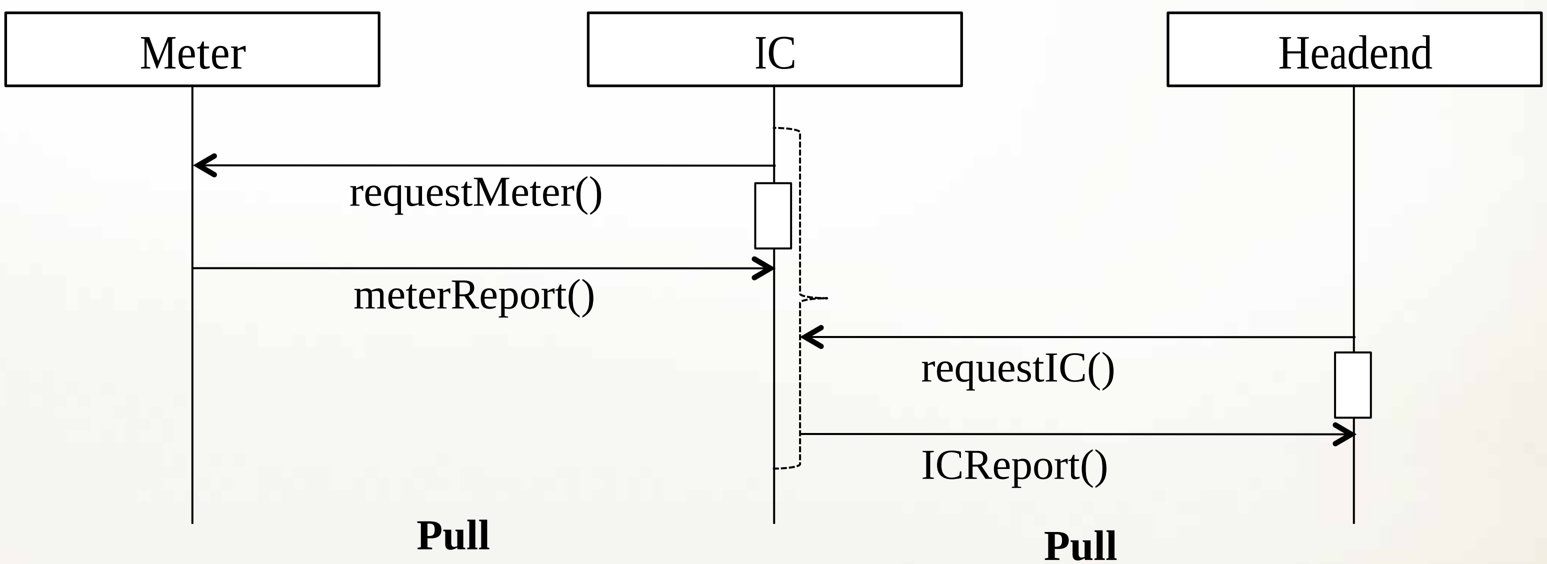
AMI Traffic

- Date (power usage) Reporting -- Outbound
- Alarm Reporting -- Outbound
- Remote Configuration (control command) -- Inbound
- Patching -- Inbound

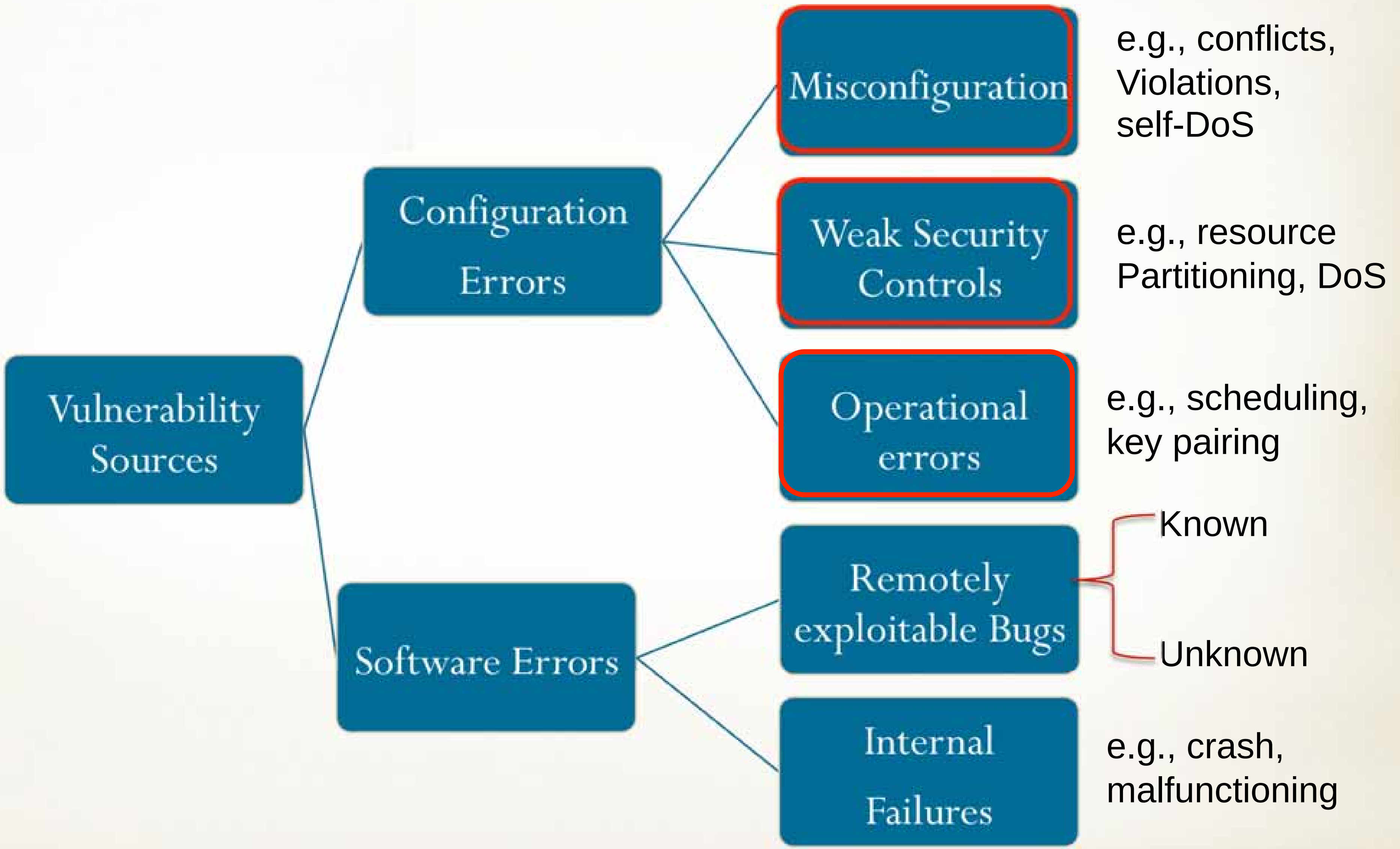
AMI Data Delivery Operation Modes

- Data Reporting/Delivery Mode:
  a. **push** driven (based on schedule)
  b. **pull** driven (based on request)

Category	I	II	III	IV
Between IC & HS	Pull	Push	Pull	Push
Between SM & IC	Pull	Push	Push	Pull



Vulnerability Root Cause Tree for Smart Grid



Potential Threat Impact for Smart Grid

- Impact due to Misconfiguraiton
  - Self-Dos
  - Data loss
  - Alarm loss
  - Unauthorized access
- Impact due to Attakcs
  - DoS
  - Services control hijacking ➔ massive outage
  - *Fault injection ➔ instability*
  - *Privacy issues ➔ low customer incentive*

SG Threat Analyzer Objective

- Threat Analysis
  - Identification,
  - Evaluation
  - mitigation
- End-to-End automated analysis
- Mapping vulnerabilities to threats
  - One vulnerability might cause multiple threats
  - An attack is a combination of specific vulnerability and threat
- Identify attacks surface
- Use non-invasive and off-line analysis
- Scalability to large number of meters and ICs over wide geographical areas

Phase I: Threat Analyzer Tool Capabilities

- Encoding many security controls from NIST and DHS Best Security Practices

- Smart Grid Analysis
  - Reachability analysis
  - Security verification and diagnosis
  - Threat / vulnerability identification

A brief description of Model Properties

Component & Topological Model

1. *Meter-Profile* maintains neutron ID, vendor, MAC-id, list of patches, report data size (traffic rate/time), meter status (active/passive)
2. *IC-Profile* maintains ID, MAC-id, IP address, list of patches, buffer size, IC status (active/passive)
3. *Link-Property* maintains link type (power/ wireless/ ethernet/ fiber/ UMTS/ GPRS etc), bandwidth, delay, encryption type (if any) and security level
4. *Auth-Profile* maintains authentication type (id, protocol), authentication keys associated to a pair of devices.
5. *Crypt-Profile* maintains ID, encryption type (id, protocol), encryption keys associated to a pair of devices.
6. Models routing tables, firewalls, links, paths etc.

AMI Smart Grid Configuration and Operational Analysis

1. **Reachability Analysis Module**
    a. Investigating if a node n_1 is reachable from n_2 across AMI smart grid devices
2. **Data Reporting/Delivery Analysis Module**
    a. Investigating data scope delivered to H at time T based on a given report schedule.
3. **Link and Device Capacity Analysis Module**
    a. Bandwidth availability and link congestions analysis
4. **Vulnerability Analysis Module**
    a. Misconfiguration and hardening: inconsistency, compliance with NISTR, DHS)

Threat Analyzer – Examples of Vulnerability/Threat Analysis

- In general, we will focus on network availability threats, mainly DoS that could be due to one or more of vulnerabilities
  - Lack of separation of duties
  - Lack of resource isolation
  - Lack of monitoring

Threat Analyzer – Examples from AMI Security Profile

- Resource Partitioning and Isolation
  - AMI components must isolate telemetry/data acquisition services from management services
- DoS Protection
  - The AMI system must restrict the ability of internal or external users to launch denial-of-service attacks against other AMI components or networks
  - The AMI system must manage excess capacity, bandwidth, or other redundancy to limit the effects of information flooding types of denial-of-service attacks
  - Wireless assets and networks are also vulnerable to radio-frequency jamming and steps must be taken and personnel trained to address tracking and resolution of such issues.

Threat Analyzer – Examples from AMI Security Profile (cont.)

- Trusted Path:
  - The AMI system must establish trusted communications paths between the user (or agent) and the components making up the AMI system. That is, for every intermediate node in the path, the node is trusted and the communication is protected.
- Access Control:
  - The smart grid system shall employ mechanisms in the design and implementation of AMI to restrict public access to the AMI system from the organization's enterprise network.

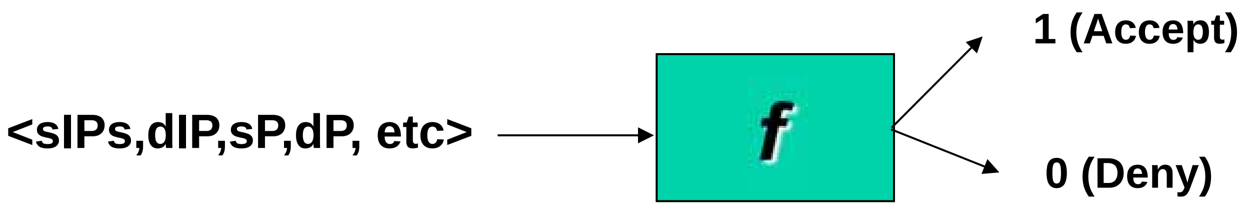
Who Can Use this Tool

- **Accurate, fast and provable analysis**
- **Technical Side**
  - Automated verification, diagnosis and risk analysis
  - Optimal Hardening
  - Capacity planning
  - Anomaly Detection
- **Business Side**
  - Quality assurance
  - Return on investment
  - Technology Planning
  - Others

Configuration Modeling

- **Canonicity:** It can integrates network configurations different syntactically and semantically
- **Composability :** It provide for logical integration of isolated but connected network configuration
- **Reasoning support**
- **Efficient to work with:** scale in term of space and computation complexities

Modeling Access Control Configuration as Boolean Formulas



- Evaluate
- Compare
- Compose

Modeling ACL Configuration Using BDDs

- An ACL policy is a sequence of filtering rules that determine the appropriate action to take for any incoming packets: $P = R1, R2, R3, .., Rn$
- Each rule can be written in the form:

$$R_i := C_i \rightsquigarrow a_i$$

  where C_i is the constraint on the filtering fields that must be satisfied in order to trigger the action a_i

- The condition C_i can be represented as a Boolean expression of the filtering fields $f_1, f_2, \ldots, f_k$ as follows:

$$C_i = fv_1 \wedge fv_2 \wedge \cdots \wedge fv_k$$

where each fv_j expresses a set of matching field values for field f_j in rule R_i. Thus, we can formally describe a ACL policy as:

$$P_a = (C_1 \wedge b_1) \vee (\neg C_1 \wedge C_2 \wedge b_2) \ldots \vee (\neg C_1 \wedge \neg C_2 \ldots \neg C_{i-1} \wedge C_i \wedge b_i)$$

rule1 rule2 rule_n

$$\text{where } b_i = \begin{cases} 1 \text{ if } action_i = a \\ 0 \text{ if } action_i \neq a \end{cases}$$

Concise Formalization

- Single-trigger policy **is an access policy where only one action is triggered for a given packet.** C_i is the 1st match leads to action *a*

$$P_a = \bigvee_{i \in index(a)} (\neg C_1 \wedge \neg C_2 \ldots \neg C_{i-1} \wedge C_i)$$

$$P_a = \bigvee_{i \in index(a)} \bigwedge_{j=1}^{i-1} \neg C_j \wedge C_i$$

- Multiple-trigger policy **is an access policy where multiple different actions may be triggered for the same packet.** C_i is any match leads to action *a*

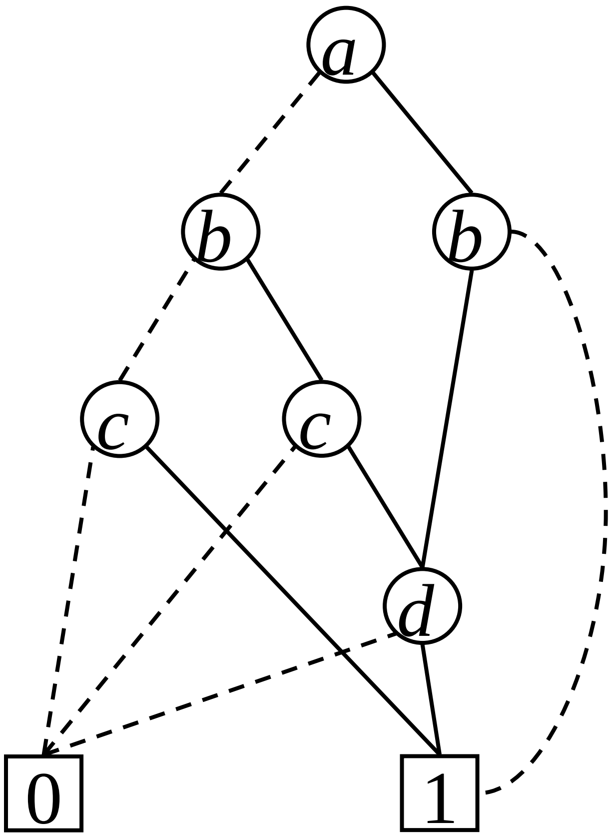
$$P_a = \bigvee_{i \in index(a)} C_i$$

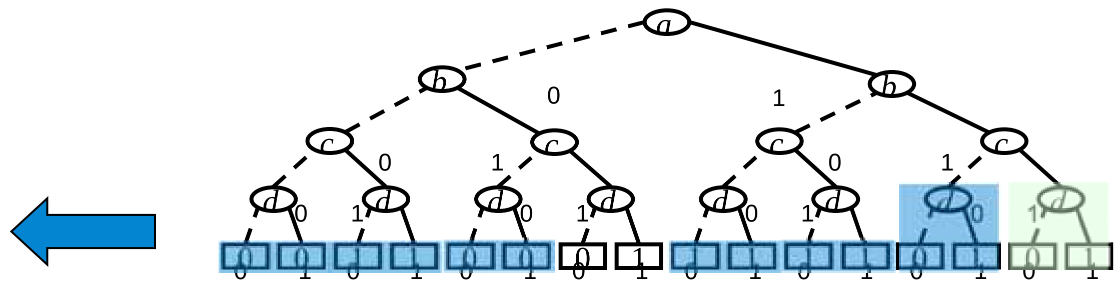
where

$$index(a) = \{i \mid R_i = C_i \rightsquigarrow a\}$$

Ordered Binary Decision Diagram

$$(a \vee c) \wedge (b \rightarrow d)$$





Properties of BDD

Storage Efficiency (often compact)

Many common Boolean functions have small OBDD representations.

Canonicity

If the order in which the variables are tested is fixed, then there exists only one OBDD for each Boolean formula.

- **Lemma 1:** (Canonicity lemma)

For every function $f: Bn \rightarrow B$, there is **exactly one** ROBDD u with variable ordering x1<x2<…<xn such that $fu = f$(x1 , x2, …, xn)

Efficient operations

data structure for propositional logic formulas

- BDD operations: Build, Apply, Restrict, Existential quantification. SATCount, anySAT, allSAT

BDD Applications in Network Security Configuration Analysis

Applications

(1) Conflict Detection

(2) Configuration Hardening

Intra-Policy Conflicts Formalization : C… access List

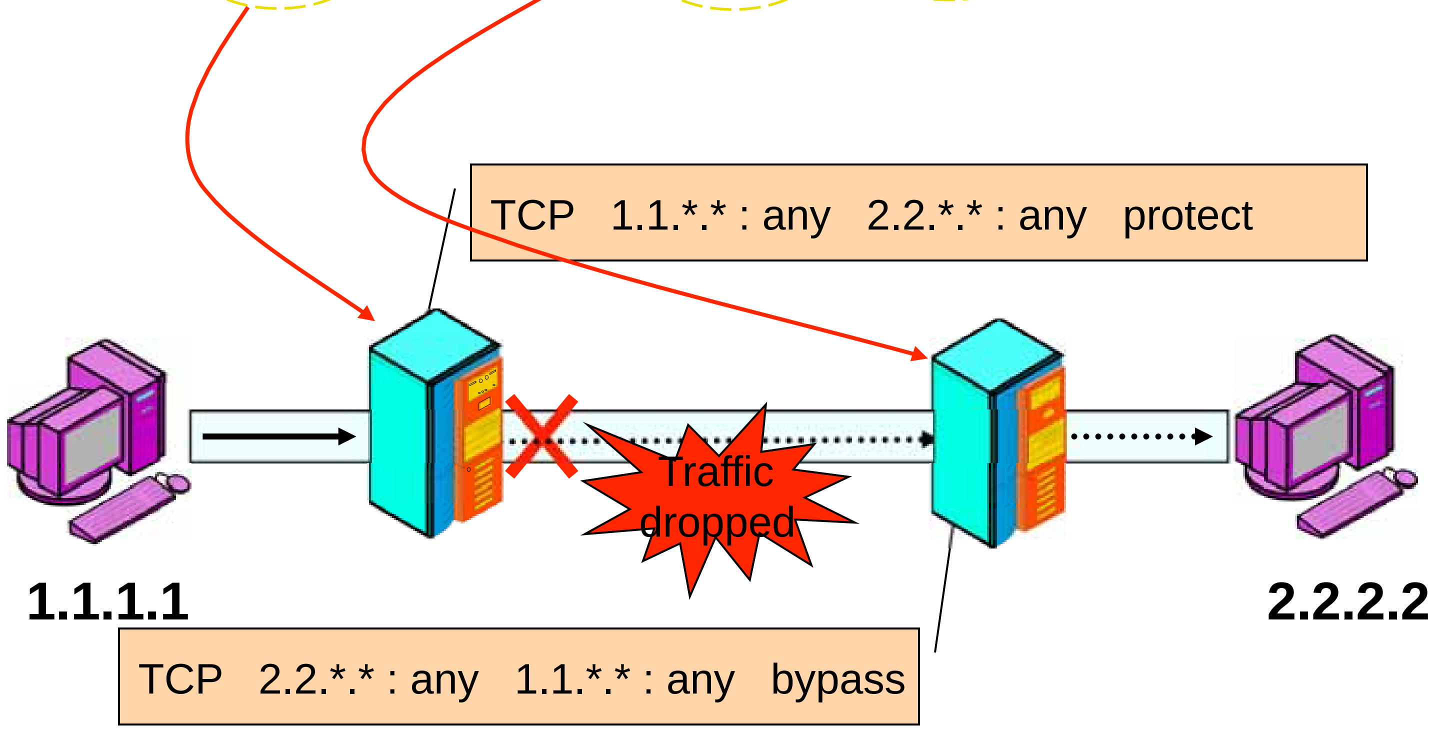
Soundness & Completenes
s

- **Policy expression S_a represents a policy that incorporates rule R_i, and S'_a is the policy with R_i excluded. R_i may be involved in the following conflicts:**
  - **Shadowing:**
    $[(S'_{a_i} \Leftrightarrow S_{a_i}) = true]$ and $[(C_i \Rightarrow S'_{a_i}) = false]$
  - **Redundancy:**
    $[(S'_{a_i} \Leftrightarrow S_{a_i}) = true]$ and $[(C_i \Rightarrow S'_{a_i}) \neq false]$
  - **Exception:**
    $[(S'_{a_i} \Leftrightarrow S_{a_i}) \neq true]$ and $[(C_i \Rightarrow S'_{a_i}) = false]$
  - **Correlation:**
    $[(S'_{a_i} \Leftrightarrow S_{a_i}) \neq true]$ and $[(C_i \Rightarrow S'_{a_i}) \neq false]$

IPSec Inter-Policy Conflicts Formalization: Crypto-access Lists

- **Shadowing**: upstream policy blocks traffic

$$[(S^u_{discard} \wedge \neg S^d_{discard}) \vee (S^u_{protect} \wedge \neg S^d_{protect})] \neq false$$



Composable Security Configuration Verification & Analysis

Themes:

- Security Configuration Hardening
- Integrating other device and host configuration
- Property based verification

Modeling Routing Access Control

- We can define the routing policies as follows: let a routing rule be encoded as $R_i := D_i \leadsto n$
  - Where *n* is integer representing the forwarding port ID

where D_i is the destination and n_i is a unique integer (id) designating the next hope in the network. Thus, the policy of the routing entries (ordered based on longest-common prefix) that forward to next hope n_k can be defined as follows:

$$T_n = \bigvee_{i \in index(n)} \bigwedge_{j=1}^{i-1} \neg D_j \wedge D_i \;\; s.t. \;\; index(n) = \{i \mid R_i = D_i \leadsto n\}$$

- We can then represent the entire routing table for a node *j* as follows:

$$T^j = \bigvee_{\forall n \, = \, next \;\; hope} T_n$$

.

Modeling Routing Access Control (2)

- We can define the routing policies as follows: let a routing rule be encoded as $R_i := D_i \rightsquigarrow n$
  - where D_i is the destination and n is a unique integer (id) designating the forwarding port (or next hope in the network).
- Thus, the model of an entire routing policy for node *j is* defined as follows:

$$T^j = \bigvee_{i \in index(n)} \bigwedge_{j=1}^{i-1} \neg D_j \wedge D_i \wedge n \quad s.t. \;\; index(n) = \{i \mid R_i = D_i \rightsquigarrow n\}$$

- To get the routing entries for a specific port, say x, we can do the following: $T^j \mid n{=}x$ or T^j_n

Composability: Path Conflict Analysis for Firewalls

- ***Lemma:*** **If S_A^u, S_A^d are the upstream and downstream firewalls in a path, then**
  (a) S^u causes inter-policy shadowing with S^d *iff* $[(\neg S_A^u \wedge S_A^d) \neq false]$
  (b) S^u causes inter-policy spuriousness with S^d *iff* $[(S_A^u \wedge \neg S_A^d) \neq false]$
- ***Lemma:*** **Shadow-free and spurious-free are *transitive* relations. Thus, assume S_A^i, S_A^j and S_A^k are upstream to downstream firewall polices in a path a, the following relation is always true (shadowing-free case) :**

$$[(\neg S_A^i \wedge S_A^j) = false] \bigwedge [(\neg S_A^j \wedge S_A^k) = false] \Rightarrow [(\neg S_A^i \wedge S_A^k) = false]$$

- Path Conflict: **Assuming S_A^1 to S_A^n are the firewall policies from upstream to downstream in the path from *x* to *y*, a *path conflict (x,y)* between any two firewalls from *i* to *n* path is defined as follows:**

  (a) Path-Shadowing (x,y):

$$[\bigvee_{i=1,n-1 \ and \ i \in path(x,y)} \neg S_A^i \wedge S_A^{i+1} \neq false]$$

  (b) Path-Spuriousness (x,y):

$$[\bigvee_{i=1,n-1 \ and \ i \in path(x,y)} S_A^i \wedge \neg S_A^{i+1} \neq false]$$

Diagnosing Unreachablility Problems between Routers and Firewalls

- **Flow-level Analysis: Is the flow C_k that is** forwarded **by routers in path *P* (each routing tables is represented as BDD T^i_j for router *i* and port j) but** blocked **due to conflict between *Routing* and *FW Filtering*:**

$$[(C_k \Rightarrow \bigwedge_{(i,j)\in P} T^i_j) \wedge (C_k \Rightarrow \neg S^n_A)] \neq false$$

  - This shows that a traffic C_j is forwarded by the routing policy, T^i_j, from node *i* to *n* but yet blocked by the filtering policy, $S^n_{discard}$, of the destination domain.
- **Path-level Analysis: What are all** unreachability Conflicts **between *Routing* and *Filtering*:**

$$\phi_k \leftarrow [SAT^*(\bigwedge_{(i,j)\in path(P)} T^i_j \wedge \neg S^n_A \wedge \neg(\bigwedge_{i=1,k-1} \phi_i))] \neq false$$

  - For phi=1, n misconfiguration examples, and phi(0) = ture
- **Network or Federated-level Analysis: Spurious conflict between downstream *d* and upstream *u* ISP domains:**

$$[(S^u_{bypass} \wedge \neg S^d_{bypass}) \vee (S^u_{limit} \wedge S^d_{discard})] \neq false$$

  - Notice that $S_{discard}$, S_{bypass} and S_{limit} are filtering policies representations related to the filtering actions as described in [POLICY08, ICNP05, CommMag06].

*: AnySAT

Automated Security Configuration Verification – ConfigChecker

- Global analysis of network behaviors using device configuration and policies
  - routing, firewalls, NAT, IPSec/VPN, multicast, proxy server etc.
- Uses BDD/SAT and Model Checker: track the packet state transformation
- Applications
  - Basic reachability and security requirements verification
  - Analysis that requires history/state exploration like
    - Route cycles
    - Hidden tunnels
    - Packet transformation (IPSec or proxies)
  - Measure “network resistance” or attack surface
- Scales to **1000s** of devices and **millions** of rules

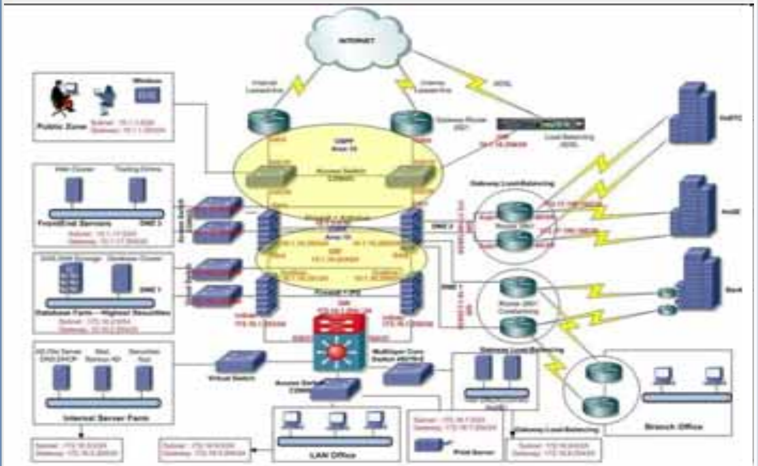
UNC CHARLOTTE
College of Computing and Informatics

ConfigChecker Interface Design

SPA: Security Policy Advisor v1.0 [ips-access.spa]

File Edit View Insert Analysis Help

Network topology



IPSec2 Access Rules

Rule	Protocol	Source	Destination	Action
A1	tcp	10.0.0.0/24:0	10.0.2.2/32:0	Accept
A2	tcp	10.0.0.0/24:0	10.0.2.3/32:0	Protect
A3	tcp	10.0.0.0/24:0	10.0.3.2/32:0	Accept
A4	tcp	10.0.0.0/24:0	10.0.3.3/32:0	Protect
A5	tcp	10.0.0.0/24:0	10.0.3.0/24:0	Accept
A6	tcp	0.0.0.0/0:0	0.0.0.0/0:0	Deny

IPSec2 Transform Rules

Rule	Protocol	Source	Destination	Transform	Tunnel
T1	tcp	10.0.0.0/24:0	10.0.3.0/24:0	ESP-Transport	
T2	tcp	10.0.0.0/24:0	10.0.2.0/24:0	AH-Transport	

IPSec2 | IPSec1 | IPSec3

Conflict analysis

Reachability Analysis			
Device	**Rule**	**Violation Type**	**Counter Example**
R1	33	Unreachable	* → 150.15.3.*
FW23	90	Unreachable	Internet → officePC/80
R16	40	Loop	(n1, n4, n5, n6, n1)
Security Analysis			
Device	**Rule**	**Violation Type**	**Counter Example**
R19	224	Backdoor	(N1, R1, R19, DB_Srv)
GW34	101	IPSec Violation	3DES instead of AES
GW39-41	34	Broken IPSec Tunnel	(N9/any, N90/555, ESP)
WWW	402	Unauthorized Access	User(External) → Put(file,NFS)
GW55	90	Integrity Violation	Can Decrypt(GW55) → T
FW3	none	Backdoor	(N1, R20, FW3, DB_Srv)
GW1	15	IPSec Violation	3DES instead of AES
FW10	None	Attack Surface	Compromised(WWW) → Compromised(SQLServer)
Consistency Analysis			
Device	**Rule**	**Violation Type**	**Description**
FW1	65	Shadowing	
FW9	20	Spuriousness	
IPSec10-12	10-13	Mismatch	
Reliability Analysis			
Device	**Rule**	**Violation Type**	**Counter Example**
R15	None	Recovery	Faulty (R14, FW1) → Unreachable (DNS)
R15	None	Confidentiality	Faulty (R14, IPSec3) → Unprotected (N3, N5)

Messages | Report

Formalization – The Basic Model

- The network is modeled as a state machine
  - each state determined by the packet header information and packet location on the network:

    States = Locations X Packets

  - The ***characterization function*** to encode the state of the network in the basic model (abstracting payload)

$$\sigma : IP_s \times port_s \times IP_d \times port_d \times loc \rightarrow \{\textbf{true}, \textbf{false}\}$$

IP_s	the 32-bit source IP address
$port_s$	the 16-bit source port number
IP_d	the 32-bit destination IP address
$port_d$	the 16-bit destination port number
loc	the 32-bit IP address of the device currently processing the packet

Formalization – The Basic Model

- Network devices are modeled based on the **packet *matching semantic*** and ***packet transformation***
  - Each rule consists of a condition (Ci) and an action (a): Ci➔a
  - Policy are set of rules matched sequentially with single- or multi-trigger actions
  - Firewall (single trigger) policy encoding using BDD

$$P_a = \bigvee_{i \in index(a)} (\neg C_1 \wedge \neg C_2 \ldots \neg C_{i-1} \wedge C_i)$$
$$= \bigvee_{i \in index(a)} \bigwedge_{j=1}^{i-1} \neg C_j \wedge C_i$$

- **Transformation:**
  - if a pkt *state* matches the rule *condition*, the Action can change the packet <u>location</u> and possibly the <u>headers</u> ➔ means change over the bits of the *state*
- **Transition relation** is *characterization function* as follows:
  - **t**: (Curr_pkt x Curr_loc)x (New_pkt x New_loc) → {true, false}
  - Device Model ϕ = ***loc*** ∧ **M*atch_Condition*** ∧ t → **{true, false}**

Formalization – The Basic Model

- Global Transitions relation of the entire network:

$$T = \bigvee_{i \in devices} \Phi_{device_i}$$

- Variables
  - Locations is every place that can describe packet position: firewall, router, IPSec device, or application layer service, etc.
  - We allow Location to be different than IPsrc for spoofing
  - There are two versions of each variable: current and new state.
- Each property and field describing the state (i.e., location IP; packet properties: src/dst IP; port, proto, transformation, etc) is represented by bits, according to its size.
- These variables are used via a symbolic representation using Ordered Binary Decision Diagrams.
- Model Checking and CTL are used to answer the queries posed by the administrator.

Formalization: The Basic Model

- Firewall Modeling (Example)

FW-IP=1, next-hop-IP=3
IPsrc=2, IPdest=* ➔ allow
IPsrc=*, IPdest=3, Pdest=1 ➔ allow

$$(s_1 \wedge \overline{s_0}) \vee (d_1 \wedge d_0 \wedge d_p) \wedge \bigwedge_{i\in\{0,1,p\}} (s'_i \Leftrightarrow s_i) \wedge \bigwedge_{i\in\{0,1,p\}} (d'_i \Leftrightarrow d_i) \wedge l'_1 \wedge l'_0 \wedge \overline{l_1} \wedge l_0$$

- Router Modeling (Example)

Router-IP=2
IPdest=0 ➔ nexthop=0
IPdest=1 ➔ nexthop=0
(default-gateway) ➔ nexthop = 3

$$(\overline{d}_1 \wedge \overline{l'_1} \wedge \overline{l'_0}) \vee (d_1 \wedge l'_1 \wedge l'_0) \wedge \bigwedge_{i\in\{0,1,p\}} (s'_i \Leftrightarrow s_i) \wedge \bigwedge_{i\in\{0,1,p\}} (d'_i \Leftrightarrow d_i) \wedge l_1 \wedge \overline{l_0}$$

- NAT Modeling (Example)

IP(NAT)= 2 connected to IP= 1
IPsrc=3/sport=1, IPdes=1➔
IPsrc=2/sport=0, IPdes=1

outgoing $$[s_1 \wedge s_0 \wedge s_p \wedge \overline{l'_1} \wedge l'_0 \wedge s'_1 \wedge \overline{s'_0} \wedge \overline{s'_p} \wedge \bigwedge_{i\in\{0,1,p\}} (d'_i \Leftrightarrow d_i)] \vee$$

incoming $$[d_1 \wedge \overline{d_0} \wedge \overline{d_p} \wedge l'_1 \wedge l'_0 \wedge d'_1 \wedge d'_0 \wedge d'_p \wedge \bigwedge_{i\in\{0,1,p\}} (s'_i \Leftrightarrow s_i)] \wedge l_1 \wedge \overline{l_0}$$

Formalization – The Extended Model

- IPSec encapsulation requires new headers and saving the old headers ➔ copier, stack, valid bit
- IPSec Modeling
  - Example: IPsrc=0, IPdest=3 ➔ enc_tunnel

(from Gateway of IP=1 to Gateway of IP=2)

➡ $l_0 \wedge \overline{l_1}$ — Current location

➡ $\wedge \overline{s_0} \wedge \overline{s_1} \wedge d_0 \wedge d_1$ — Matching Condition

➡ $\wedge \overline{v} \wedge v'$

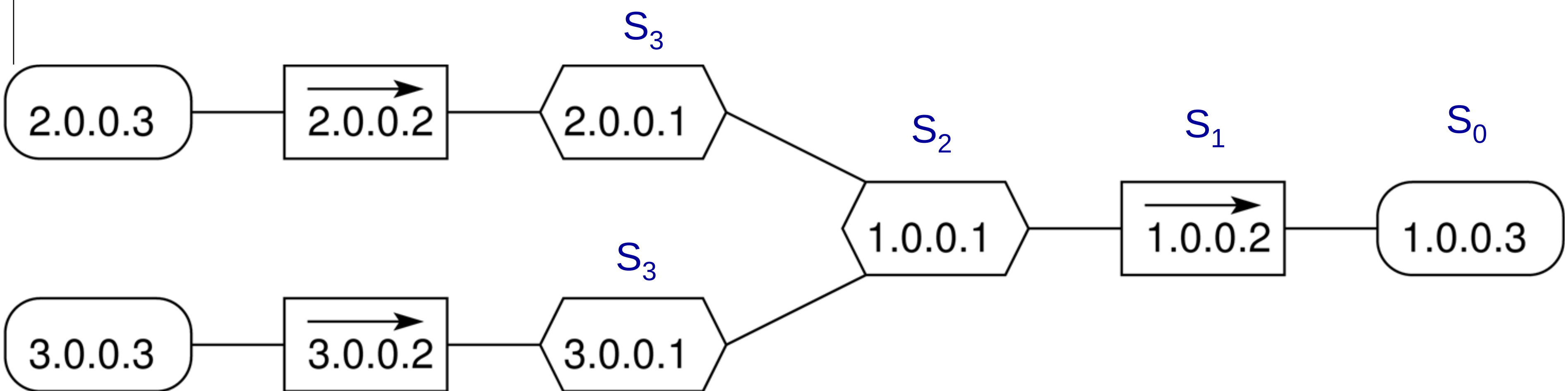
➡ $\wedge \bigwedge_{i \in \{0,1,p\}} (\widehat{s'_i \Leftrightarrow s_i}) \wedge \bigwedge_{i \in \{0,1,p\}} (\widehat{d'_i \Leftrightarrow d_i})$ — Copying headers

➡ $\wedge \overline{s'_1} \wedge s'_0 \wedge s'_p \leftrightarrow s_p \wedge \overline{d'_0} \wedge d'_1 \wedge d'_p \leftrightarrow d_p$ — New headers

$\wedge \overline{l'_0} \wedge l'_1$ — New location

Example

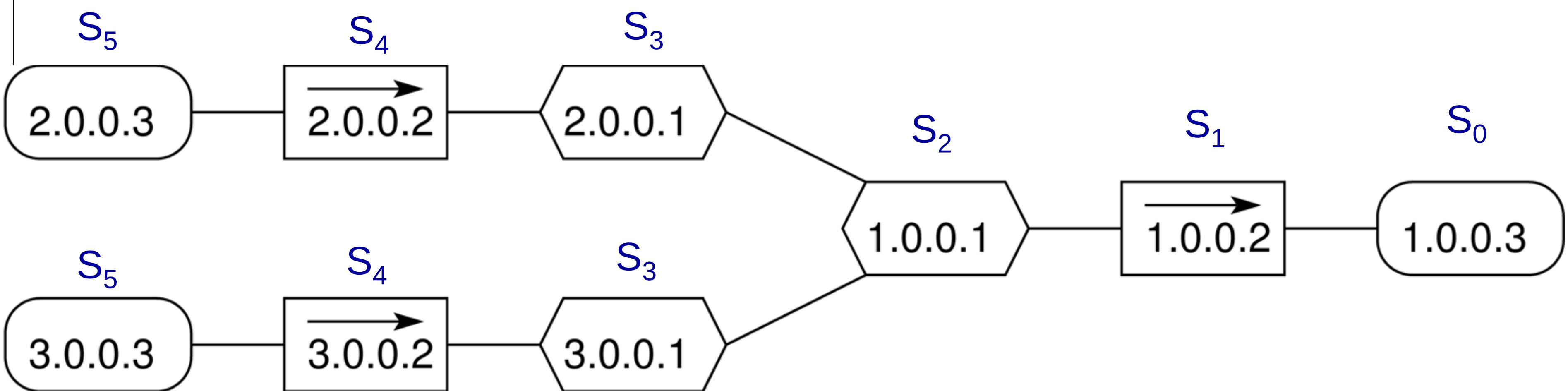
EF(loc=1.0.0.3)



loc	src	dst	loc’	src’	dst’
2.0.0.3	*	*	2.0.0.2	src	dst
2.0.0.2	*	*	2.0.0.1	src	dst
2.0.0.1	*	1.*.*.*	1.0.0.1	src	dst
2.0.0.1	*	3.*.*.*	1.0.0.1	src	dst
3.0.0.3	*	*	3.0.0.2	src	dst
3.0.0.2	*	*	3.0.0.1	src	dst
3.0.0.1	*	1.*.*.*	1.0.0.1	src	dst
3.0.0.1	*	2.*.*.*	1.0.0.1	src	dst
1.0.0.1	*	1.*.*.*	1.0.0.2	src	dst
1.0.0.1	*	2.*.*.*	2.0.0.1	src	dst
1.0.0.1	*	3.*.*.*	3.0.0.1	src	dst
1.0.0.2	2.*.*.*	1.0.0.3	1.0.0.3	src	dst

Example

EF(loc=1.0.0.3)



loc	src	dst	loc’	src’	dst’
2.0.0.3	*	*	2.0.0.2	src	dst
2.0.0.2	*	*	2.0.0.1	src	dst
2.0.0.1	*	1.*.*.*	1.0.0.1	src	dst
2.0.0.1	*	3.*.*.*	1.0.0.1	src	dst
3.0.0.3	*	*	3.0.0.2	src	dst
3.0.0.2	*	*	3.0.0.1	src	dst
3.0.0.1	*	1.*.*.*	1.0.0.1	src	dst
3.0.0.1	*	2.*.*.*	1.0.0.1	src	dst
1.0.0.1	*	1.*.*.*	1.0.0.2	src	dst
1.0.0.1	*	2.*.*.*	2.0.0.1	src	dst
1.0.0.1	*	3.*.*.*	3.0.0.1	src	dst
1.0.0.2	2.*.*.*	1.0.0.3	1.0.0.3	src	dst

S_1 =SAT(T(current_state and Next_state=S_0))

S_2 =SAT(T(current_state and Next_state=S_1))

S_3 =SAT(T(current_state and Next_state=S_2))

= (Loc=**2.0.0.1** ^ src=2.*.*.*. ^ dst=1.0.0.3) v

(Loc=**3.0.0.1** ^ src=2.*.*.*. ^ dst=1.0.0.3)

And so on

Thus the answer will be a set of all states=

(S1 v S2 v S3 v S4 v S5)

ConfigChecker Box-- Querying the Network

- After loading the configuration files and digesting them into the unified model, CTL- (or LTL) based queries can be issued
- Configuration soundness and completeness (e.g., routing, VPN)
- Any general property-based verification
- Satisfying assignments to the CTL-based queries, are the answer to our queries.

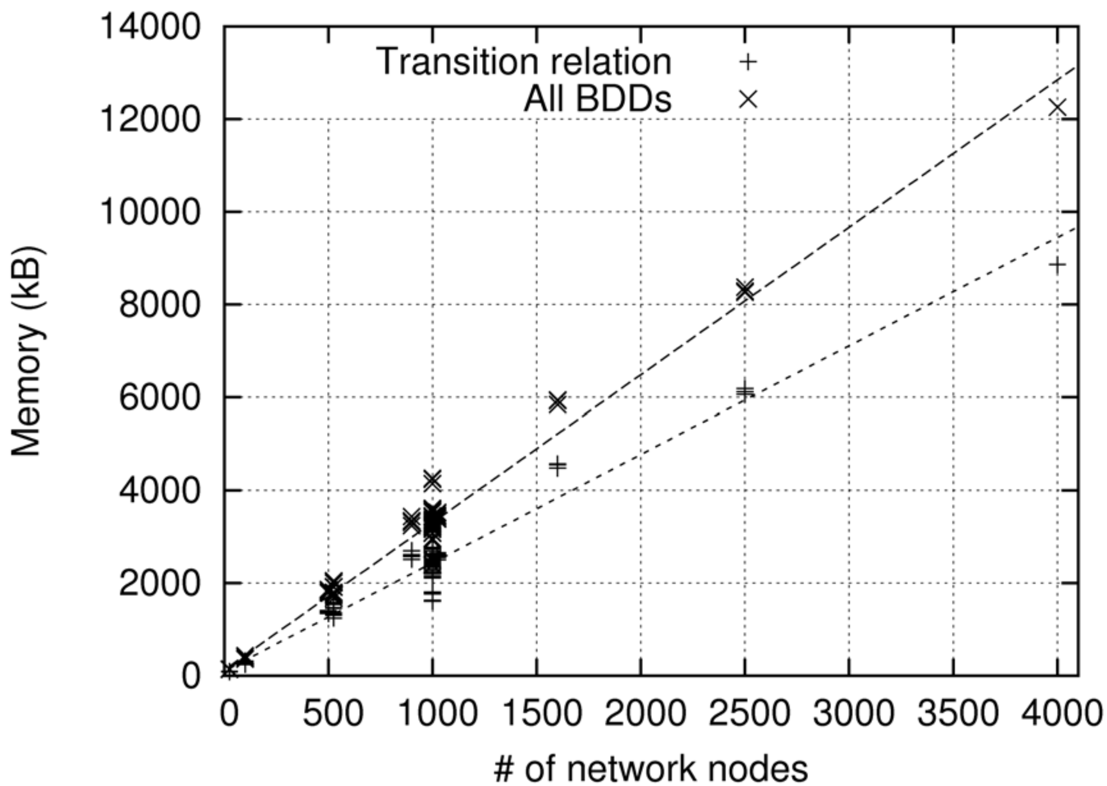
Examples of Configuration Analysis using ConfigChecker Query Interface

	Basic reachability
Q1:	$(src = a1 \wedge dest = a2 \wedge loc(a1)) \rightarrow \mathbf{AF}(src = a1 \wedge dest = a2 \wedge loc(a2))$
	Given a starting location and a flow, d 's packets of this flow eventually reach the destination?
	Reachability Soundness
Q2:	$[loc(a1) \wedge src(a1) \wedge dst(a2) \wedge \mathbf{EF}(loc(a2))] \rightarrow \mathcal{P}connect(a1, a2)$
	If the src can reach the destination in configuration then it must be allowed in CRP.
	Reachability Completeness
Q3:	$\mathcal{P}connect(a1, a2) \Rightarrow [loc(a1) \wedge src(a1) \wedge dst(a2) \rightarrow \mathbf{EF}(loc(a2))]$
	if CRP allows $a1$ to reach $a2$, then there must a path in the configuration that eventually allows $a1$ to reach $a2$.
	Discovering routing loops
Q4:	$loc(a1) \wedge \mathbf{EX}(\mathbf{EF}(loc(a1)))$
	Is there a node that can reach $a1$ and for the same flow it is the next hop of $a1$?
	Shadow or Bogus routing entries
Q5:	$\mathbf{EX}(true) \wedge \neg\mathbf{EX}\_(true) \wedge (loc(router1) \vee loc(router2) \ldots)$
	Given all routers, does any have a decision for traffic will never reach it from its previous hop?
	End-to-end integrity of single/nested or cascaded IPSec encrypted tunnel
Q6:	$(src = a1 \wedge dest = a2 \wedge loc(a1) \wedge IPSec(encT)) \rightarrow \mathbf{AU}((IPSec(encT) \vee loc \rightarrow \mathcal{G}), loc(a2))$
	If the traffic is encrypted in a tunnel from the src then it will appear decrypted only at the destination or at intermediate authorized gateways ($\mathcal{G}$) that allow for cascaded tunnels. If $\mathcal{G}=false$, then there are no intermediate gateways and the traffic must travel through a single tunnel.
	Comparing configuration for backdoors or broken flows after route changes
Q7a:	$\mathcal{C}_{org} \triangleq [\neg multiroute \wedge src = a1 \wedge dest = a2 \wedge loc(a1) \rightarrow \mathbf{AF}(loc(a2) \wedge src = a1 \wedge dest = a2)]$
Q7b:	$\mathcal{C}_{new} \triangleq [multiroute \wedge src = a1 \wedge dest = a2 \wedge loc(a1) \rightarrow \mathbf{AF}(loc(a2) \wedge src = a1 \wedge dest = a2)]$
Q7:	*Backdoors:* $\neg\mathcal{C}_{org} \wedge \mathcal{C}_{new}$, *Broken flows:* $\neg\mathcal{C}_{new} \wedge \mathcal{C}_{org}$
	what is different in the new configuration as compared with the ordinary original one. Is there any backdoor?

Evaluation

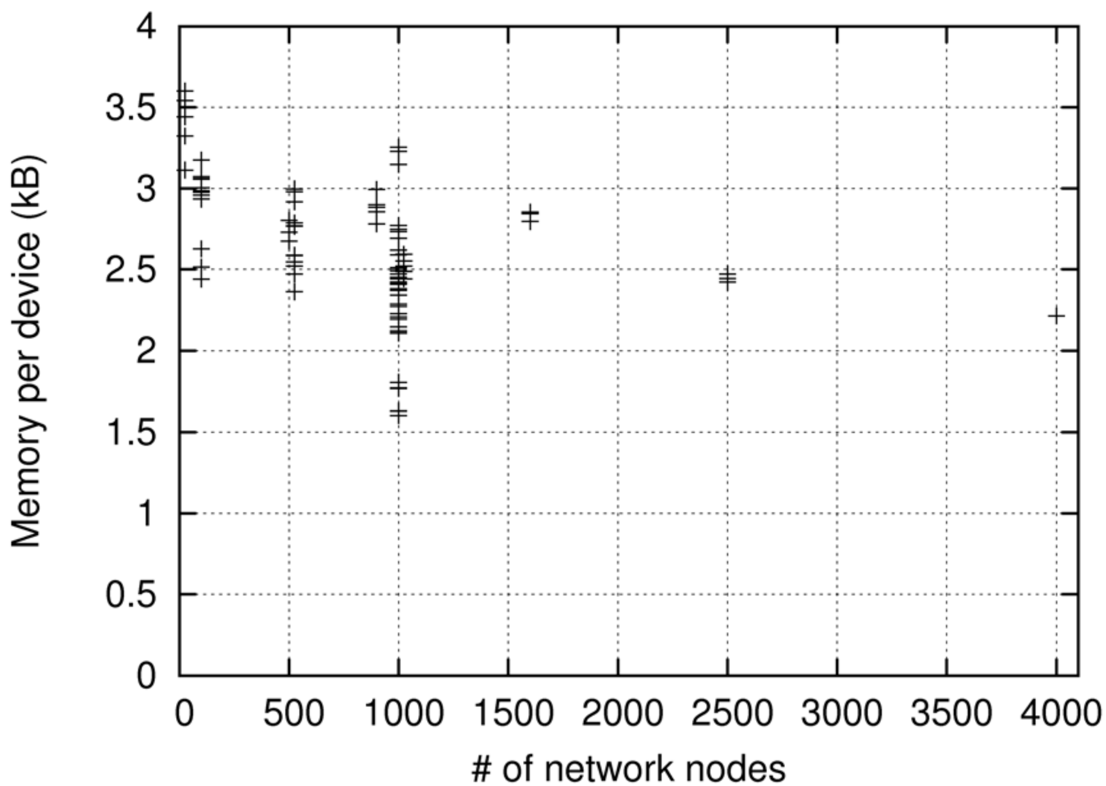
- Using 90 networks with real and random network configuration
- Random (yet reasonable) configuration is important
- Random Policy/Configuration Generation
  - Hierarchical topology network
  - Evaluation parameters: network size, policy size, rule interaction/overlapping, subnet distribution, branching factor or network depth vs. breadth, device type
  - BDD can handle up to 30K rule per device
  - Created 4000 nodes and 6M rules
  - **Details, examples of format, and configurations can be found in http://www.cyberDNA.uncc.edu/projects/ConfigChecker**
- We measure the space requirement and building time
  - Query time is negligible in most of the case

Evaluation



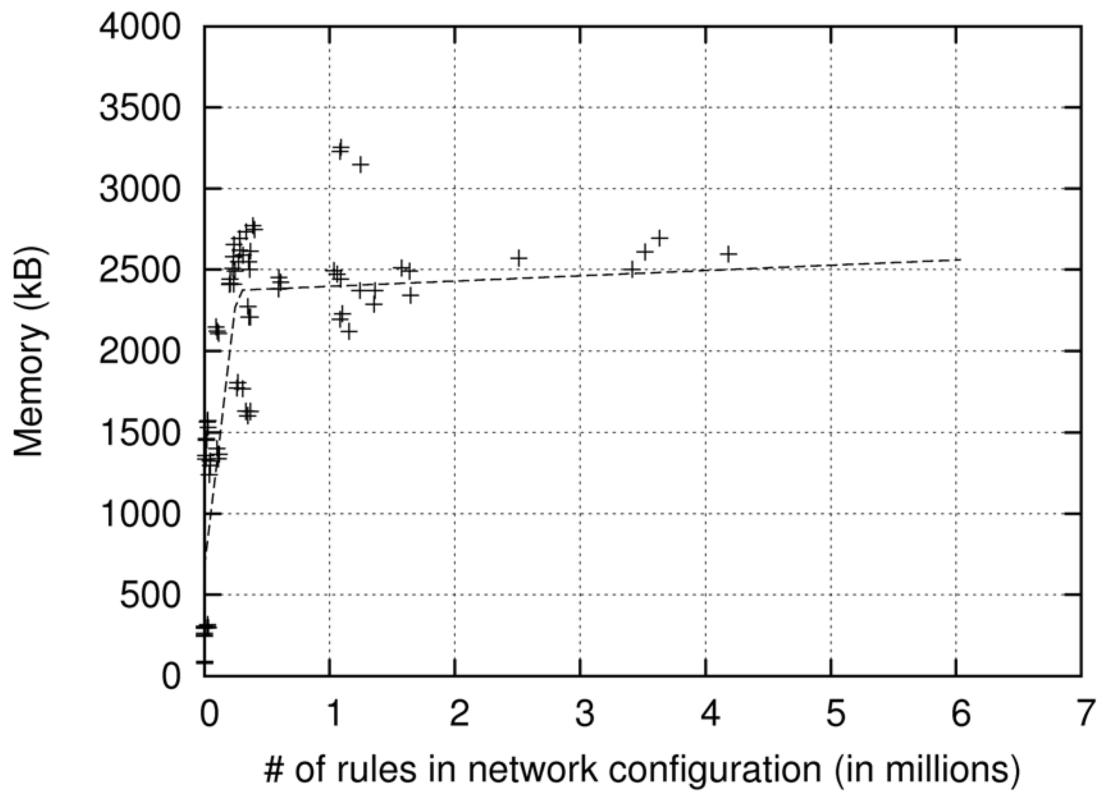
- Memory Required versus Network size
  - The growth is evidently linear in both transition relation size and in overall BDD table entry count.

Evaluation



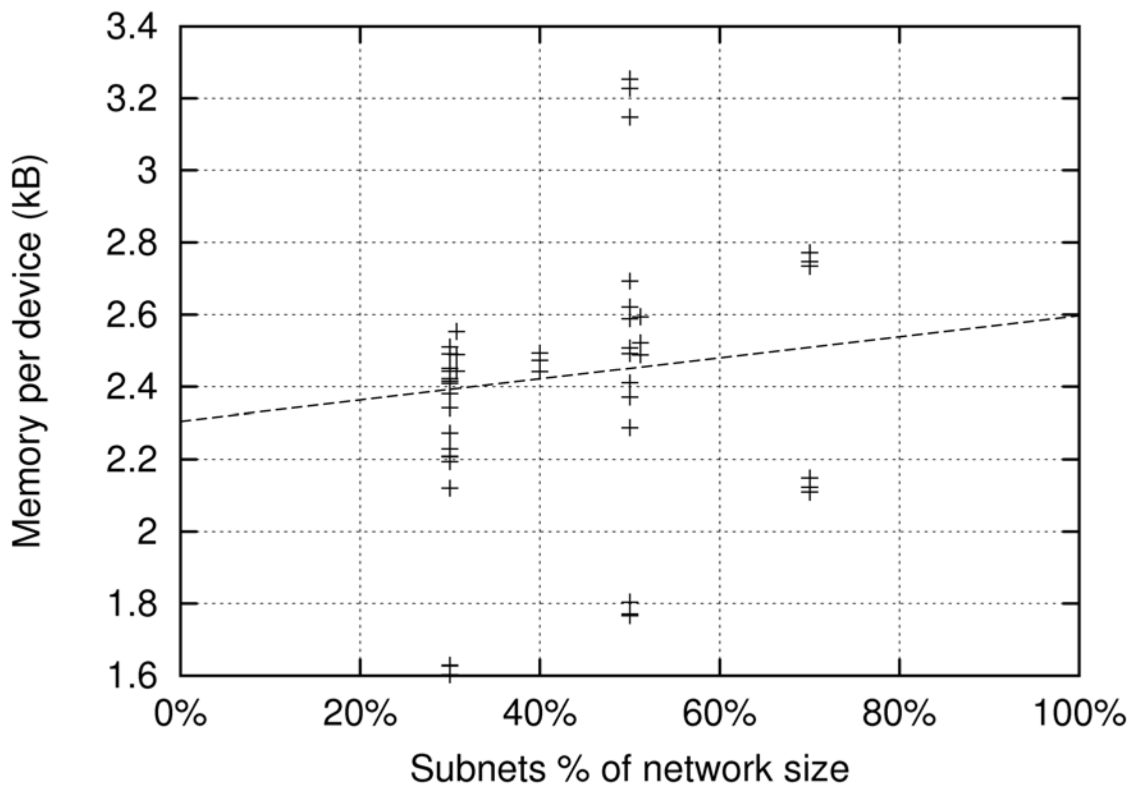
- Memory Required per device versus Network size
  - Almost constant

Evaluation



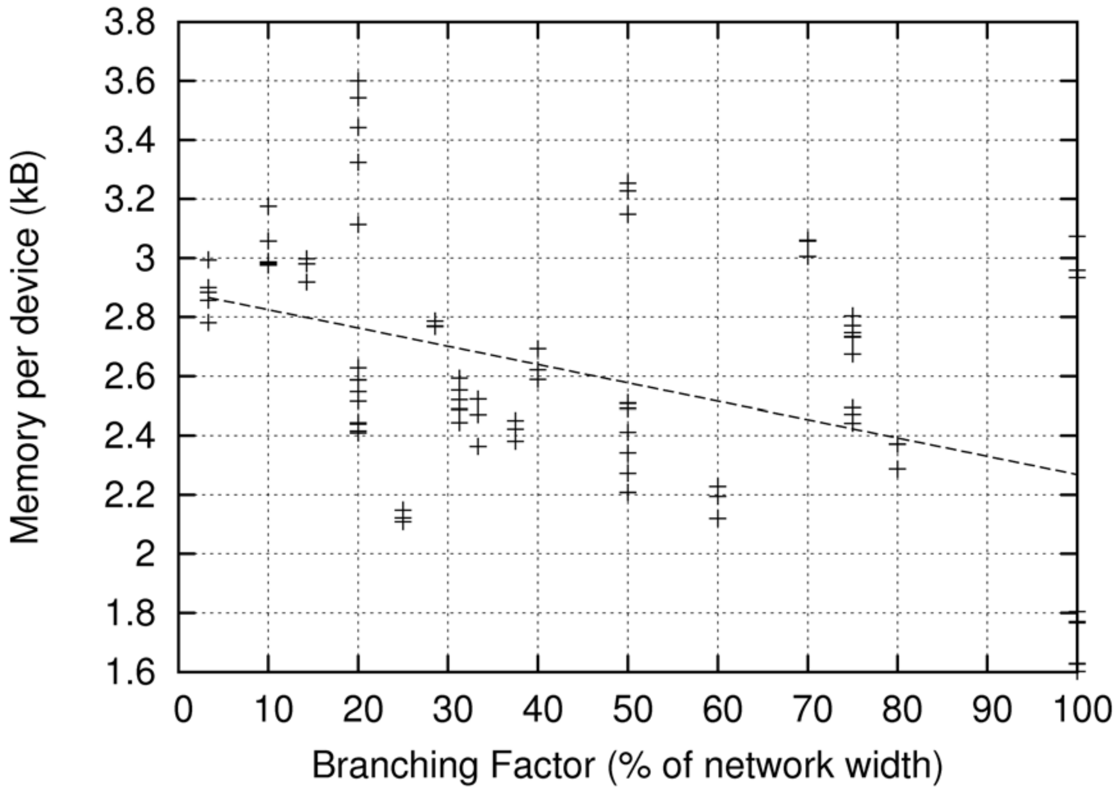
- Space versus number of rules
  - Increase then almost steady state

Evaluation



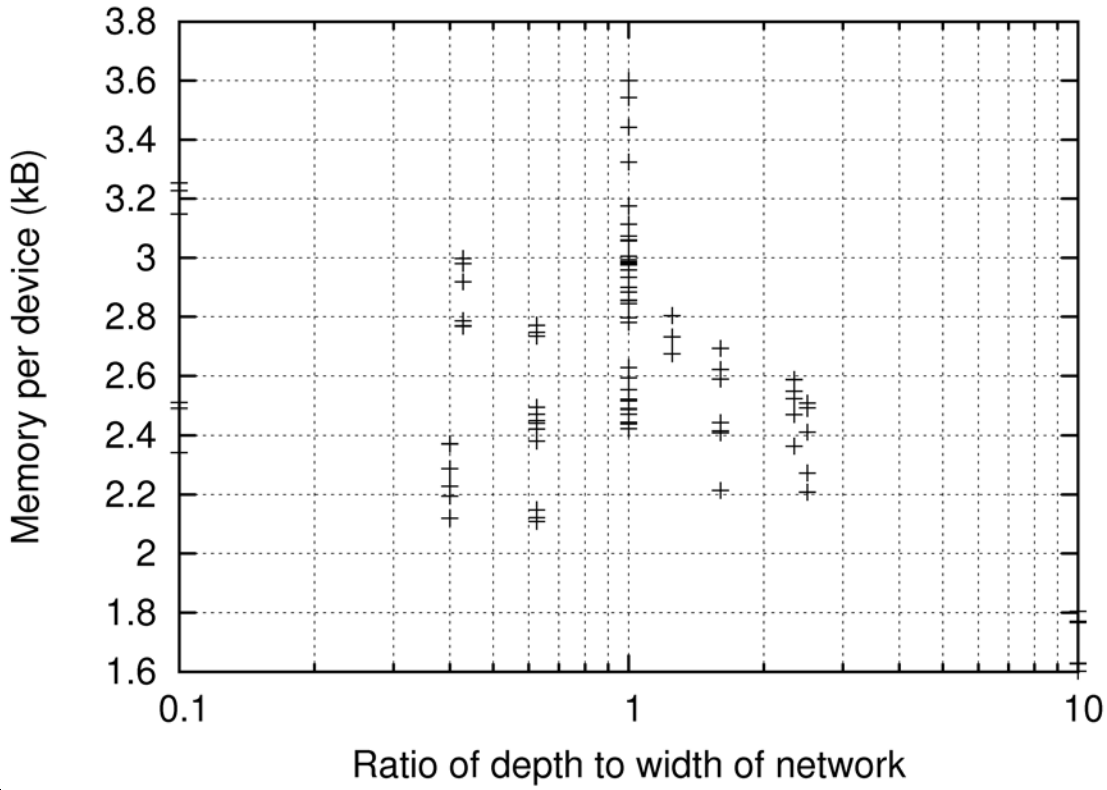
- Space versus number of rules
  - Increase then almost steady state

Evaluation



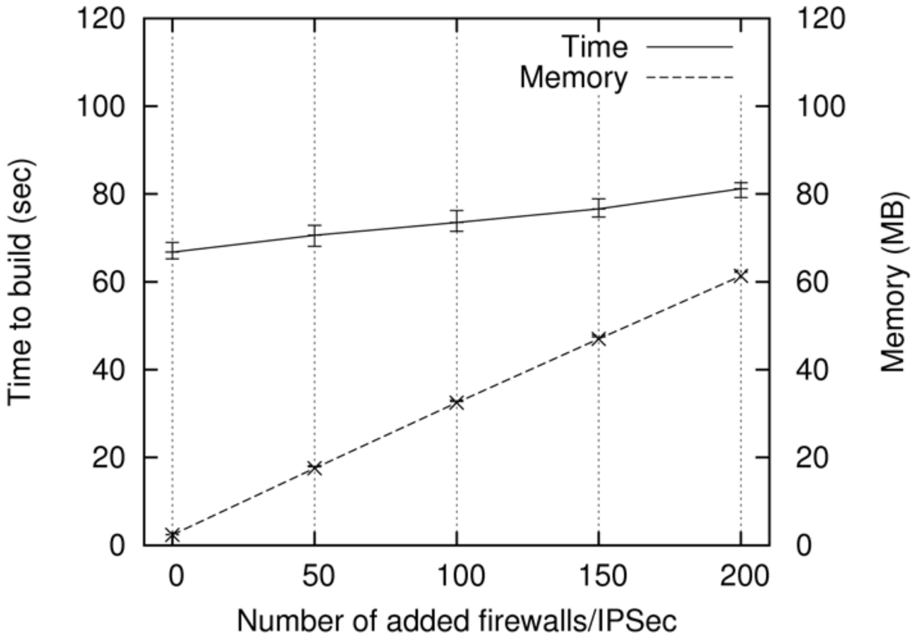
- Effect on Branching Factor on memory required per device

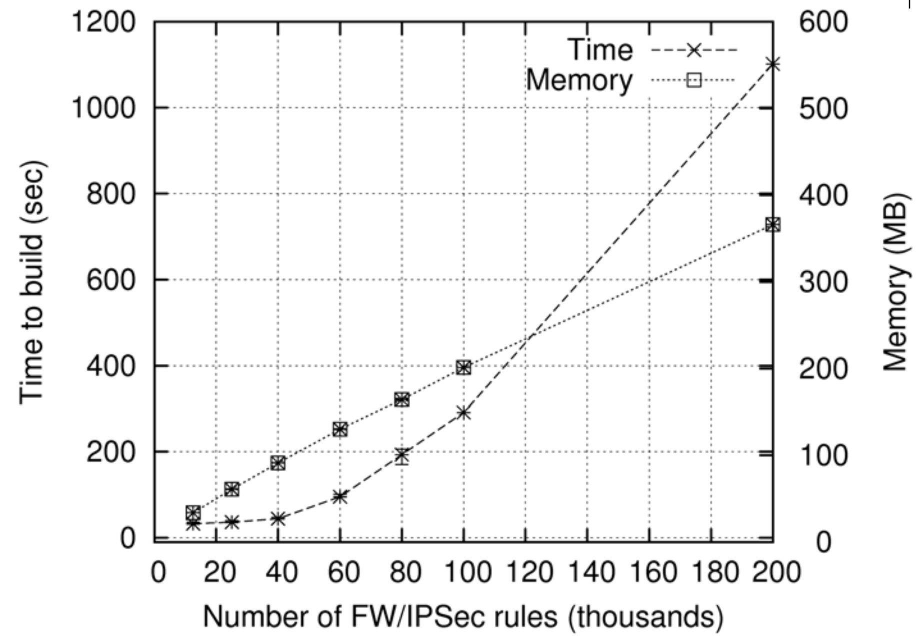
Evaluation



- Effect on depth to width of network on the memory required per device

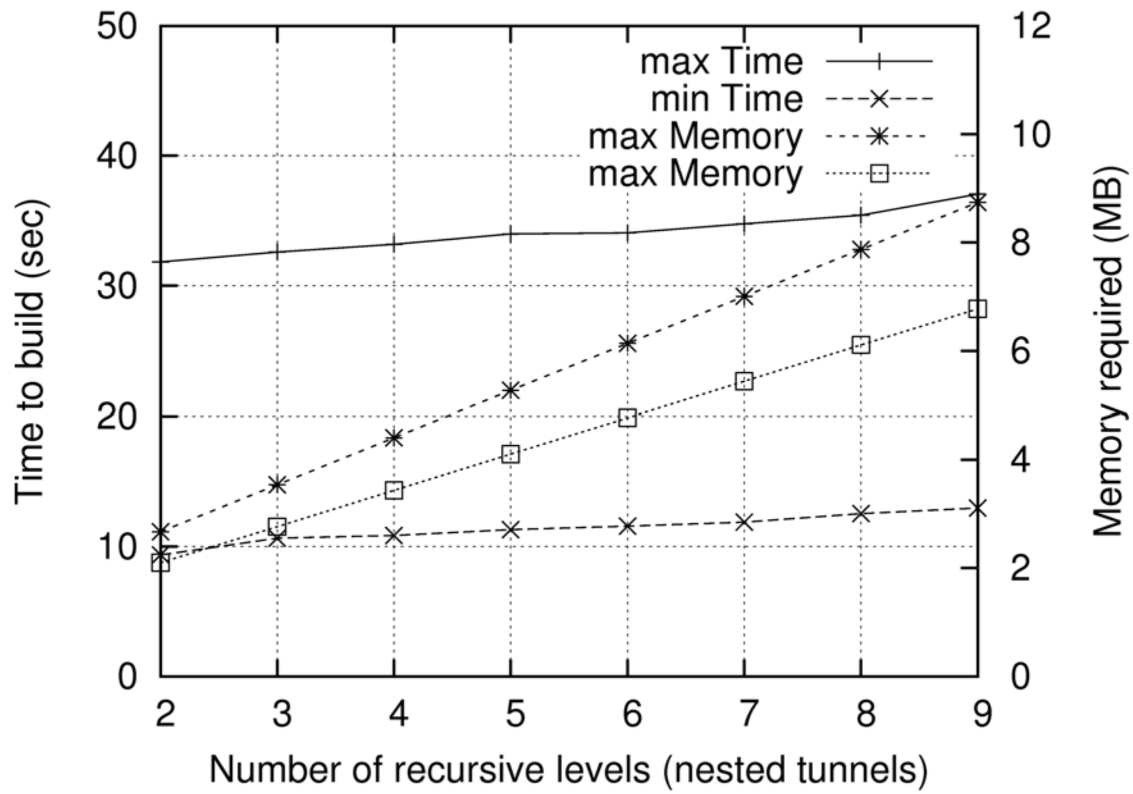
Evaluation





- Effect of number of firewalls, and the size of each of their policies on overall model memory requirement

Evaluation



- Effect of number of nested tunnels supported in the model on the memory and time required to build the overall model.

Summary of Evaluation

- Configchecker looks scalable for this application domain
  - 4K nodes and 6+ Millions of rules ➔ Max 14M and order of minutes
  - O(V) instead of O(V^3) – ignoring the cost of set/bdd operations
  - Wildcard; common prefixes; overlapping rules, and variable ordering
- Supporting rich and logically expressive interfaces such as CTL is powerful and important, although clumsy for regular users

Conclusion -- Future Challenges

- Proactive Defense
  - On-line automation for misconfiguration and fault detection and repair
  - On-line Threat Assessment (identification and impact)
  - Real-time monitoring & response for intrusion
- Insider threats (SG is semi-closed networks)
- Agility
  - Tolerance, Self healing, Survivability
- Real-time Monitoring and Response
  - Intrusion Response Systems
  - Fault / misconfiguration mitigation
- Non-invasive Static Analysis (vs. penetration testing)
- Non-intrusive (Light weight) IDS due to limited resources
- Patch management for smart grid – scalability and agility

UNC CHARLOTTE
College of Computing and Informatics

Questions/Comments!

References

- Bin Zhang and Ehab Al-Shaer, **Towards Automatic Creation of Usable Security**, IEEE INFOCOM 2010 Miniconference, April 2010
- Ehab Al-Shaer, Will Marrero, Adel El-Atawy, and Khalid Elbadawi, **Network Security Configuration in A Box: End-to-End Security Configuration Verification**, IEEE International Conference in Network Protocols (ICNP' 09), October 2009
- Mohamed Salim, Ehab Al-Shaer and Latif Khan, **A Novel Quantitative Approach For Measuring Network Security**, INFOCOM 2008 Mini Conference, April 2008
- Ehab Al-Shaer, Latif Khan and M. Salim Ahmed, **A Comprehensive Objective Network Security Metric Framework for Proactive Security Configuration**, ACM Cyber Security and Information Intelligence Research Workshop, Oak Ridge, Tennessee, USA, May 2008
- Mohamed Salim, Ehab Al-Shaer, Mohamed Taibah, Mohamed Arshad and Latif Khan, **Towards Autonomic Risk-aware Security Configuration**, Accepted in the 11th IEEE/IFIP Network Operations and Management Symposium (NOMS 2008), April 2008