



AEGIS : An Automated Permission Generation and Verification System for SDNs

ACM SIGCOMM 2018 Workshop on SecSoN

Heedo Kang, Seungwon Shin, Vinod Yegneswaran*, Shalini Ghosh*, Phillip Porras*

KAIST, SRI International*

Contents

1. Background

2. Motivation & Challenge

3. AEGIS Design

- Static Engine
- Dynamic Engine

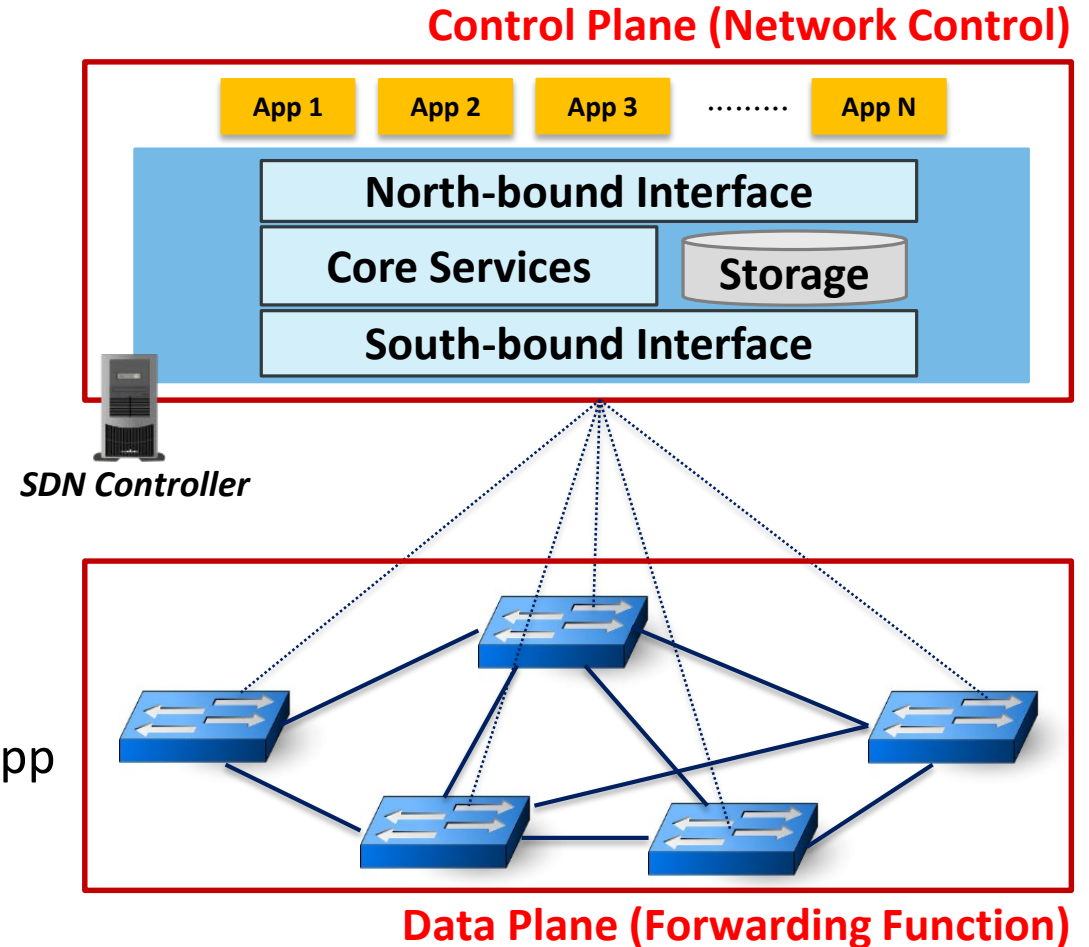
4. Evaluation

5. Conclusion

Background

Software Defined Networking(SDN)?

- Network decoupling
 - Network control and forwarding functions
- Programmable network
 - Flexible and dynamic network control
 - Innovative network service
- Potential abuse
 - SDN controller API can be abused by SDN app
 - Entire resources can be manipulated



Background

Abusing SDN controller API

```
-----  
LINK INFO FROM DB : Count = 1  
-----  
LINK_TABLE_NAME = controller_link  
LINK_ID          = 00:00:00:00:00:00:01-2-00:00:00:00:00:00:02-2  
LINK_SRC_SWITCH  = 00:00:00:00:00:00:01  
LINK_SRC_PORT    = 2  
LINK_SRC_PORT_STATE = 0  
LINK_DST_SWITCH  = 00:00:00:00:00:00:02  
LINK_DST_PORT    = 2  
LINK_DST_PORT_STATE = 0  
LINK_VALID_TIME  = 1390964347029  
LINK_TYPE        = internal  
-----
```

Link 1

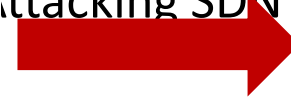
```
-----  
LINK INFO FROM DB : Count = 2  
-----  
LINK_TABLE_NAME = controller_link  
LINK_ID          = 00:00:00:00:00:00:02-2-00:00:00:00:00:00:01-2  
LINK_SRC_SWITCH  = 00:00:00:00:00:00:02  
LINK_SRC_PORT    = 2  
LINK_SRC_PORT_STATE = 0  
LINK_DST_SWITCH  = 00:00:00:00:00:00:01  
LINK_DST_PORT    = 2  
LINK_DST_PORT_STATE = 0  
LINK_VALID_TIME  = 1390964347026  
LINK_TYPE        = internal  
-----
```

Link 2

Manhee Lee, Seungwon Shin, Vinod Yegneswaran, Phillip Porras,
Network for Social

Attacking SDN II

and Seungwon Shin
Environment", SDN-INFOV SECURITY 2010.



```
>W - [ATTACK]-----  
>W - [ATTACK] LINK INFO FROM DB : Count = 1  
>W - [ATTACK]-----  
>W - [ATTACK] LINK_TABLE_NAME      = controller_link  
>W - [ATTACK] LINK_ID              = 00:00:00:00:00:00:02-2-00:00:00:00:00:00:01-2  
>W - [ATTACK] LINK_SRC_SWITCH      = 00:00:00:00:00:00:02  
>W - [ATTACK] LINK_SRC_PORT        = 2  
>W - [ATTACK] LINK_SRC_PORT_STATE  = 0  
>W - [ATTACK] LINK_DST_SWITCH      = 00:00:00:00:00:00:01  
>W - [ATTACK] LINK_DST_PORT        = 2  
>W - [ATTACK] LINK_DST_PORT_STATE  = 0  
>W - [ATTACK] LINK_VALID_TIME      = 1390964347026  
>W - [ATTACK] LINK_TYPE            = internal  
>W - [ATTACK]-----  
>W - [ATTACK] Access InternalDB : delete Link Information
```

Link 2 Only
Link 1 has been deleted

- Shin, Seungwon, et al. "Rosemary: A robust, secure, and high-performance network operating system." CCS 2014.

Background

Existing SDN permission systems

- SE-Floodlight
 - Porras, Phillip A., et al. "Securing the Software Defined Network Control Layer." NDSS 2015.
 - Role based access control (for only Data-Plane related resources)
- SDNShield
 - Wen, Xitao, et al. "SDNShield: Reconciling Configurable Application Permissions for SDN App Markets." DSN 2016.
 - Permission & policy based access control (for only Data-Plane related resources)
- Security-Mode ONOS
 - Changhoon Yoon, et al. "A Security-Mode for Carrier-Grade SDN Controllers", ACSAC 2017.
 - Permission based access control (for all resources)

Contents

1. Background

2. Motivation & Challenge

3. AEGIS Design

- Static Engine
- Dynamic Engine

4. Evaluation

5. Conclusion

Motivation

1. Automation deficiency

- To build SDN permission system..

(i) Analyze what resources(assets) should be protected

(ii) Inspect what resources are accessed by each APIs

(iii) Design permission model

(iv) Implement permission system

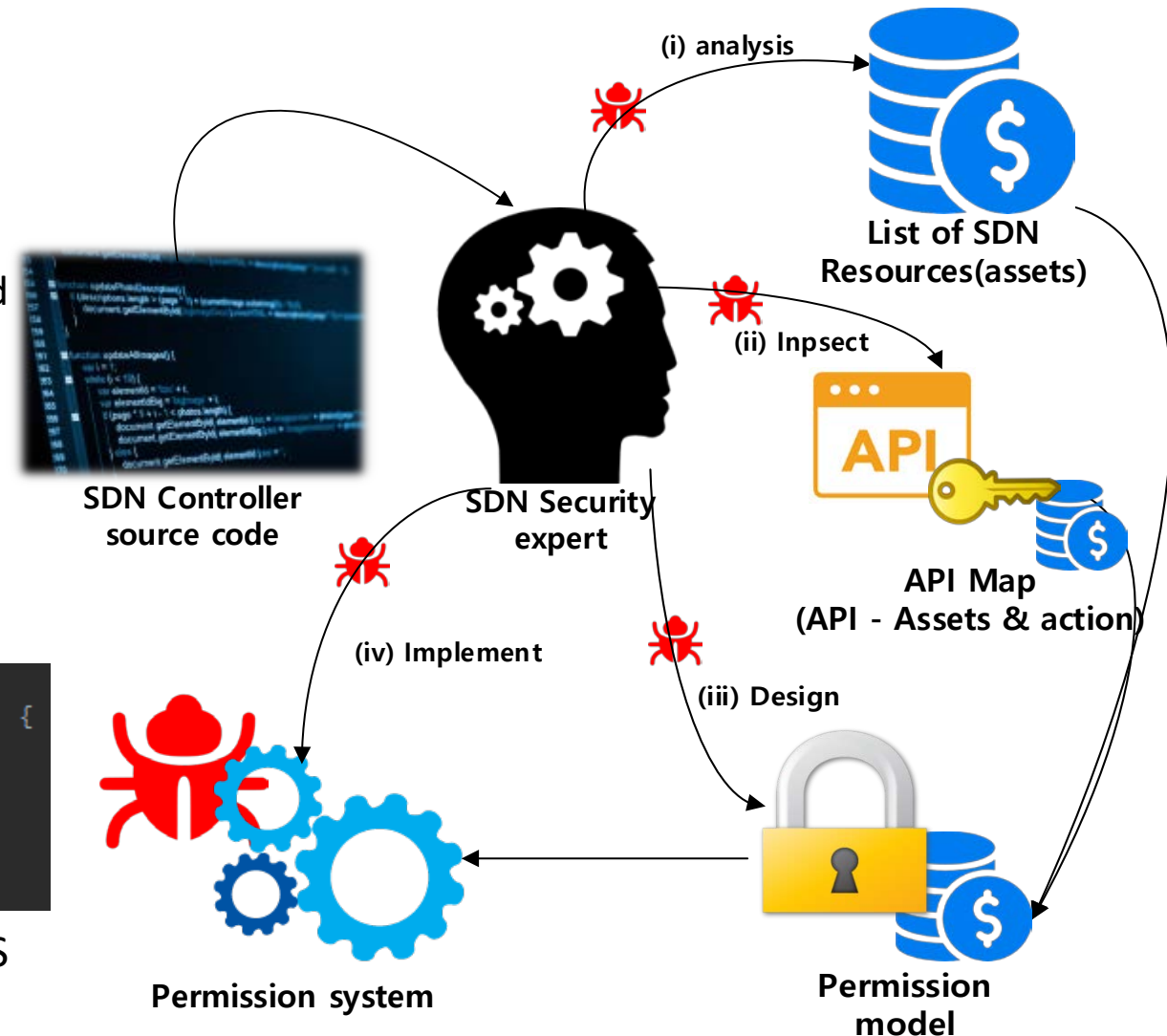
Human errors!

```
@Override
public void registerDeactivateHook(ApplicationId appId, Runnable hook) {
    checkPermission(AppId, appId);
    checkNotNull(appId, APP_ID_NULL);
    checkNotNull(hook, "Hook cannot be null");
    deactivateHooks.put(appId.name(), hook);
}
```

→ APP_WRITE should be checked!

→ This is WRITE action!

Example of **human error** existed in Security-Mode ONOS



Motivation

2. Portability deficiency

- Procedure for building SDN permission system
 - Too complicated task
 - Error prone
- Existing SDN permission systems
 - Tightly coupled with SDN controller implementation
 - e.g) SE-Floodlight (Floodlight), Security-Mode ONOS (ONOS)
 - Cannot be ported to any other controller

Motivation

3. Flexibility deficiency

- Different security requirements



Bob
(Network operator)

Our network needs **fine-grained access control over only topology resource.**



Alice
(Network operator)

Our network needs **fine-grained access control over all resources**

- Existing SDN permission systems
 - Permission model is fixed

Challenges

- **Ultimate goal**
 - Suggest new automated permission generation and verification system for SDN
- **Summary of challenges**
 - **Automation**
 - Automatically generate permission model for SDN controller
 - **Portability**
 - Independently designed and implemented from specific SDN controller implementation
 - **Flexibility**
 - Provide way to flexibly generate permission model

Contents

1. Background

2. Motivation & Challenge

3. AEGIS Design

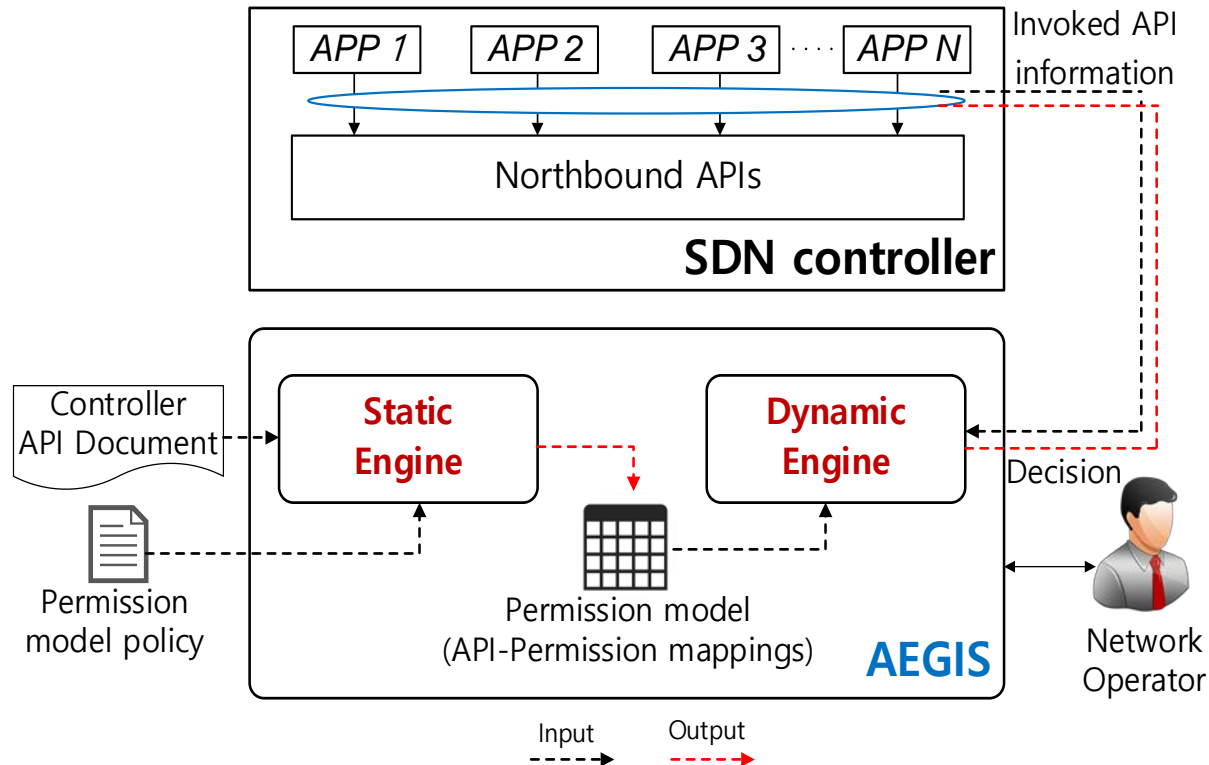
- Static Engine
- Dynamic Engine

4. Evaluation

5. Conclusion

AEGIS Design

- Overview



- **Static Engine** (execute before run-time)
 - Automatically generates permission model
 - Various NLP techniques
- **Dynamic Engine** (execute on run-time)
 - Verifies if application has right permissions to execute API
 - Hooking & Code injection technique

AEGIS Design

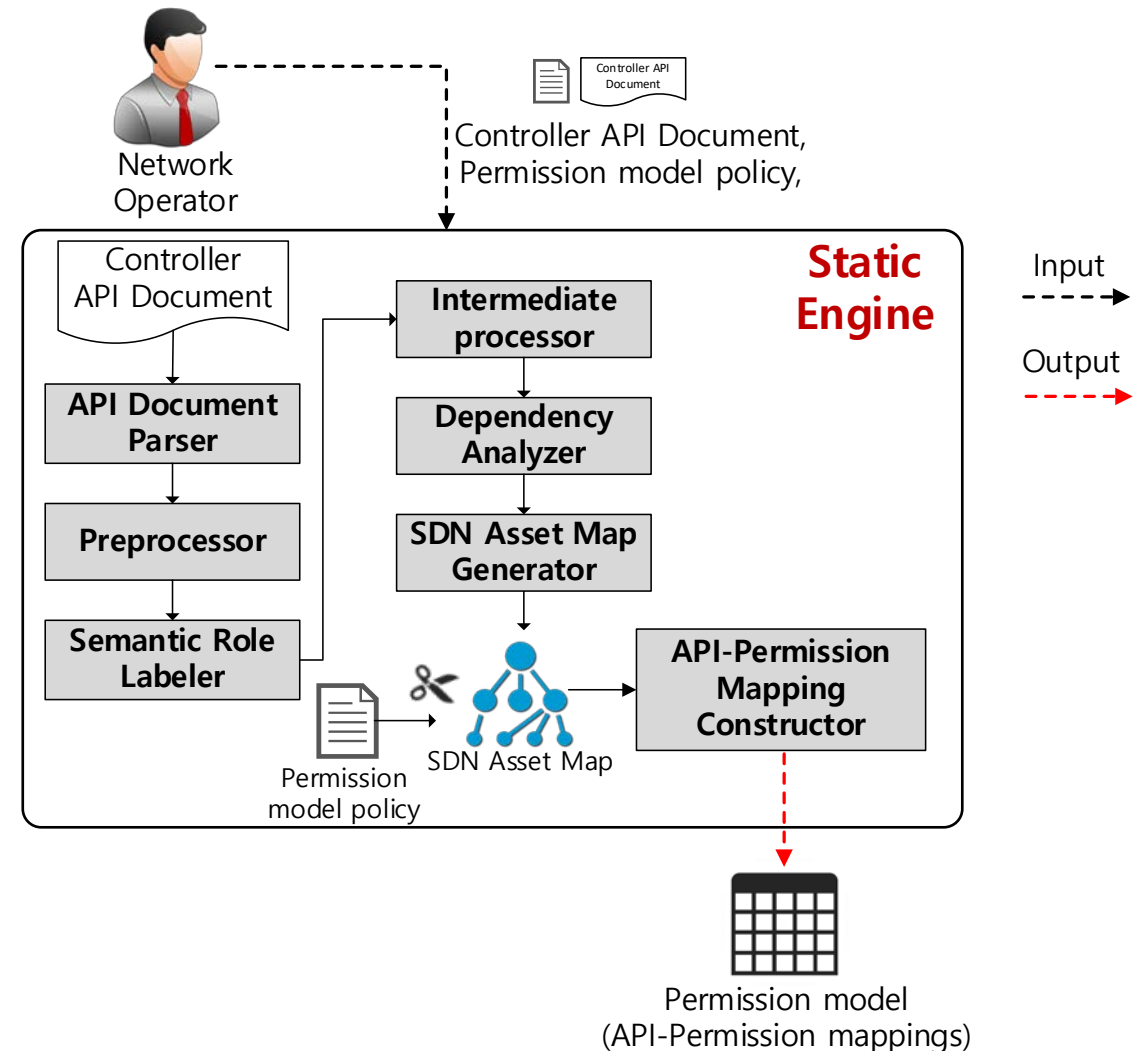
• Static Engine

• Consists of seven modules

- API Document Parser
- Preprocessor
- Semantic Role Labeler
- Intermediate processor
- Dependency Analyzer
- SDN Asset Map Generator
- API-Permission Mapping Constructor

• Takes controller API document & permission model policy as inputs

• Generates permission model as output



AEGIS Design

- API document Parser

- Extract following features from API document

- Package path

- Class name

- API name

- API description

```
org.onosproject.cluster
Interface ClusterAdminService

All Superinterfaces:
ClusterService, ListenerService<ClusterEvent, ClusterEventListener>
```

ONOS controller API document

```
public interface ClusterAdminService
extends ClusterService

Service for administering the cluster node membership.
```

Method Summary

All Methods Instance Methods Abstract Methods

Modifier and Type	Method and Description
void	addNode(NodeId nodeId, Ipaddress ip, int tcpPort) Adds a new controller node to the cluster.
void	formCluster(Set<ControllerNode> nodes) Forms cluster configuration based on the specified set of node information. This method resets and restarts the controller instance.
void	formCluster(Set<ControllerNode> nodes, int partitionSize) Forms cluster configuration based on the specified set of node information. This method resets and restarts the controller instance.
void	markFullyStarted(boolean started) Marks the current node as fully started or not.
void	removeNode(NodeId nodeId) Removes the specified node from the cluster node list.



SDN controller
API document

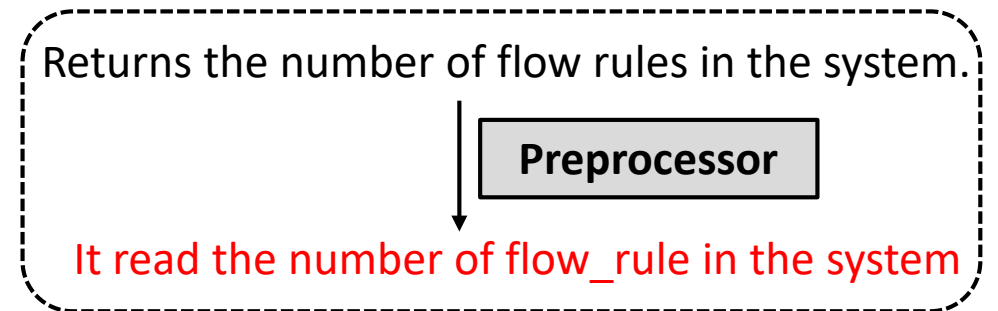
API document
Parser

API = org.onosproject.net.flow.FlowRuleService.getFlowRuleCount
Description = Returns the number of flow rules in the system.

AEGIS Design

- Preprocessor

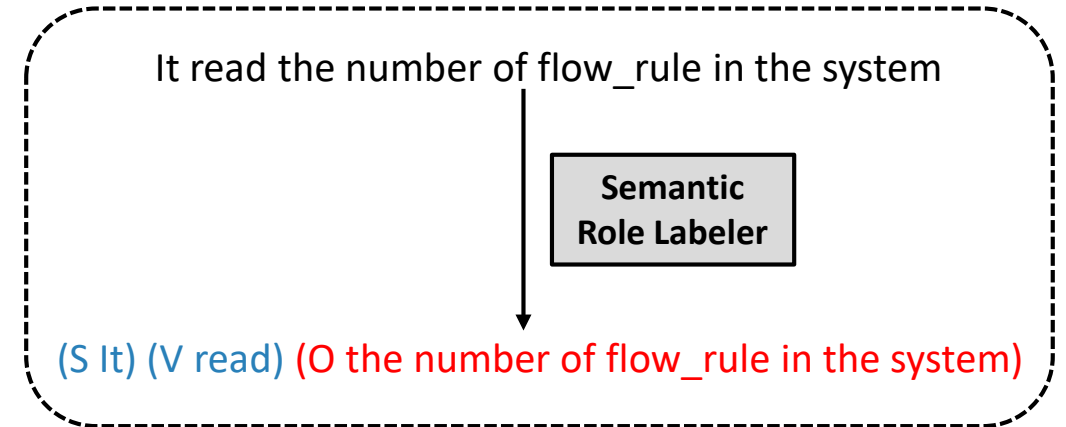
- Replace all uppercase letters with lowercase letters
- Remove special characters
- Inject fake subject
- Converge entity n-grams into one word
- Change verb into three types of action word
 - e.g)
 - obtain, fetch, get, find, check -> **read**
 - Send, create, remove, add, unregister-> **write**
 - Invoke, activate, stop, perform.....-> **execute**



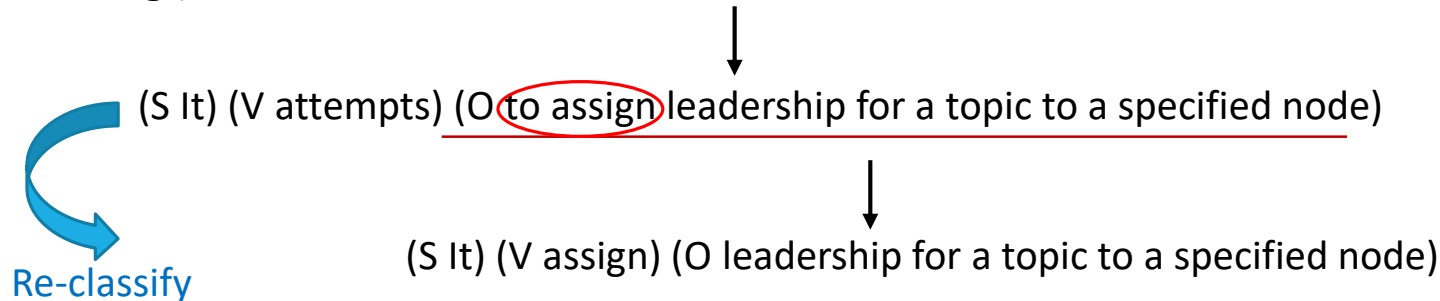
AEGIS Design

- Semantic Role Labeler

- Classifies description into semantic constituents
 - Object contains resources that API access
- Investigates classified object
 - ✓ Starts with to-infinitive or gerund?
 - Re-classifies object sentence

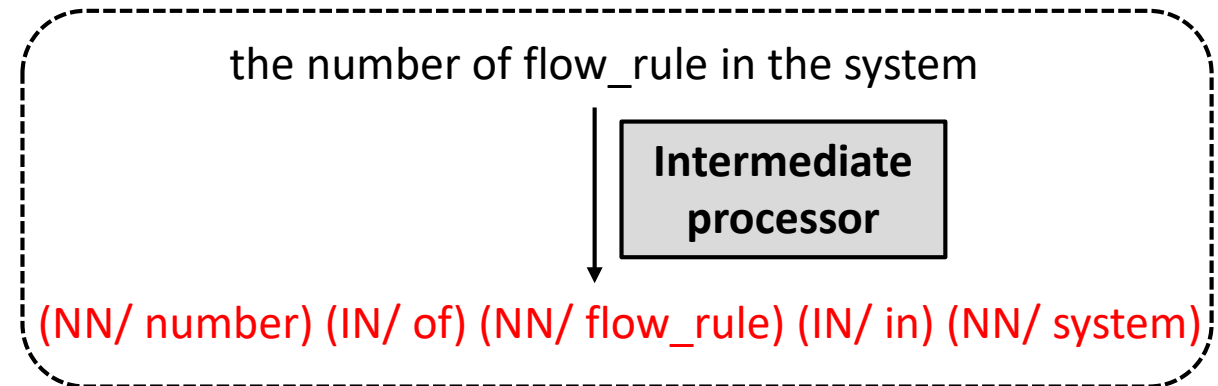


eg.) It attempts to assign leadership for a topic to a specified node



AEGIS Design

- Intermediate processor
 - Tags Part of speeches(POS)
 - e.g.) ~ (NN/flow_rule) (IN/in) ~
 - Removes determiner words
 - e.g.) ~~the~~ number of ~
 - Converts word to stem of the word
 - e.g.) ~ devices~~s~~



AEGIS Design

- Dependency Analyzer

- Analyzes relationships between each word

- Dependency parsing

root (Root-0, number-1)

case(flow_rule-3, of-2)

Example: nmod:of(number-1, flow_rule-3)

case(system-5, in-4)

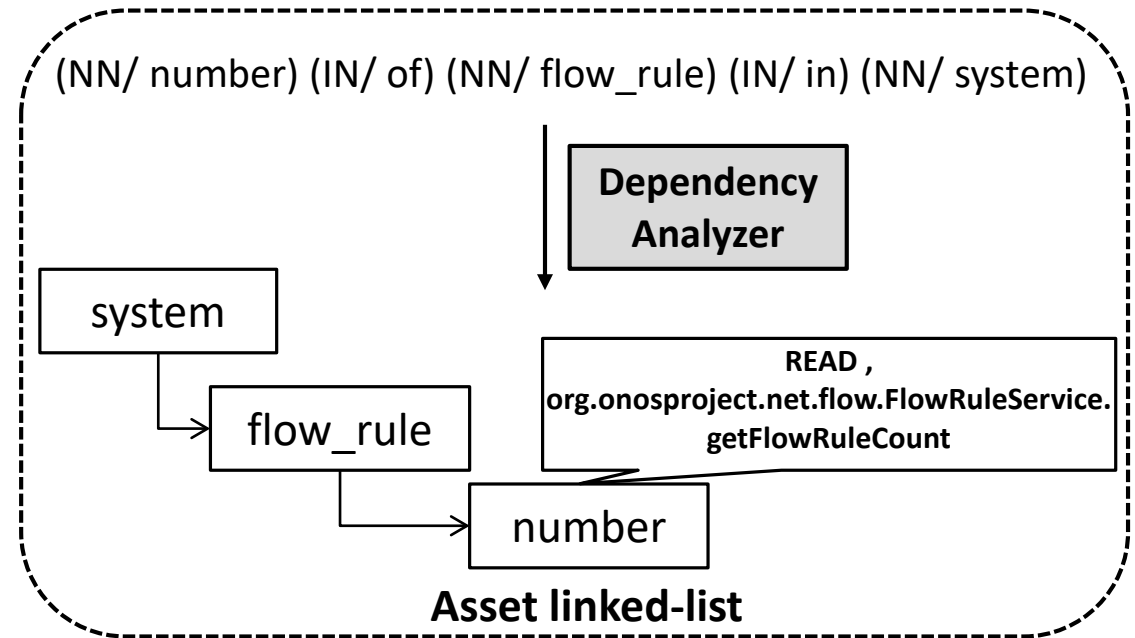
nmod:in(number-1, system-5)

- Extract set of nominal modifier(nmod) relation

- Generates asset-linked list

- Based on predefined rules

- Tag API path & action



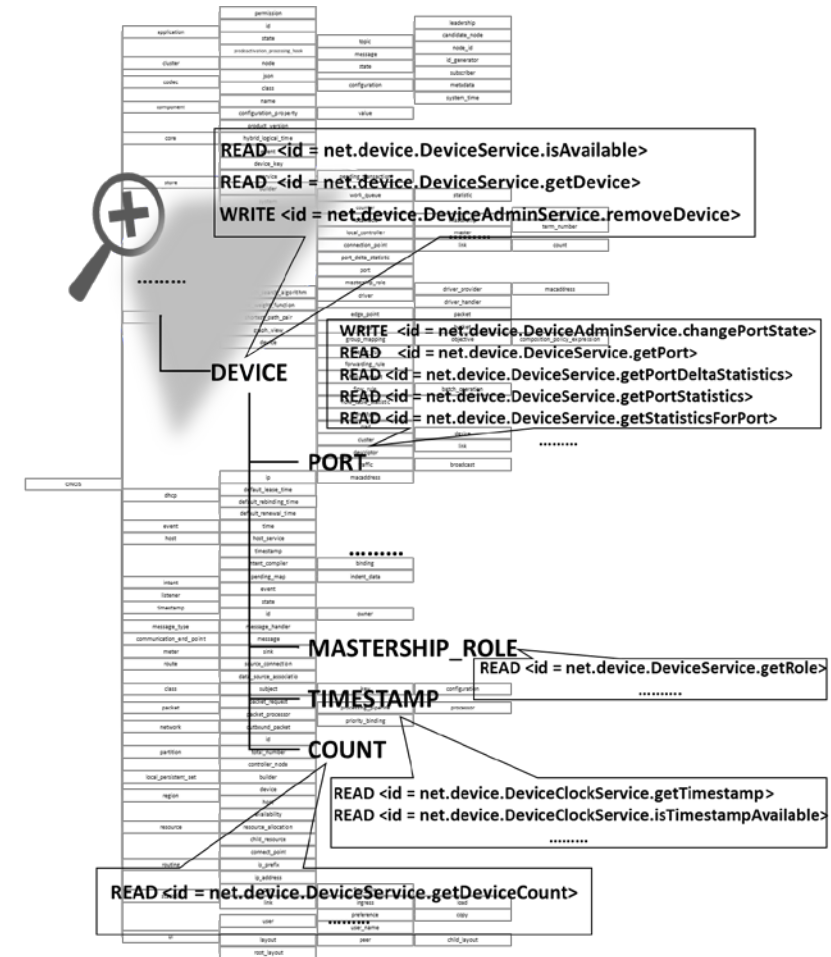
AEGIS Design

- SDN Asset Map Generator

- Integrates all asset-linked list
- Flexible permission model generation
 - Pruning map based on permission model policy
 - e.g) Remove STATSTIC node and move tags to PORT node

- API-permission Mapping Constructor

- Creates permission type
- By concatenating node name from each starting node to root node and action word
- Maps each generated permission type to API path



ONOS Asset map

AEGIS Design

- Permission model

AEGIS permission token	Accessible API
DEVICE_WRITE	DeviceAdminService.removeDevice(DeviceId)
DEVICE_PORT_WRITE	DeviceAdminService.changePortState(DeviceId, PortNumber, boolean)
DEVICE_READ	DeviceService.isAvailable(DeviceId) DeviceService.getDevices(Device.Type) DeviceService.getDevices() DeviceService.getDevice(DeviceId) DeviceService.getAvailableDevices(Device.Type) DeviceService.getAvailableDevices()
DEVICE_PORT_READ	DeviceService.getPort(ConnectPoint) DeviceService.getPort(DeviceId, PortNumber) DeviceService.getPorts(DeviceId)
DEVICE_PORT_STATSTIC_READ	DeviceService.getPortDeltaStatistics(DeviceId) DeviceService.getDeltaStatisticsForPort(DeviceId, PortNumber) DeviceService.getPortStatistics(DeviceId) DeviceService.getStatisticsForPort(DeviceId, PortNumber)
DEVICE_TIMESTAMP_READ	DeviceClockService.getTimestamp(DeviceId) DeviceClockService.isTimestampAvailable(DeviceId)
DEVICE_COUNT_READ	DeviceService.getDeviceCount()
DEVICE_MASTERSHIP_ROLE_READ	DeviceService.getRole(DeviceId)

Example of ONOS API – permission mappings

AEGIS Design

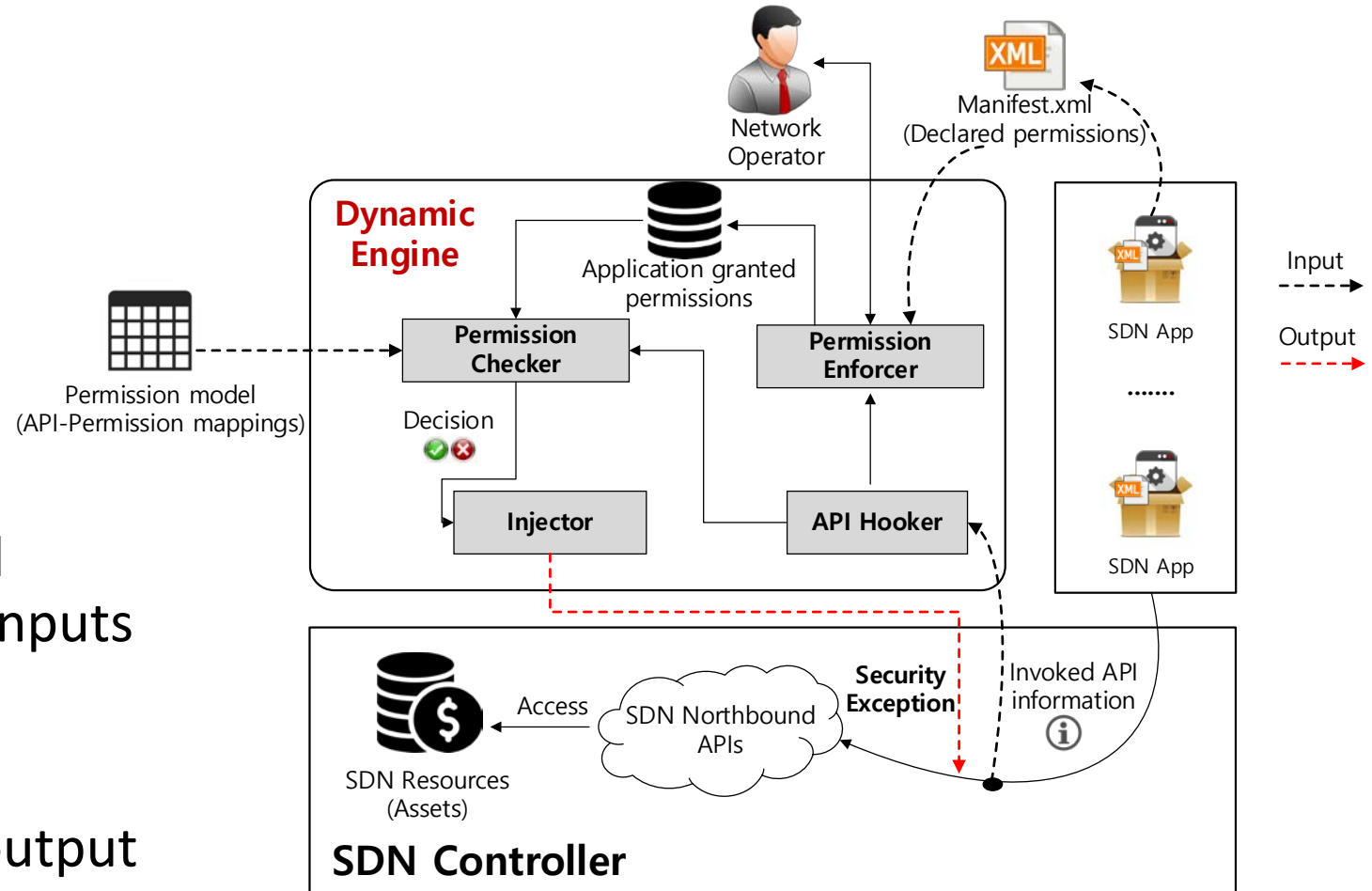
- Dynamic Engine

- Consists of four modules

- API Hooker
- Permission Enforcer
- Permission Checker
- Injector

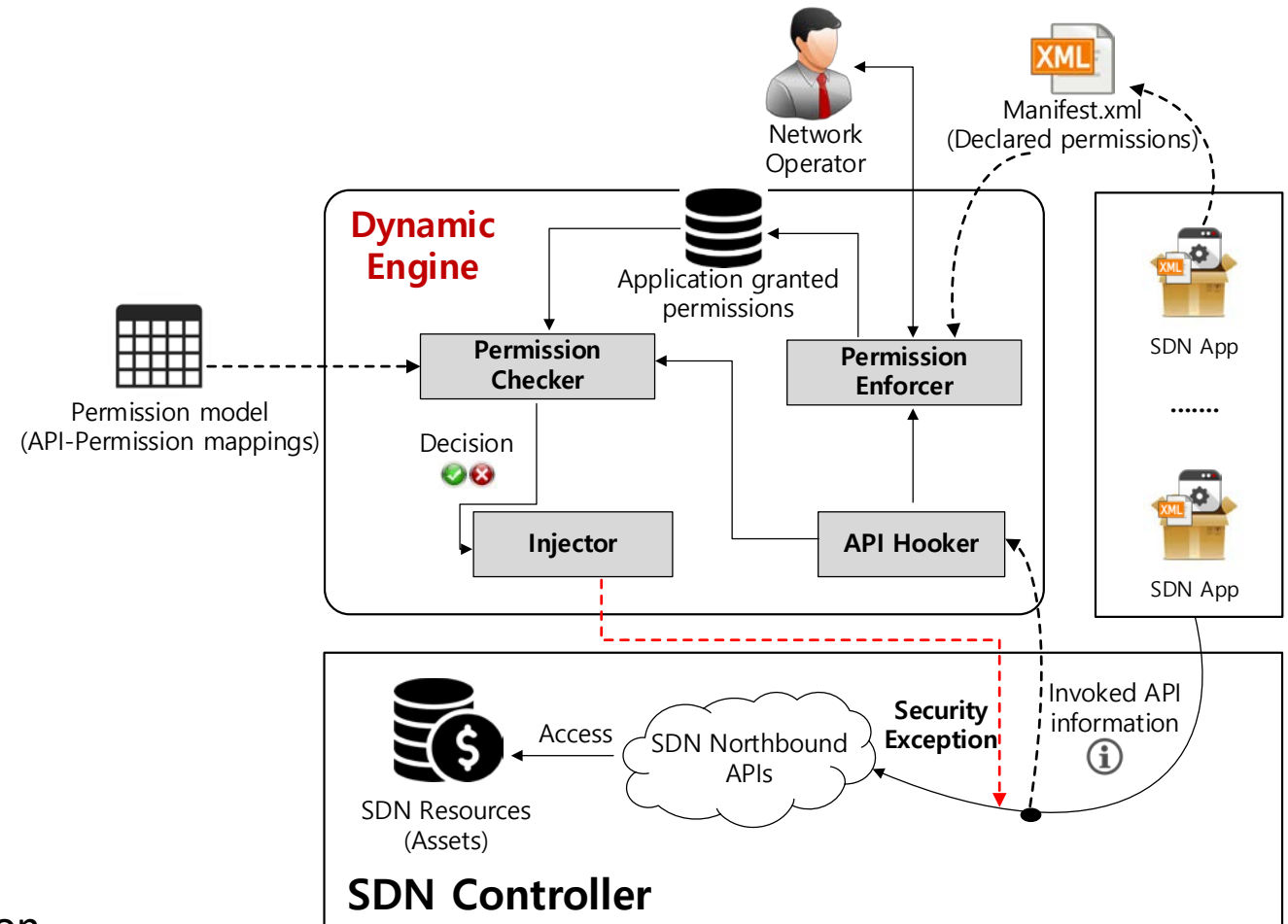
- Takes permission model and invoked API information as inputs

- Generates and injects security exception code as output



AEGIS Design

- API Hooker
 - Sniffs all of Northbound-API calls
 - By using hooking technique
- Permission Enforcer
 - Enforces permission reviewing process
 - Grant(store) declared permissions
- Permission checker
 - Checks if application has right permissions
 - Makes decision
- Injector
 - Injects code that generates security exception
 - By using code injection technique



Contents

1. Background

2. Motivation & Challenge

3. AEGIS Design

- Static Engine
- Dynamic Engine

4. Evaluation

5. Conclusion

Evaluation

- Completeness

- How many SDN API descriptions can be covered by AEGIS?

Controller	# of total APIs	# of covered APIs	Coverage
ONOS	355	348	98%
Floodlight	198	186	94%
POX	14	14	100%
Total	567	548	96.6%

- Failure case examples

- A **builder** for the creation of local persistent maps backed by disk
- Removes all links **between between** the specified src and dst connection point

Evaluation

- Soundness

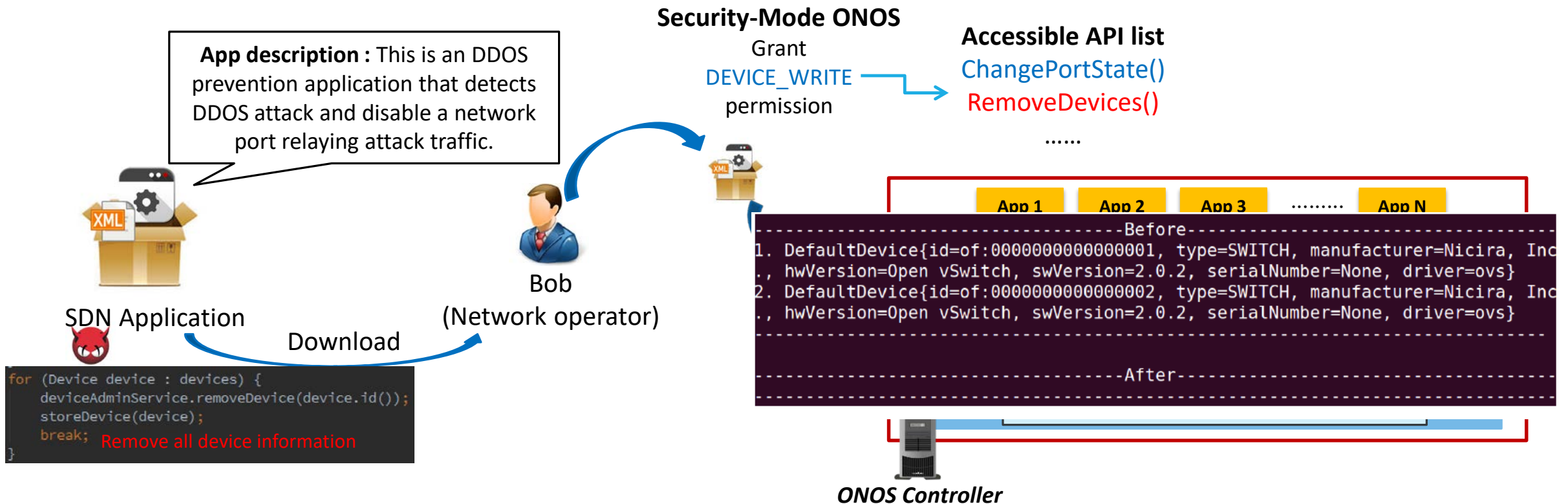
- How much accurately does AEGIS extract?
 - No ground truth
 - Survey of 20 SDN experts
 - Randomly select 30 Northbound-API descriptions from ONOS, Floodlight and POX controller

Question	# of positive responses	# of negative responses	Correctness
Action word & resources	583	17	97.2%
Relation	574	23	95.7%

Evaluation

- Use case

- Can AEGIS invalidate attack scenario that is valid on existing permission system?



Evaluation

- Use case

Security-Mode ONOS permission token		AEGIS permission token	Accessible API
DEVICE_WRITE		DEVICE_WRITE	DeviceAdminService.removeDevice(DeviceId)
		DEVICE_PORT_WRITE	DeviceAdminService.changePortState(DeviceId, PortNumber, boolean)
DEVICE	DEVICE_READ		DeviceService.getDevice(DeviceId) DeviceService.getAvailableDevices(Device.Type) DeviceService.getAvailableDevices()
		DEVICE_PORT_READ	DeviceService.getPort(ConnectPoint) DeviceService.getPort(DeviceId, PortNumber) DeviceService.getPorts(DeviceId)
		DEVICE_PORT_STATSTIC_READ	DeviceService.getPortDeltaStatistics(DeviceId) DeviceService.getDeltaStatisticsForPort(DeviceId, PortNumber) DeviceService.getPortStatistics(DeviceId) DeviceService.getStatisticsForPort(DeviceId, PortNumber)
		DEVICE_TIMESTAMP_READ	DeviceClockService.getTimestamp(DeviceId) DeviceClockService.isTimestampAvailable(DeviceId)
		DEVICE_COUNT_READ	DeviceService.getDeviceCount()
		DEVICE_MASTERSHIP_ROLE_READ	DeviceService.getRole(DeviceId)

```

2017-04-22 00:19:03,568 | ERROR | 1 for user karaf | onos-app-attack
| 180 - org.onosproject.onos-app-attack - 1.5.0 | [org.onosproject.attack.
AppComponent(128)] The activate method has thrown an exception
java.lang.SecurityException
    at org.onosproject.net.device.impl.DeviceManager.removeDevice(DeviceMana
ger.java:246)
    at org.onosproject.attack.AppComponent.attack(AppComponent.java:76)
    at org.onosproject.attack.AppComponent.activate(AppComponent.java:58)
    
```

Accessible DEVICE resource related APIs of ONOS with each permission token in **Security-Mode ONOS** and **AEGIS**

Attack scenario result with **AEGIS**

Conclusion

- Address some deficiencies of existing SDN permission system
- Propose AEGIS
 - Automatically and flexibly generates SDN permission model
 - Verifies permissions of SDN app in separated process from SDN controller
- Implement prototype
- Evaluate its completeness and soundness & demonstrate its usecase

Q & A