CoordiNetZ: Coordinated Dataflow Protection for Ultra-High Bandwidth Science Networks

Vasudevan Nagendra

Joint work with
Vinod Yegneswaran, Phillip Porras, Samir R. Das

Stony Brook University  SRI International

ACSAC 2019
Agenda of the Talk

- Background on ESNet & SDMZ
- SDMZ Requirements / Limitations
- CoordiNetZ Architecture
  - Functional Components
- Evaluations
Agenda of the Talk

• Background on ESNet & SDMZ

• SDMZ Requirements / Limitations

• CoordiNetZ Architecture
  • Functional Components

• Evaluations
Background: Energy Sciences Network (ESNet)

**Project (P1):**

**Project (P2):**

ESNet: Largest Science Networks in the World
Background: Energy Sciences Network (ESNet)

ESNet: Largest Science Networks in the World

40+ DoE sites connected with dedicated 100Gbps WAN links (within & across countries)
140 Campus networks peered to ESNet
50 Petabytes per month
Background: Science DMZ Network Architecture

DoE/Campus Site1 SDMZ

DoE/Campus Site2 SDMZ

Switch

Border Router

SDMZ Core / Internet

X *10Gbps

X *10Gbps

Switch

Border Router

SDMZ Core / Internet

X *10Gbps

X *10/100 Gbps

DTNs

IDS

Switch

Edge Firewall

Elephant Flow ≥ 10Gpbs

LAN Hosts

DoE/Campus Site1 LAN

DoE/Campus Site2 LAN

Reference: https://www.es.net/assets/pubs_presos/sc13sciDMZ-final.pdf

Isolated from Stateful Firewalls & DPI devices for performance
Background: Science DMZ Network Architecture

Science DMZ: Network isolated from stateful firewall/DPI devices

Network with 0.0046% packet drops in TCP-based elephant flows with RTT greater than 20msec could result in **10X drop in throughput**.

Reference: [https://www.es.net/assets/pubs_presos/sc13sciDMZ-final.pdf](https://www.es.net/assets/pubs_presos/sc13sciDMZ-final.pdf)

Isolated from Stateful Firewalls & DPI devices for performance
**Background: Collaboration among projects across sites**

**Sites**

- **Project 1**: Site 1, Site 2
- **Project 2**: Site 1, Site 3, & Site 4

**Hosts**

- **Project 1**: H1, H2, H3, H4, H5, H6
- **Project 2**: H3, H4, H5, H6, H7, H8

**Lacks Isolation across projects & Infrastructure**
Background: Collaboration among projects across sites

Sites\{project\}:
- Project1: Site1, Site2
- Project2: Site1, Site3 & Site4

Hosts\{project\}:
- Project1: H1, H2, H3, H4, H5, H6
- Project2: H3, H4, H5, H6, H7, H8

Lacks Isolation across projects & Infrastructure
Agenda of the Talk

• Background on ESNet & SDMZ

• SDMZ Requirements / Limitations

• CoordiNetZ Architecture
  • Functional Components

• Evaluations
SDMZ Requirements (1): Intuitive & Unified Policy Specification

Data-flow policies

Policy (P1)

D1: Sensitive Data of Exp1

S1 \{P1(E1)\} \rightarrow S2 \{P1(E1)\}

Temporal and spatial policies

Policy (P2)

D2: Sensitive & Export Controlled

Time >9PM <7AM

S1 \{P2\} \rightarrow S2 \{P2\}

Dynamic Policy

Policy (P3)

D3: Sensitive Data & Network under Brute-force

Notify Admin & Quarantine

S1 \{P2\} \rightarrow S1 \{P2\}

Dynamic Data flow-based policies
Tracking data flows and enforcing rules is challenging.
SDMZ Requirements (3): Isolation in Shared infrastructures (multiple projects & sites)

Site 1 specific policies: Project 1 & Project 2

Rules pertaining to projects (1 & 2) for Host DTN 1

Policy Framework

Switch1:
P1: Host1 -> Internet
P2: Host1 !-> Internet

Site 1: Research / Production DTN Nodes
Site 2: Research / Production DTN Nodes

Isolation in Abstraction & Fine-grained Policy Enforcement
SDMZ Requirements (4): Context-aware security enforcement

**Context**
- Context-awareness and aggregation for enforcement

**Dynamic data transfer ports (FTP PORT Command):** 9001:9025

**Globus FTP Control**

**GridFTP Source DTNs**

**GridFTP Target DTNs**

**SDMZ Core**

**Clustered IDS**

**Challenge:**
- Proactively specifying rules for enforcement on unknown data ports.

**Context-awareness and aggregation for enforcement**
SDMZ Requirements (4): Context-aware security enforcement

**Context**
- Context-awareness and aggregation for enforcement

**Dynamic data transfer ports (FTP PORT Command):** 9001:9025

**Challenge:**
Lack of context to detect distributed attacks
  - e.g., DDoS, data exfiltration, network scans and so on

**Context-awareness and aggregation for enforcement**
Limitations: Science DMZ

1. No stateful inspecting devices along data path
   - Offline DPI & coarse-grained security
   - E.g., IDS, shunting techniques, Router/switch ACL

2. Lack of fine-grained data security
   - Varying levels of sensitivity, security, privacy and compliance.
   - Light-weight data tracking
   - Fine-grained enforcement

3. Policy specification for non-admin SDMZ users
   - Multi-administrative domain
     - multiple projects spans across multiple sites
   - E.g., Researchers, Scientists & Professors
   - Isolation In Abstractions
Agenda of the Talk

- Background on ESNet & SDMZ
- SDMZ Requirements / Limitations
- CoordiNetZ Architecture
  - Functional Components
- Evaluations
Our Contributions: Performance, Programmability & Security Challenges

Science DMZ Networks

CoordiNetZ:

- Intent-based Graph policy framework
  - Graph-based specification
  - Fine-grained data-flow policies
  - Graph-Composition Techniques

- Context-aware Security
  - Host & network context
  - Stateless microservices

- Inter-site & Intra-site context-aware tagging
CoordiNetz: High level System Architecture

CoordiNetz Dashboard

Cross-Site Coordinator for Context-aware Tagging and Security Enforcement

SDMZ Core (tier0/1) Sites

Coordinated, Context-aware & fine-grained dataflow-based policy specification
CoordiNetz: High level System Architecture

Graph-based Policy Specification
- Tree-based Abstractions
- Data-specific policies
- Dataflow tracking

Inter-site & Intra-site Tagging
- Enforcement beyond sites
- 20 bits of IPV6
- Intra-site tag assignment
- Inter-site tag space allocation

Conflict detection & resolution
- Dataflow-based graph composition algorithms

CoordiNetz Dashboard

Cross-Site Coordinator for Context-aware Tagging and Security Enforcement

Coordinated, Context-aware & fine-grained dataflow-based policy specification

Light weight stateless Microservices
- Spoofing Protection, Data Exfiltration, Collaborative Protection
- On-demand security services with lightweight microservices
CoordiNetZ: Key components

1. Host DTN
   - SciMon
   - SciFlow

2. CNZ Controller

3. CNZ Coordinator

4. Stateless microservices
SciMon: Science DMZ Monitor for Host DTN

- Track data flows to generates data-flow records
  - Sent to CNZController (-> CNZ Coordinator) to build data-flow graphs

SciMon Flow Record:

# [SciMon]: username, hostname, processID, appname, execpath, execArguments, execCredential, openFileList, integrity, pProcessID, pAppname, pExecPath, sensorID, sensorVer

- Contextual information required for host-level enforcement
  - WHO (user/applications/process/experiment), WHAT (file/network I/O), HOW (remote login), WHEN (timestamp), WHERE (country, city, IP).

- Monitor and enforce host/process-specific data policies.
  - Tag-based policies (IPv6 Flow label)
SciFlow: Science Flow tracking for Host DTN

- Provides additional flow-specific context compared to NetFlow
  - e.g., DNS transaction summaries, unfinished SYN handshakes, unsolicited ACKs, ACK timeouts, IP address reputation, geography information (domain, country, city, latitude and longitude)

SciFlow flow record:

```
```
CNZ Controller (1): Context & Reconciliation

- Collects host/process-layer context
- Consolidates flow records and forwards to Coordinator
- Project-specific, site-specific, and host-specific rules for policy enforcement
- Triggers SDN Controller to insert flow rules for filtering malicious traffic

Context from host & network to tag the packet
CNZ Controller (2): Intra-site tag assignment

- **Traditional Approach:** Bit per network attribute
  - Projects, experiments, hosts, users, and so on

- **CNZ Controller** locally assigning tags to each policy
  - Assign contiguous tags to policies having:
    - Same action attributes
    - Grouped together using bit masking.

- # tags required is approximately equals the number of conflict-free policies.

Tag Assignment Algorithm for optimizing tag utilization
CNZ Coordinator (1): Tree-based Infrastructure Abstractions

(a) Abstraction tree for experimental data outcome (AM = data{*}.experiment {Exp1}).

(b) Abstraction tree for dynamic host security (AM = security-state{*}.site {Site1}:hosts{*}).

(c) Network vs Host-specific abstractions of Site 1. buildings{*}. site{Site1}:networks{*}:hosts{*}

Abstractions for Isolation & Intuitive Policy Specification
CNZ Coordinator (2): Dataflow tracking Dashboard (within host & across geographies)

(a) Experimental data transformation

(b) Dataflow tracking across sites

Lightweight dataflow tracking dashboard
CNZ Coordinator (3): Graph-based ACL policy specification

Spec{Site-Admin}: src-node{BLDG1}.parent-path{Hosts_Site1}.traffic-type{"*"} >> FW >> WAN-Accelerator => target-node{Internet}.parent-path{networks}

Spec{Site-Admin}: src-node{Net1}.parent-path{Hosts_Site1->BLDG1}.traffic-type{"*"}.time{Time3}.state{compromised} >> DPI => target-node{Net6}.parent-path{Hosts_Site1->BLDG2}

Spec{Project-Admin}: src-node{Site1}.parent-path{Hosts_Project1}.traffic-type{GridFTP}.data{D1/*} !=> target-node{Site2}.parent-path{Hosts_Project1}
Simplified Data Policy Syntax:

- site{Site₁} → data{D₁} → site{Site₂}
- site{Site₁} → data{D₁} → site{Site₃}
- site{*} → data{D₁/*} !→ site{Site₃}

/* Default drop enforced on rest of traffic automatically */
CNZ Coordinator (4): Composition for Conflict detection

Graph-based Composition (ACLs & Dataflow Policies):
1. Normalize policies
2. Graph-based composition
3. Precedence for resolution
Tag-space Assignment:

- Efficiently assign non-overlapping tag space across sites
- **Edge Coloring** to assign non-overlapping tags for Optimizing Tag space reuse

Reuse the same color among other projects across sites:

- **C1**: “Tag space should never overlaps with the tag space assigned to its immediate adjacent sites with which the current site has project association”
- **C2**: Color and Tag size depends on the number of policies enforced by the project.

*Px – Project ID, Cx – Color assigned to a project*
Reuse the same color among other projects across sites:

- **C1**: “Tag space should never overlaps with the tag space assigned to its immediate adjacent sites with which the current site has project association”
- **C2**: Color and Tag size depends on the number of policies enforced by the project.

*Px – Project ID, Cx – Color assigned to a project

Tag Assignment Algorithm for optimizing tag space reuse
CNZ Coordinator (5): Context-aware Inter-site Tagging (cTags)

- Allocates necessary tag space to each project
  - Additional slack tag space for future policies.

- Tag space allocation done globally at the CNZ Coordinator

**Goal:** Maximize efficient reuse of tag space cross-site projects, while avoiding overlaps.
Stateless microservices for data flow protection

- Tag-based filtering
  - vs traditional ACL-based filters
- Tag-based rate limiting
- Tag-based connection tracking
- Preventing DTN hosts from tag spoofing
  - In order to bypass SDN-enforced flow controls
- Preventing malicious exfiltration of sensitive data

Flexibility: Stateless & functional decomposition
Overview: High-level Architecture

Cross-Site CNZ Coordinator (Policy & Data Management Module)

- CNZ Controller
- SDN Controller

- Dynamic Data Flow Policies & Tag assignment
- Process Policy Table
- SciFlow + (GeoIP DB)
- Host DTN

- Flow Records
- Tag-based flow rules to dynamically steer/shunt Traffic

- SciMon
- Grid FTP
- Inline context-aware protection

- SDN Switch
- Border Router
- SDMZ WAN Core

Context from host & network to tag the packet
Recap: CoordiNetZ Capabilities

1. Context-awareness
   - Host-process level context to tag flows
     - SciMon, SciFlow modules

2. Graph-based Policy framework
   - Dataflow level policies
   - Dataflow tracking

3. Graph Composition algorithms
   - Dataflow-based graph composition algorithms

4. Context-aware tagging
   - 20 bits of IPV6 tagging
   - Intra-site tag assignment & Inter-site tag allocation algorithms
   - Optimize the tag assignment and allocation
     - Edge coloring

5. Light weight security stateless microservices
   - Spoofing Protection, Data Exfiltration, Collaborative Protection, On-demand security services with lightweight microservices
Agenda of the Talk

• Background on ESNet & SDMZ
• SDMZ Requirements / Limitations
• CoordiNetZ Architecture
  • Functional Components
• Evaluations
Evaluation: Policies & Dataset for benchmarking

- **PS-1**: From 2 different SDMZ network infrastructures
  - With ~150 & ~400 SDMZ policies
    - ~5325 and ~7987 enforceable rules respectively

- **PS-2**: Large synthetic policy set of 20k policies
  - Derived from PS-1
  - Emulating 40 different SDMZ networks.

- **DS-1**: High Energy Physics - Theory collaboration network dataset
  - Employs ≈9.8k nodes, with ≈25k edges.
Evaluation: Composition

Composition efficiency:

- Composing 20K policies took ~49 sec
- ~30 abstraction trees and ~15% conflicts.
Composition efficiency:

- Reduced composition latency by up to 2.25X
- compared to composition with out caching
Tagging efficiency:

- \( \approx 4 - 5 \times \) fewer bits than bit segmentation and \( \approx 3 - 4 \times \) fewer than Alpaca and FlowTags.

- For Syn Cam Net1: \( 2.2 - 3 \times \) fewer than Alpaca and FlowTags.
Evaluation: Rule Optimization

Rule space efficiency:

- ~40% – 47% rule-space improvement compared to Alpaca, FlowTags and BS.

- For Syn Cam Net1: ~55% rule-space improvement
**Evaluation: Customized Microservices**

**Flow processing Performance (Tag-based filtering):**
- ~8 – 12% throughput improvement with tag-based filtering
  - ~92% (128-bytes) & ~99% (9000-bytes) packets.
  - Vs traditional stateless IPv6 ACL-based filtering

- ~6.6% drop in throughput for Spoofing Protection
  - Vs line rate.

- ~10% Improvement for Tag-based connection tracking
  - Vs flow-based connection tracking.
Summary: CoordiNetZ

- Provides situational-awareness, policy specification and enforcement across SDMZ sites.

CoordiNetZ: Dataflow policy specification and enforcement architecture for SDMZ
Questions?

Feel free to contact Vasudevan Nagendra
vnagendra@cs.stonybrook.edu