

I2NSF Framework Project @ IETF-105 Hackathon

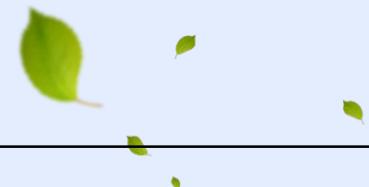


IETF 105, Montreal

July 21, 2019

Champion: Jaehoon Paul Jeong
pauljeong@skku.edu
Sungkyunkwan University

Introduction (1/2)



Goals of IETF-105 I2NSF Hackathon

1. Implementation of the I2NSF Framework for NSF in OpenStack Environment with
 - ✓ Registration Interface via NETCONF/YANG
 - ✓ Consumer-Facing Interface via RESTCONF/YANG
 - ✓ NSF-Facing Interface via NETCONF/YANG
 - ✓ Security Policy Translator in Security Controller
2. Integration of I2NSF Security Controller with ETRI's Public Cloud Control Platform (SoA: Security-on-Air) based on SoA Controller for WYSWYG Network Configuration
3. Application of Commercial Firewall (from Wins) and Open-Source Web Filter (from Suricata)

Introduction (2/2)



Build Environment

1. OS
 - Ubuntu 18.04 LTS
2. ConfD
 - 6.6 Version
3. MySQL
 - 14.14 Version
4. Apache2
 - 2.4.7 Version
5. Django
 - 1.11.14 Version
6. OpenStack
 - Mitaka
7. Suricata
 - 3.2.1 RELEASE
8. Jetconf
 - Python Open API for RESTCONF



I2NSF Internet Drafts for Hackathon

- NSF Capability Data Model
 - ✓ [**draft-ietf-i2nsf-capability-data-model-04**](#)
- Consumer-Facing Interface Data Model
 - ✓ [**draft-ietf-i2nsf-consumer-facing-interface-dm-05**](#)
- NSF-Facing Interface Data Model
 - ✓ [**draft-ietf-i2nsf-nsf-facing-interface-dm-06**](#)
- Registration Interface Data Model
 - ✓ [**draft-ietf-i2nsf-registration-interface-dm-04**](#)
- Security Policy Translation
 - ✓ [**draft-yang-i2nsf-security-policy-translation-03**](#)

I2NSF Hackathon Project Poster

I2NSF (Interface to Network Security Functions) Framework Project

Champions: Jaehoon Paul Jeong (SKKU) and Jong-Hyun Kim (ETRI)



Professor

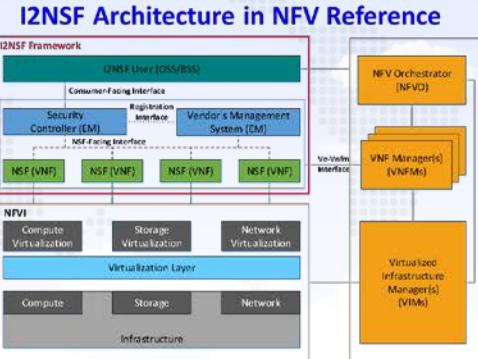
- Jaehoon Paul Jeong (SKKU)

Collaborators

- Jong-Hyun Kim (ETRI)
- Young-Soo Kim (ETRI)
- Jong-Geun Park (ETRI)
- Jung-Tae Kim (ETRI)
- Gu-Min Nam (Wins)

Students

- Jinyong Tim Kim (SKKU)
- Jinhyuk Yang (SKKU)
- Chaehong Chung (SKKU)



ETRI Security on Air Dashboard



Wins Firewall (COTS)



Where to get code

- Github – Source Code
 - ✓ <https://github.com/kimjinyong/i2nsf-framework>

What to pull down to set up an environment

- OS: Ubuntu 18.04 LTS
- ConfD for NETCONF: 6.6 Version
- JetConf for RESTCONF
- Apache2: 2.4.7 Version
- MySQL: 14.14 Version
- Django: 1.11.14 Version
- OpenStack: Mitaka

Manual for Operation Process

- Detailed description about operation process in Manual.txt
(It can be found in Open Source Project folder.)

Contents of Implementation

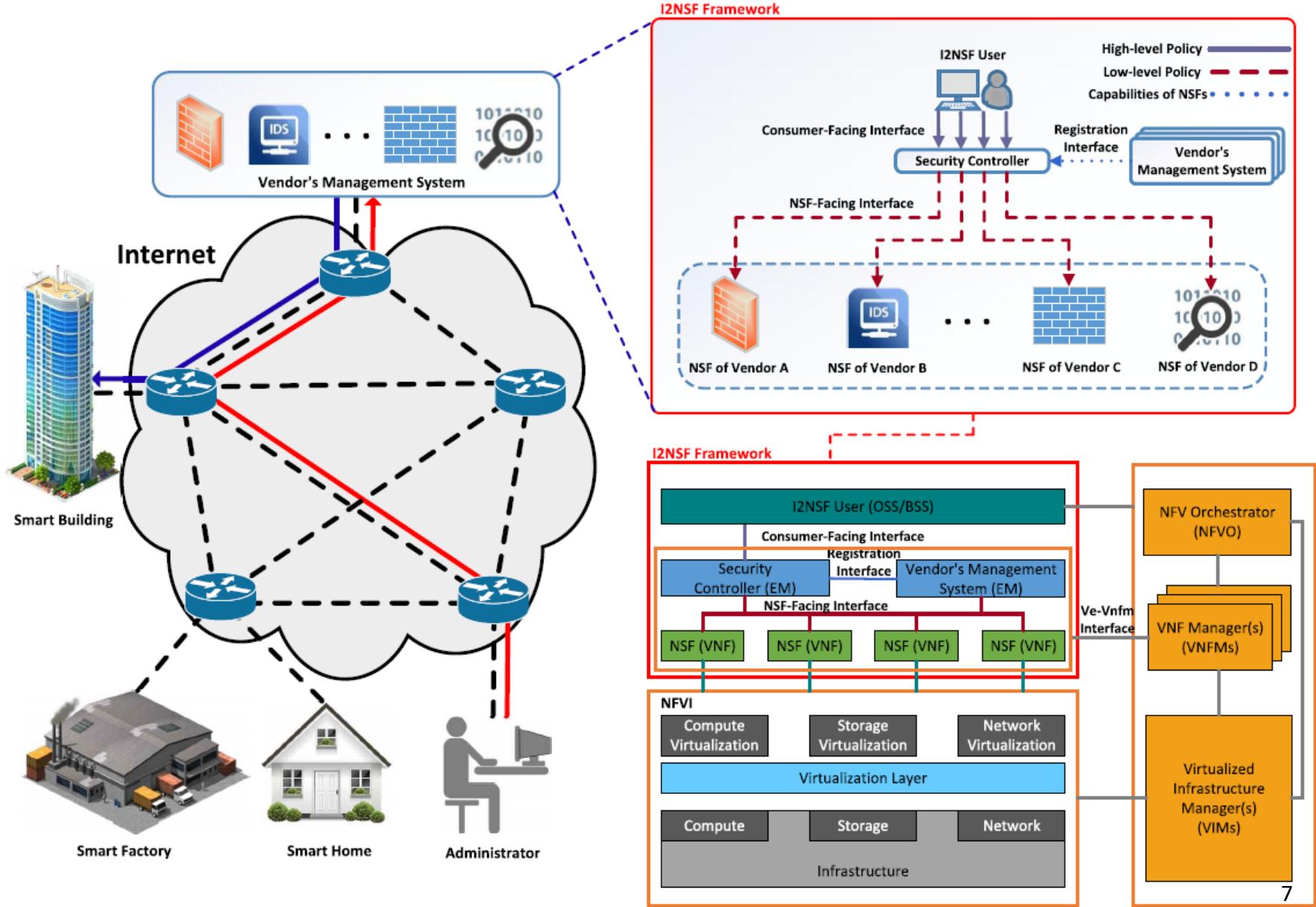
- I2NSF Framework for Network Security Functions (NSFs)
 - ✓ Registration Interface via NETCONF/YANG
 - ✓ NSF-Facing Interface via NETCONF/YANG
 - ✓ I2NSF Framework in OpenStack NFV Environment
 - ✓ NSF Database Management via Consumer-Facing Interface
 - ✓ Interface Data Model Auto-Adoption
- Network Security Functions
 - ✓ Commercial Firewall(Wins) and Web-filter(Suricata)
- Advanced Functions
 - ✓ Security Policy Translation
 - ✓ Application of Wins commercial Firewall for Network Security Function (New Feature)
 - ✓ Integration of Security on Air(SoA) and I2NSF Services (New Feature)



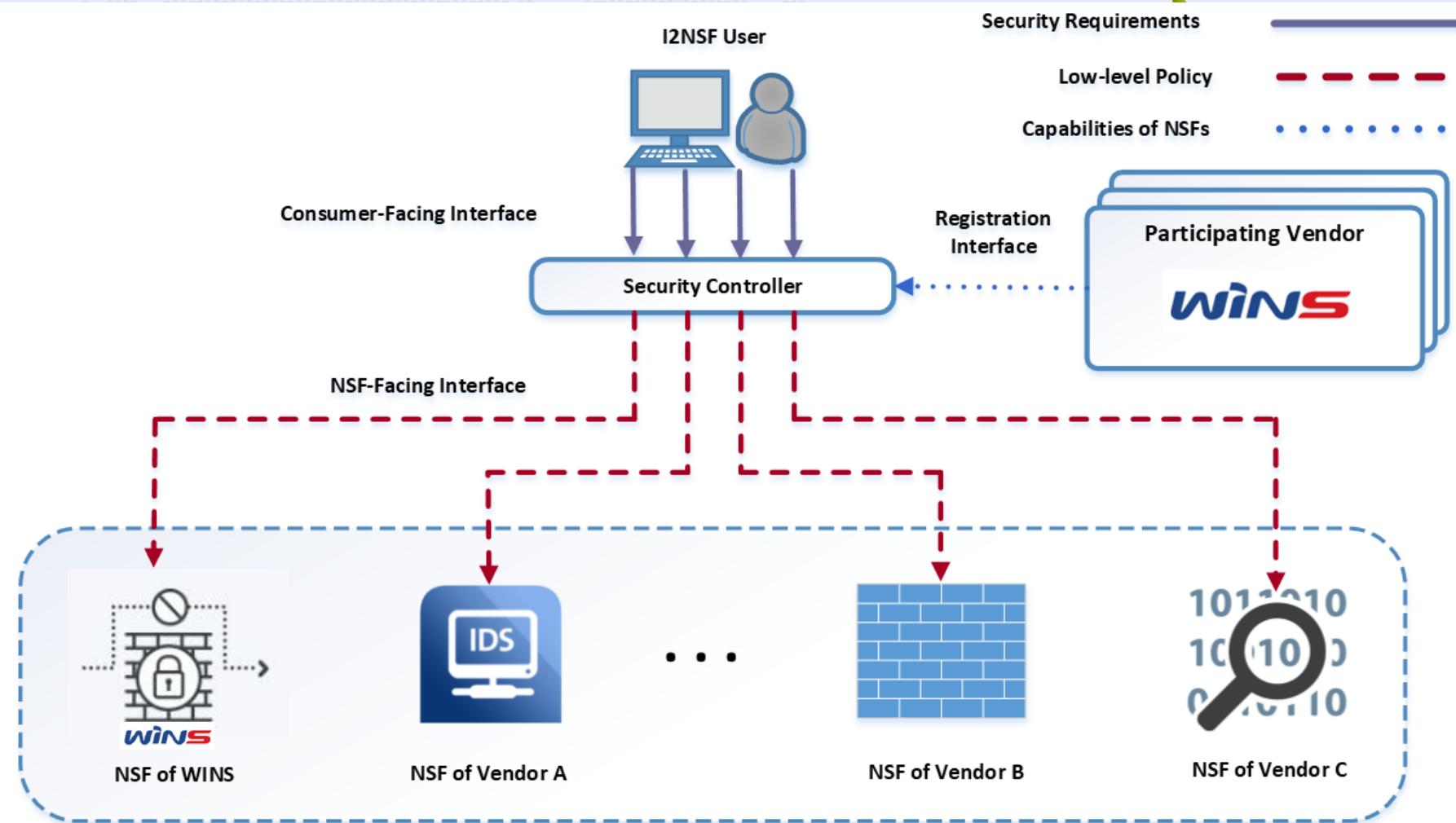
I2NSF Hackathon Project Team



I2NSF System using NSF Framework

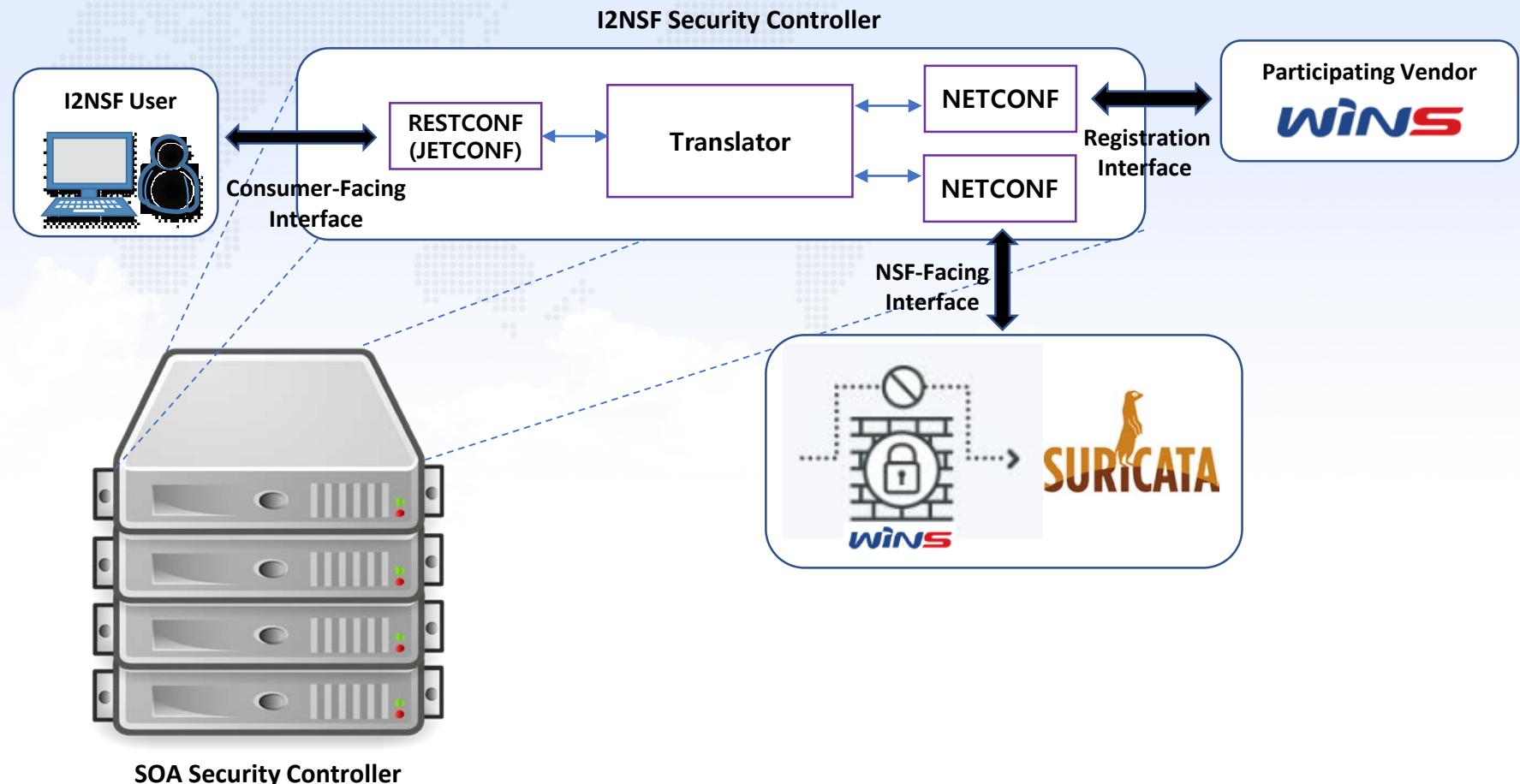


Implementation of I2NSF Hackathon Project (1/2)



1. Application of Commercial Firewall (from Wins) as an NSF

Implementation of I2NSF Hackathon Project (2/2)



2. Integration of I2NSF Security Controller with ETRI's Public Cloud Control Platform (SoA: Security-on-Air)

Hackathon Demonstration (1/5)

- Registration Interface via NETCONF/YANG

The screenshot shows a web-based management interface for security devices. The left sidebar has a dark theme with icons for Service, Security Device (selected), Security Device Registration, Security Devices, Monitoring, User Management, and Cloud Management. The main header says "Security On-AIR" and "Security". A sub-header "Registration Interface" is displayed above a central dialog box titled "Create Security Device". The dialog contains fields for Product Name (vFW), Product Capability (virtual firewall), Software Version (v2.0), Manufacturer (wins), and three logo selection buttons (Product Logo, Manufacturer Logo, Product Capability). Below these is a "Capability Customizing" button. At the bottom are "확인" (Confirm) and "취소" (Cancel) buttons. The status bar at the bottom shows "vFW".

Register the security capabilities: 'VoIP VOLTE filter', 'General Firewall' and etc

Hackathon Demonstration (2/5)

- Consumer-Facing Interface via RESTCONF/YANG

The screenshot shows the ETRI Security On-AIR web interface. The main header reads "Consumer-facing Interface". A central modal window titled "Create Endpoint" is open, containing fields for "Endpoint Type" (set to "User Registration"), "Policy name" (set to "employees2"), "Range of IP address - Begin" (set to "172.16.182.2"), and "Range of IP address - End" (set to "172.16.182.255"). Below the modal are two buttons: "추가" (Add) and "취소" (Cancel). In the background, there's a table with columns "Type" and "Actions" showing entries for "user", "device", and "url", each with a "Resend" button. The left sidebar includes links for Service, Security Device, Monitoring, User Management, and Cloud Management.

Input the corresponding endpoint information
with Policy Name, Range of IP Addresses

Hackathon Demonstration (3/5)

- NSF-Facing Interface via NETCONF/YANG

ETRI

Service

- All Service
- Create Service
- Create Intent Based Service

Services

Security Device

Monitoring

User Management

Cloud Management

Security On-AIR Service

NSF-facing Interface

Service Topology Service Chaining Service

The diagram illustrates a service topology. It starts with a node labeled "SOAC-NET-2" which connects to two intermediate nodes: "Internal-R" and "External-R". "Internal-R" then connects to "Ingress", "Management", and "External". "External" connects to "Suricata". "Suricata" connects to "FW2SURICATA". A "vFW" node is also shown connected to "FW2SURICATA".

English

Edit Service All Service

Virtual Machine

Name vFW

ID 12060fbe-9c74-4cc8-bc3c-29f59ed1d341

Status ACTIVE

Available Zone nova

Date of creation 2019-07-05T06:46:52Z

Date of renewal 2019-07-05T06:47:03Z

Time since creation

Host soac-mitaka6-compute1

Flavor name m1.xlarge

Flavor ID 5

VCPUs 8

RAM 16384(MB)

Disk 160

FW2SURICATA

10.39.0.5/fa:16:3e:a7:a6:f2

Ingress

10.29.0.5/fa:16:3e:a7:a6:f2

When the service topology is created completely, check each instance information

Hackathon Demonstration (4/5)

- Scenario Case 1: Block the access to SNS during office hours

The screenshot shows the ETRI Security On-AIR Service interface. A central modal window titled "Create Rule" is open, displaying the configuration for "Case 1". The rule details are as follows:

Policy name	policy_for_blocking_sns
Source	employees (172.16.181.2 ~ 172.16.181.255)
Destination	sns (www.facebook.com, www.instagram.com)
Time range usage status	* Enable <input type="radio"/> Disable <input type="radio"/>
Start Time	09:00 AM <input type="button" value="Change"/>
End Time	06:00 PM <input type="button" value="Change"/>
Action Type	Drop

At the bottom of the modal, there are two buttons: "추가" (Add) and "취소" (Cancel). The background shows a blurred view of the rule list and other service management options.

Blocking accesses the SNS during office hours using the Suricata's URL filter

Hackathon Demonstration (5/5)

- Scenario Case 2: Block the access to all the websites

The screenshot shows the 'Security On-AIR Service' interface with a blue header bar labeled 'Case 2'. A modal window titled 'Create Rule' is open, containing fields for Policy name (off_time_all_block), Source (employees2 (172.16.182.2 ~ 172.16.182.255)), Destination (all_deny (0.0.0.0 ~ 255.255.255.255)), Time range usage status (Enable), and Action Type (Drop). Below the modal, the main interface shows a table for 'Source' with columns for Name, Type, and IP Range, and a table for 'Destination' with columns for Name, Type, and URL. At the bottom of the interface, there is a banner with the text 'Blocking All access using Wins' Firewall' and a list of URLs: www.facebook.com, www.instagram.com.

Case 2

Create Rule

Policy name	off_time_all_block
Source	employees2 (172.16.182.2 ~ 172.16.182.255)
Destination	all_deny (0.0.0.0 ~ 255.255.255.255)
Time range usage status	<input checked="" type="radio"/> Enable <input type="radio"/> Disable
Action Type	Drop

Source

Name	employees
Type	user
IP Range	172.16.181.2 ~ 172.16.181.255

Destination

Name	sns
Type	
URL	

Blocking All access using Wins' Firewall

www.facebook.com, www.instagram.com



- **Proof of Concept (POC) of I2NSF Framework**
 - I2NSF Framework on NFV Framework
 - I2NSF Interfaces (Consumer-Facing, NSF-Facing, and Registration Interfaces)
 - I2NSF Security Policy Translator

- **Integration of I2NSF to Commercial Platform**
 - Application of a Commercial Vendor's NSF (e.g., Wins Firewall)
 - Integration of I2NSF Security Controller into a Commercial Security Cloud Platform (called SOA)

Information of I2NSF Hackathon Project (1/2)

YouTube for Video Demonstration

- <https://www.youtube.com/watch?v=jD4ndqzN0is>



Demonstration of I2NSF Framework with Security on Air

Information of I2NSF Hackathon Project (2/2)

GitHub for I2NSF Hackathon Source Code

- <https://github.com/kimjinyong/i2nsf-framework>

kimjinyong / i2nsf-framework

Code Issues 0 Pull requests 0 Projects 0 Security Insights

Join GitHub today

GitHub is home to over 36 million developers working together to host and review code, manage projects, and build software together.

Dismiss

Sign up

Hackathon-104

26 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find File Clone or download

Author	Commit Message	Latest commit
kimjinyong	qwe	1a411b0 3 days ago
	Hackathon-104	Delete test.txt 5 days ago
	Hackathon-105	Add files via upload 5 days ago
dms	Source Code	last month
kubernetes	Source Code	last month
security_controller	Source Code	last month