Managing Security in Dynamic Networks

Alexander V. Konstantinou
Yechiam Yemini
Columbia University

Sandeep Bhatt
S. Rajagopalan
Telcordia Technologies
(formerly Bellcore)
Overview

1. Dynamic Network Example
2. Automating Network Configuration
3. NESTOR Architecture
4. Example Revisited
5. Future Work
Dynamic Networks

- *Network*: elements, services, and policy
- *Dynamic Network*: components may change

**Goal**: manage configuration to maintain policy through change
Configuration Mgmt is Difficult

• Human-intensive
• Distributed heterogeneous data
  – Single task involves multiple elements
  – Duplication & dependencies
• No verification of integrity rules
• Manual recovery

Static configuration & network failure
Dynamic Network Example

- Consultant visiting client needs to access home directory
- **Goal:** Plug laptop & double-click on home folder
Example Security Policies

Client

• No visitor access to internal hosts
  – switch, router, physical configuration
• Restricted visitor Internet access
  – firewall configuration

Consultant

• VPN clients obtain restricted file access
  – file, http, ftp server configuration
Solution: Unified Configuration Semantic Layer

- Unified object-relationship configuration model
- Consistency rules
- Change propagation
- Rollback and recovery
NESTOR: An Architecture for Network Self Management & Organization

- Create transaction
- Lookup object
- Update
- Commit

Transaction Manager

commit
abort

subscribe
notify

Constraint Manager

IpHost
String name;
reboot()

<<constraint>>
All host names must be unique

Object Repository

Adapter Layer

NFSd
NISd
Switch
Router
Host

SNMP Adapter
Integrity Constraint Example

- Constraints expressed in OCL (Object Constraint Language -- part of UML)
- Example: “All nodes connected to an internal VLAN port should be trusted”

\[
\text{EthernetVlanSwitchPort} \to \text{allInstances} \\
\to \text{select} (\text{port} \mid \text{port.isEnabled}) \\
\to \text{forall} (\text{port} \mid \\
\text{if} (\text{port.securityMgr.isTrusted(port.vlanID)}) \\
\text{port.forwardNodes} \to \text{forall} \\
(\text{node} \mid \text{node.securityMgr.isTrusted(node)})
\]
Policy Script Example

- Constraint violations handled by policy scripts (Java methods)
- Example (cont.): policy script changes the VLAN id of the violating port

```java
public void constraintHandler
    (Object[] stack, Transaction trans) {
    EthernetVlanSwitchPort port =
        (EthernetVlanSwitchPort) stack[1];
    port.vlanID =
        port.securityMgr.getPublicVlanID();
}
```
Dynamic Network Example Revisited

- High-level security policies
- Model network elements & services
- Instrument model interfaces
- Policies as constraints on configuration
- Policy scripts for change propagation
- Deploy and populate NESTOR server
Laptop Plug-In Interactions

: Laptop
 : Switch
 : NESTOR
 : Constraint Mgr

plug-in

SNMP GET

create transaction
update model, commit
verify()

model update
commit()

SNMP SET

execute policy script
Laptop Plug-In Interactions (2)

:Laptop authenticate

:VPN-Server notify()

:NESTOR

1. create transaction,
2. update model,
3. commit

1. constraint violation,
2. execute policy script

commit (constraint mgr)

update access list

:NESTOR

:NFS-Server
Summary

- Dynamic network challenges
- Solution: unified configuration semantic layer
- NESTOR architecture
- Policy-based dynamic network configuration

Future SA role: defining policies for change propagation
Future Work

• Translating high-level security policies to constraints on configuration (Telcordia)
• Model evolution (Telcordia project on reconfiguring networks of firewalls)
• Scalability
• NESTOR security model
• Distributing NESTOR/pushing down to device
Managing Security in Dynamic Networks

Alexander V. Konstantinou
akonstan@cs.columbia.edu

http://www.cs.columbia.edu/dcc/nestor

Yechiam Yemini (yemini@cs.columbia.edu)
Sandeep Bhatt (bhatt@research.telcordia.com)
S. Rajagapalan (sraj@research.telcordia.com)
Backup Slides
Configuration Modeling

- Model expressed in the MODEL language (SMARTS)
- MODEL extends IDL with relationships, problems ...

```java
interface nestor::IpHost : nestor::ManagedObject {
    attribute String hostname "Name of host";
    relationshipset interfacedThrough,
    IpNetworkInterface, partOf;
}
```

```
Host.mdl  >  model2java compiler  >  Host.java (interface)  >  Modeler (+ adapter lib.)  >  SnmpHost_ Adapter .java
```

USENIX Lisa'99  Alexander V. Konstantinou  19
NESTOR Transactions

• Proxy repository objects
  – Implement model interfaces
  – Log all access
  – Updates not pushed to device

• Transaction commit
  – Effect all changes on proxy objects to adapter objects (same order)
  – On failure, roll-back
  – On roll-back failure, note in recovery log