Towards Software Defined Zero Trust Networks

Doug Montgomery (dougm@nist.gov)

Traditional Enterprise and WAN Networks

- **Traditional Private WAN:**
  - Data center VPN back haul.
  - Decoupled network & application policy.
  - Limited control / computing at network edge.
  - Limited ability to exploit / manage multiple communication paths.
  - Disjoint / inflexible measurement / monitoring and security awareness.
Software Defined & Virtualized Networks

- **SD WAN + NFV:**
  - Application / user aware policy at network edge.
  - Micro segmentation of traffic based upon managed policy.
  - Distributed virtualized network functions.
  - Application / users policy aware telemetry at network edge.
  - Scalable holistic network monitoring and anomaly detection.
  - Automated load balancing over multiple transport sources.
Software Defined & Virtualized Networks

• NIST Research & Standards Efforts:
  • Application / user aware policy at network edge.
  • Micro segmentation of traffic based upon managed policy.
  • Distributed virtualized network functions.
  • Application / users policy aware telemetry at network edge.
  • Scalable holistic network monitoring and anomaly detection.
  • Automated load balancing over multiple transport sources.

Software Defined Security for IoT
Programmable Measurement and Monitoring for Software Defined Networks
Trustworthy Networking

• ISOC 2017 Report on the Future of the Internet
  • “Perhaps the most pressing danger to the future of the Internet is the rising scope and breadth of Cyber Threats.”
  • “Addressing cyber threats should be the priority”
  • “The scale of cyberattacks is steadily growing, and many anticipate the likelihood of catastrophic cyberattacks in the future.”
  • “Inadequate management of cyber threats will put users increasingly at risk, undermine trust in the Internet and jeopardize its ability to act as a driver for economic and social innovation.”

• Cultivating Trust is not Easy …
  • Challenges are technical, economic, often dominated by prevailing business models, complicated by massive installed bases, and fears of governmental interference.
Network Resilience Program

• Understanding / Controlling Network Behavior
  • “[Despite] society’s profound dependence on networks, fundamental knowledge about them is primitive. Global communication networks have quite advanced technological implementations but their behavior under stress still cannot be predicted reliably…. There is no science today that offers the fundamental knowledge necessary to design large complex networks [so] that their behaviors can be predicted prior to building them.” Network Science, a report from the National Research Council [4].

• The Need for NIST:
  • Advance Network Metrology – with emphasis on innovating and applying advanced measurement science to Internet-scale systems.
  • Foster Trustworthy Network Technology – work with industry to improve the quality and timeliness of emerging specifications and foster adoption of trustworthy Internet technologies.
  • Our efforts focus on Internet Scale problems, solutions and measurement techniques.
What are NIST’s Roles?

- Problem Identification
- Requirements Analysis
- Threat Modeling
- Problem Space Characterization
- Define USG R&D Priorities
- Technical Basis for Policy
- Deployment Requirements
- Empirical Data and Analysis
- Test and Measurement
- Protocol Design
- Protocol Prototypes & Models
- Pilot Deployment & Operational Analysis
- Consensus Standards
- Deployment Guidance
- Pilots / Testbeds

Network Resilience Program
Related Program Areas.

- Robust Inter-Domain Routing – Kotikalapudi Sriram & Oliver Borchert

- High Assurance Domains – Scott Rose

- Measurement Science for Complex Systems – Kevin Mills

- Software Defined and Virtual Networks – Yang Guo
Establishing the Technical Basis for Trustworthy Networking

Software Defined Security for Scalable IoT Defense

Establishing the Technical Basis for Trustworthy Networking

Doug Montgomery, Mudumbai Ranganathan, Charif Mahmoudi, Laurence Chang, Max Kimmelman

Things on the Internet

• Explosion of networked things
  • Key enabler for smart environments (e.g., homes, transportation, health).
  • Things themselves often limited in capabilities / uses.

• Cloud based business models
  • Things often controlled, accessed, provide data to cloud based services.
  • Creates giant attack surface for networked things.
Hackers Like Things Too

• **Things are vulnerable targets**
  • Typically not general purpose computers, thus lack the ability to protect themselves.
  • Often “networked” as an add-on to original design.
  • Poorly maintained – lack ability for secure software update, or are no longer supported.

• **Hacked things …**
  • Used to disable / alter their basic service.
  • Used to attack other systems on the network.
Pragmatic Solutions

• Thing manufacturers …
  • ..are in a position to significantly leverage this problem.
  • They know what their thing is supposed to do!
  • Network needs to know:
    • What is this thing?
      • Who made it?
      • Who owns it?
    • What network access does it need?

• Manufacturer Usage Description
  • IETF specifications under development.

• Strict Policy Based Networking
  • Any commination not explicitly specified MUP profile is not permitted.
Software Defined IoT Defense

• Project Goals:
  • Explore future architectures for software defined IoT Defense:
    • Support policy driven security profiles for individual device types.
      • Move to a policy-based model of network security for limited functionality devices.
  • Support authentication of attached devices / types.
    • Authentication to the network and within the network.
  • Support complex consumer networks and requirements for segmentation and security within the consumer network.
    • Potentially controlled by the manufacturer / consumer.
  • Support policy driven security profiles for Internet access.
    • Controlled by the network operator.
  • Support both proactive and reactive defense mechanisms using virtualized IDS / Firewall functions.
So How Will MUD Work?

Device Manufacturer

MUD File Server

Usage Description

Device emits MUD URL: https://www.example.com/.well-known/mud/v1/model/version

Dev: SW, Port, MAC,
https://www.example.com/.well-known/mud/v1/model/version

Usage

Description

Internet

SW

NPE

CPE

ACL

ACL

ACL

ACL

Device emits MUD URL:
https://www.example.com/.well-known/mud/v1/model/version

Software Defined Security for Scalable IoT Defense
• What is this thing?
  • MUD URL

• What network access does it need?
  • MUD File
  • YANG model of extended access control lists (ACLs).
  • Meta data for MUD Controller
MUD Profiles

```
"ietf-mud:mud": {
    "last-update": "2017-10-16T22:10:33+02:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "https://mud.nist.gov/toaster",
    "from-device-policy": {
        "access-lists": {
            "access-list": {
                "acl-name": "mud-42646-v4to",
                "acl-type": "ietf-access-control-list:ipv4-acl"
            }
        },
        "to-device-policy": {
            "access-lists": {
                "access-list": {
                    "acl-name": "mud-42646-v4fr",
                    "acl-type": "ietf-access-control-list:ipv4-acl"
                }
            }
        }
    }
}
```

```
"ietf-access-control-list:access-lists": {
    "acl": [
        {
            "acl-name": "mud-42646-v4to",
            "acl-type": "ipv4-acl",
            "aces": {
                "rule-name": "cl0-todev",
                "matches": {
                    "ipv4-acl": {
                        "ietf-acldns:src-dnsname": "www.nist.gov",
                        "protocol": 6,
                        "source-port-range": {
                            "lower-port": 443,
                            "upper-port": 443
                        }
                    }
                },
                "actions": {
                    "forwarding": "accept"
                }
            }
        },
        {
            "rule-name": "ent0-todev",
            "matches": {
                "acl-name": "mud-42646-v4to",
                "acl-type": "ietf-access-control-list:ipv4-acl"
            }
        }
    ]
}
```
Phase 1 Prototypes

- **SDN-based MUD Architecture**
  - Simple MUD Controller,
  - SDN ACL application
  - MUD DHCP Client / Server Extension

- **SDN-based CPE**
  - Integrated OpenVswitch on consumer grade CPEs.
  - ACLs / Segmentation on ports and wifi.

- **OpenStack Elastic NFV IDS**
  - Load balanced Snort using open stack components.
Prototype SDN / MUD Controller

- SDN Aware MUD Controller
  - Maps MUD ACLs into SDN Flow rules.
  - Resolves late bindings from MUD profiles and local network context
Multiple Flow Tables

- Mapping of Network Control Policies to Flows
  - Proactive MUD Policies
  - Reactive IDS Policies
Elastic IDS NFV

- SDN-Based Programmable Port Mirroring
  - Suspicious flows diverted to deep packet inspection
- Malicious Flows Blocked by SDN ACL Rules.
  - Reactive ACLs use similar interface as proactive MUD ACLs
Scaling IDS NVF

- OpenStack Virtualization Environment
  - Using industry standard components for virtualization, load balancing, orchestration, and event management.
Reactive IDS Rules

- OpenStack Telemetry and Alarming Infrastructure
  - Process IDS alarms into REST API calls to ODL Enforcer application

```python
module ids {
  list vm {
    description "Virtual Machine";
    key VMID;
    leaf VMID {
      description "VM Id";
      type string;
    }
    list Port {
      description "Port";
      key PortID;
      leaf PortID {
        description "Port Id";
        type string;
      }
      container actions {
        description "Define Action Criteria";
        choice port-handling {
          default deny;
          case deny {
            leaf deny {
              type empty;
              description "Deny action";
            }
          }
          case permit {
            leaf permit {
              type empty;
              description "Permit action";
            }
          }
        }
      }
    }
  }
}
```
Future Work

- **Research Prototypes → Industry Reference Implementations**
  - Re-implement SDN MUD Controller on OpenDaylight SDN Controller.
  - Use industry standard YANG models and tools to generate APIs.

- **Full Support of MUD Controller Semantics**
  - Support for MUD Controller classes (late binding) and other abstractions.
  - Stateful ACL rules, management of topology events, ACL placement algorithms.
  - Signature verification on MUD files.

- **Implement MUD support on widely used DHCP server**
  - DNSMasq for CPEs, BIND and/or OpenDaylight for Enterprise
  - Explore other means of announcing MUD URL – LLDP, IEEE 802.11AR, ANIMA.

- **Explore other means of generating MUD profiles.**
  - Use of machine learning and other means to create MUD profiles of legacy systems.
  - Explore usage profiles created by local network operator.

- **Enhance elastic IDS NVF**
  - Support topology placement algorithms and IDS optimization for different classes of traffic.
  - Couple with elastic IDS system for non-MUD devices.

- **Publish research results and release reference implementations.**
Questions and Discussion

• For more information:
  • Network Resilience Program
    • https://www.nist.gov/itl/antd/internet-scalable-systems-research
  • Advanced Network Technologies Division.
    • https://www.nist.gov/itl/antd
  • Information Technology Laboratory
    • https://www.nist.gov/itl