教育部補助5G行動寬頻跨校教學聯盟

下世代Network Slicing模組設計

課程單元： Security of Network Slicing

國立中山大學 資訊工程系
授課教師：李宗南教授
教材編撰：梁鼎忠

目錄
CONTENTS

01 課程目標

02 Terminology

03 Network Slicing Security

04 Security of SDN

05 Conclusions

01 課程目標

課程目標

- Summarizing the different threats in Network Slicing
- Summarizing the different threats in Software-Defined Networking
- Simply introduce the Concerns of in Network Slicing
- Simply introduce the characteristics of threats in Software-Defined Networking

Introduction

- Security challenges for Software-Defined Networks differ in some respects from those of a classical network due to the specific network implementation and SDN’s inherent control and programmability characteristics
- Logically centralized control may expose a series of high-value assets to attackers while the ability to directly access the control plane results in a new attack surface (i.e. the Application-Control Programming Interface (ACPI)) for adversaries

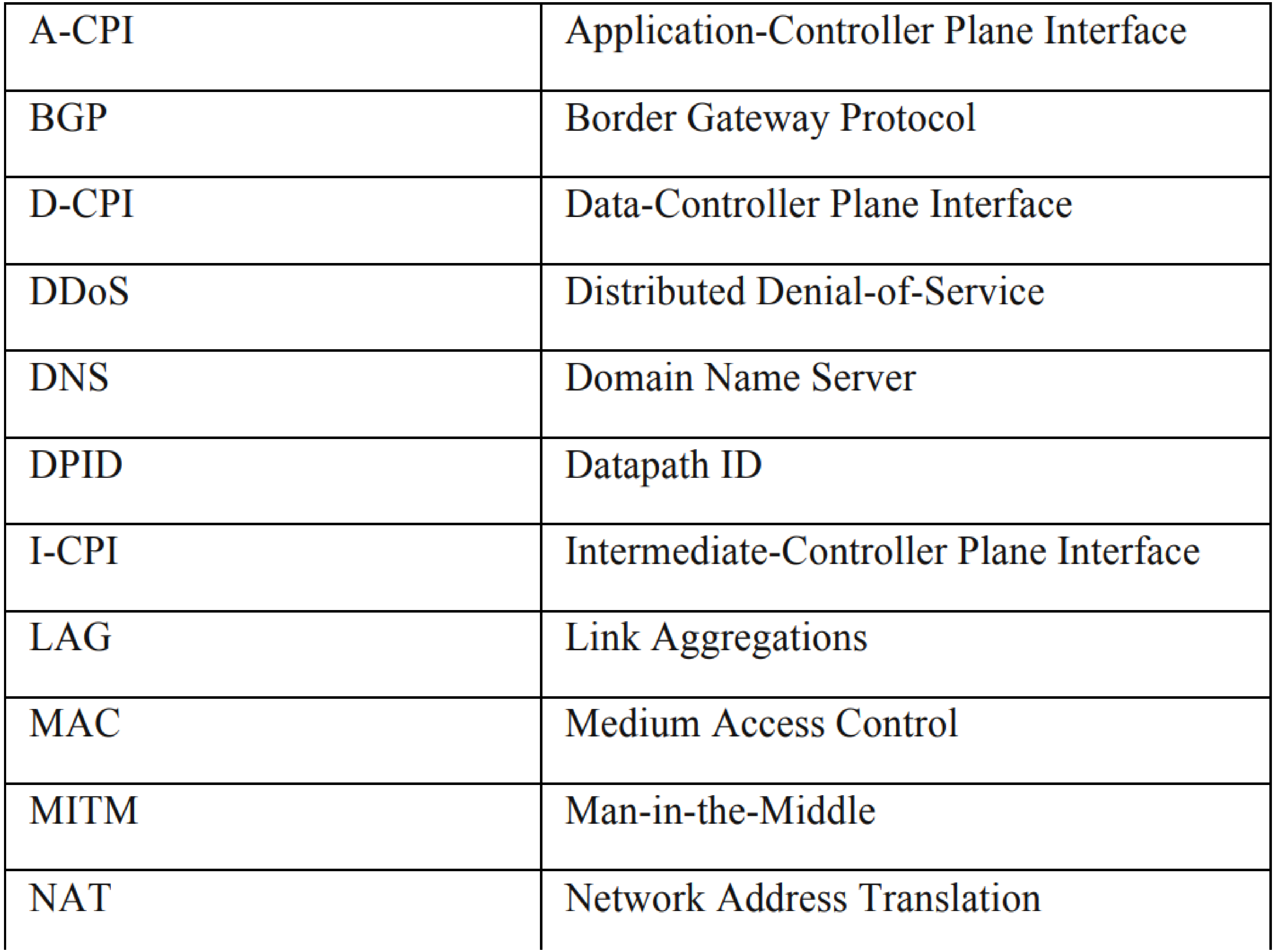
02

Terminology

Definitions

- Availability:
  - The readiness for providing correct service to authorized parties
- Confidentiality:
  - Limiting information access and disclosure to authorized parties
- Integrity:
  - The trustworthiness of information resources
- Reference Data:
  - The data objects that are related to state, configuration or status that are used by the logic of a security control
- Trust Boundary:
  - The boundary of an area between components where the privilege level changes or where data is received from or sent to an untrusted or external source

Abbreviations



A-CPI	Application-Controller Plane Interface
BGP	Border Gateway Protocol
D-CPI	Data-Controller Plane Interface
DDoS	Distributed Denial-of-Service
DNS	Domain Name Server
DPID	Datapath ID
I-CPI	Intermediate-Controller Plane Interface
LAG	Link Aggregations
MAC	Medium Access Control
MITM	Man-in-the-Middle
NAT	Network Address Translation

https://www.opennetworking.org/wp-content/uploads/2014/10/Principles_and_Practices_for_Securing_Software-Defined_Networks_applied_to_OFv1.3.4_V1.0.pdf

03 Network Slicing Security

Network Slicing Security

- Knowledge of slice security conditions is not straightforward considering the multiplicity of authority perimeters and the complexity of dependencies between sub-systems or services
- Slice security mandates continuous monitoring tracking events or anomalies end-to-end
- A particular case is the response to incident which basically require sharing of information from detection towards tenants or adjacent party interconnected
- Detection and remediation are the purpose of smart AI-based strategies that may be applied on a per-tenant basis
- Every tenant may deploy its own detector for attack detection across his slice, in addition to some reporting delivered by the providers

4 Key Concerns of Network Slicing Security

- Resource sharing – one size does not fit all
- Multi-domain security – orchestrating security policies
- Security orchestration – one-to-many attack vectors
- Network immune system – the threat posed by limitless potential entry points

Resource sharing

- Although a fundamental premise of network slicing is that the network is carved into discrete, self-contained units, in many cases each slice must still leverage network-wide resources
- While unique security parameters can be defined for network slices individually, there are security parameters that must be applied to shared network resources
- The opportunity exists for incongruences to exist between a network-wide security policy and a security policy that must be applied to an individual slice

Multi-domain security

- Under the assumption that network slices will become dynamic resources that can be set up, torn down or altered on an on-demand basis, then the presence of SDN-based orchestration is nearly a given
- Security orchestration across multiple network domains also becomes important to ensuring the overall security of individual network slices
- While security vendors have multi-domain security solutions available, it does provide an appropriate segue into the third item on the list

Security orchestration

- In monolithic network architectures, the opportunity for a malicious attack to enter through a common entry point and then gain access to an array of other network resources is somewhat limited
- However, in an SDN-based orchestration scheme, a successful attack on a multi-domain network orchestrator could provide entry points into multiple network domains and/or network slices

Network immune system

- While the previous point describes a security breach that originates from an attack on a central network management point, the converse situation is also a key security concern as applied to network slicing
- Similar to IoT security, whereby the sheer number of IoT sensors and other end-points provides a near-limitless point through which a security attack can originate
- In a similar way, as more network slices are created to support a variety of user equipment, the potential number of attack vectors will increase accordingly
- The ability of quickly identify, isolate, and mitigate threats becomes arguably more important than the ability to defend each point from the possibility of being breached

Network Slicing Security Requirements

- The performance requirements of network slicing may vary significantly in terms of throughput, QoS, latency, security, and more
- According to the 3rd Generation Partnership Project (3GPP), a network slice contains one or more network slice subnets, each of which in turn contains one or more network functions and can also contain other network slice subnets
- These network functions can be implemented and managed as virtualized network functions (VNFs) and/or physical network functions (PNFs)

Network Slicing Security Requirements

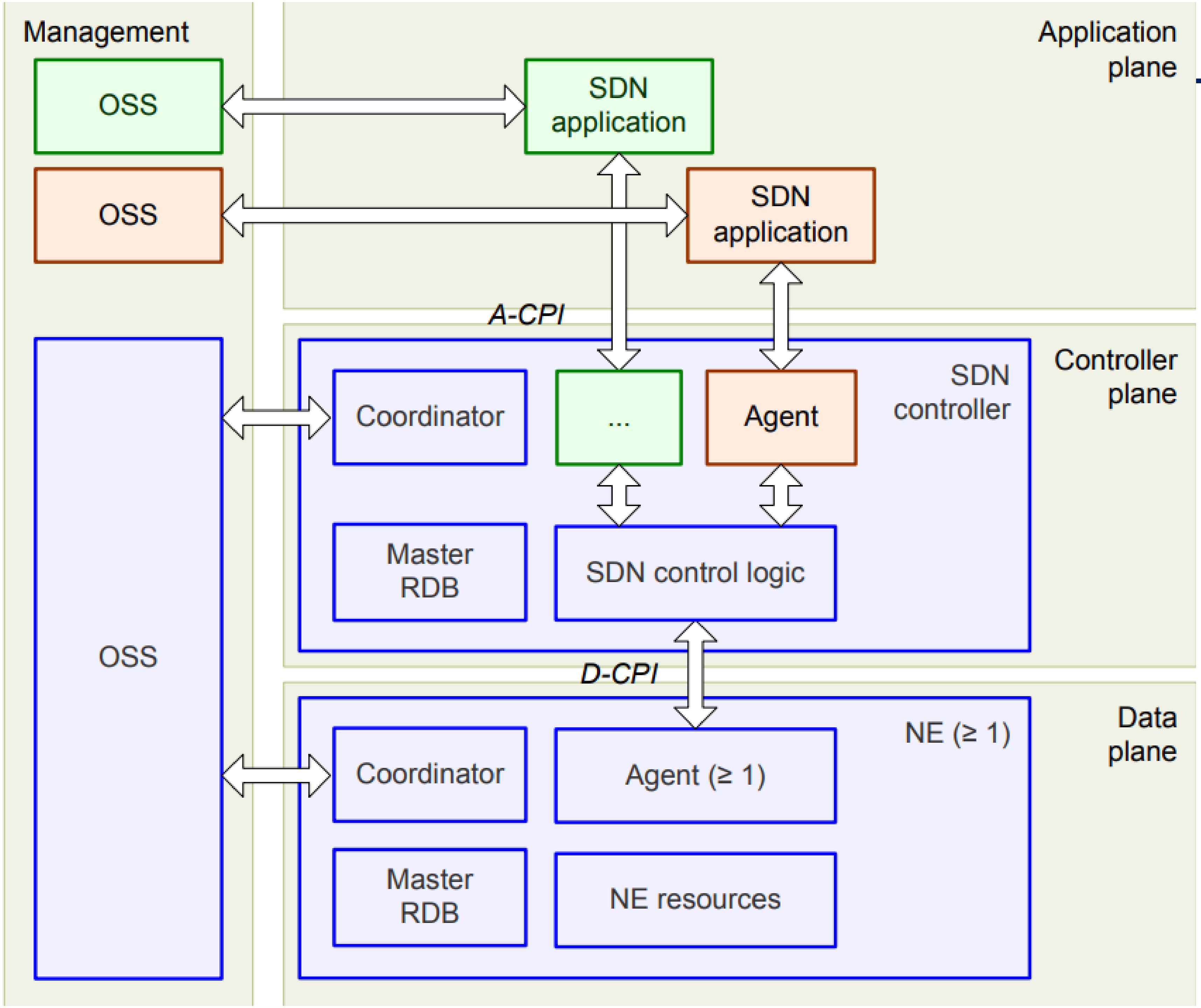
Network Slice Requirements	Corresponding Security Requirements
Deploy/instantiate required PNFs/VNFs to deliver network slice SLA	Appropriate set of security PNFs/VNFs for all use cases
Delivered from RAN (edge cloud) - through core (core cloud) - to Telco cloud/Internet	Dynamic-end-to-end deployment of security functions based on use case requirements
PNF/VNF service chaining	Integration of security functions (PNF/VNF) with SDN and NSH
Slice management and orchestration (3GPP NSMF/ETSI MANO)	Integration with management and network orchestration (MANO)
Network slices isolation	Multi-tenancy and micro-segmentation
Reliability of network slice instances	Dynamic auto scaling and high availability

04 Security of Software-Defined Networking (SDN)

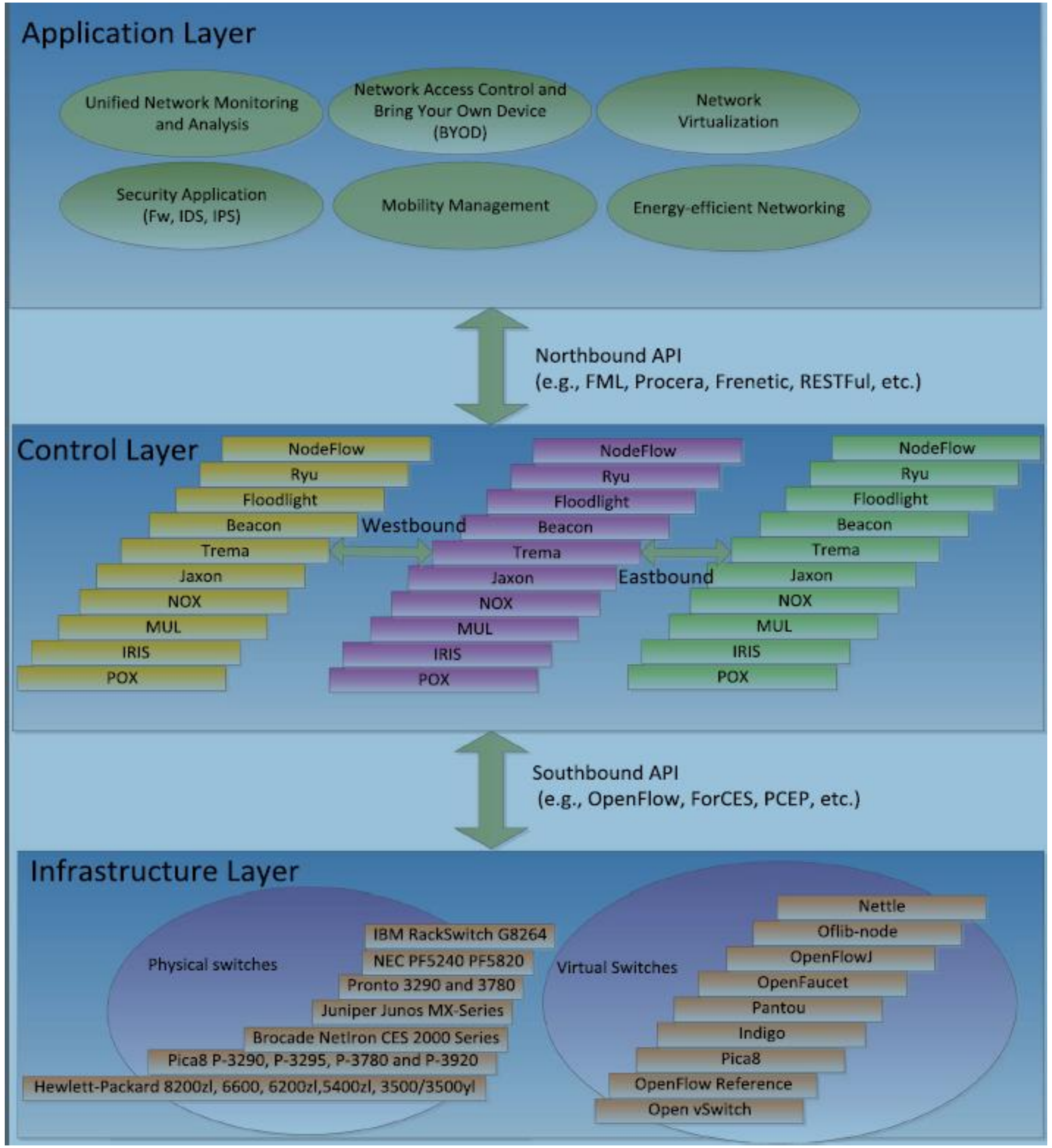
SDN Architecture

- The SDN model proposed by the Architecture and Framework working group is composed of the application plane, the controller plane and the data plane
- SDN architecture is the separation of the controller plane from the data plane
- With programmability and flexibility, new algorithms and applications can be implemented and verified efficiently

SDN Architecture



SDN-Based Cloud



OpenFLow controller

- Load balancing
- Power-saving
- Monitoring mechanisms

SDN-Specific Security Challenges

- New features and new network deployments can introduce faults and risks that open the door for threats that did not previously exist or are more serious than before
  - One provider’s SDN controller can directly access and manipulate another provider’s SDN switches
    - The traditional attack vectors on traffic flows, switches, administrative stations, and recovery and fault diagnosis, the controllers and the communications related to the Controller plane result in new security issues that are specific to SDN

Centralized Control

- Centralized control or logically centralized control (i.e. distributed but coordinated control function) exposes a high-value asset to attackers
- Attackers may attempt to manipulate the common network services or even control the entire network by tricking or compromising a controller
- This is distinct from a larger number of autonomous assets in a completely distributed control domain

Programmability

- New types of threats arise due to the explicit programmatic access SDN offers to clients that are typically separate organizational or business entities
- This new business model presents requirements that do not exist within closed administrative domains in terms of protecting system integrity, third-party data and open interfaces

Traffic and resource isolation

- Operators must ensure that business management and real-time control information of one entity is fully isolated from that of all others
- This element extends to the existing security issue of multi-tenant traffic and resource isolation to avoid interference and misuse
- dynamic interactions introducing further requirements for isolation in order to meet different SLAs, private addressing issues

Trust between third party applications and the controller

- Programmability offers flexibility to implement newly innovated market-driven applications but it also opens the door to malicious and vulnerable applications
- Authentication and different authorization levels should be enforced at the point of application registration to the controller in order to limit the controller exposure

Interface Security protection on A-CPI (Application-Controller Plane Interface) and I-CPI (Intermediate-Controller Plane Interface)

- Beyond the communication with applications through A-CPIs, a controller may be controlled either by an upper layer controller or may work in tandem with another controller at the same hierarchical level
- Lack of protection across these interfaces may lead to malicious attacks on the SDN
- Security attributes and operation checkpoints should therefore be defined for securing A-CPIs and I-CPIs

Challenge of Integrating Legacy Protocols

- SDN interfaces and protocols are being developed in the recognized context of escalating exploitation of technical and process deficiencies, with increasingly severe consequences that could lead to security issues
  - difficulty of retrofitting security capabilities into existing technologies (Domain Name Server (DNS) and Border Gateway Protocol (BGP) are notable examples)
- It is critical that compatibility be checked before implementing legacy protocols (e.g., Network Address Translation (NAT), BGP) into SDN
- It is also important that weaknesses previously addressed by legacy architectures not be repeated or even inflated when building the SDN framework

Cross Domain Connection

- An additional requirement of SDN implementation requires that infrastructure of different domains can be connected
- This can be realized by connecting controllers of different providers via the I-CPI(Intermediate-Controller Plane Interface)
- The mechanisms to establish trust relationships, to determine authorization level in order to prevent abuse and secure channel setup should all be considered

Security Principles

1. Clearly Define Security Dependencies and Trust Boundaries
2. Assure Robust Identity
3. Build Security based on Open Standards
4. Protect the Information Security Triad
5. Protect Operational Reference Data
6. Make Systems Secure by Default
7. Provide Accountability and Traceability
8. Properties of Manageable Security Controls

Principle 1: Clearly Define Security Dependencies and Trust Boundaries

- When specifying a security mechanism for SDN networks, security dependencies between different components must be clarified
  - Circular dependencies must be avoided
- The clear definition of trust boundaries allows for targeted risk analysis and security control evaluation
- Trust boundaries should be defined based on areas of privilege change, information flow across domains (i.e. ingress and egress direction), and dependency on data where confidentiality and integrity cannot be verified

Principle 1: Clearly Define Security Dependencies and Trust Boundaries

- At a minimum, any external dependency should represent a trust boundary as it is reasonable to assume that attacks may arise from external systems
- The interface to external environments should therefore provide sufficient security functionality to prevent or mitigate externally initiated attacks
- External systems should be limited in access via a method of least privilege to reduce the risk to the system
- The management or containment of internally initiated attacks should be considered to prevent impact on the external environment

Principle 2: Assure Robust Identity

- The basis for effective security is the ability to uniquely identify all components and users of a system and verify identities with a trusted source
- Without a strong identity framework, the ability to build effective authentication, authorization, and accounting implementations will be limited
- A robust identity should have the following properties:
  - Ability to distinguish its owner from other entities within a pre-defined scope
  - Ability to be generated, updated, and revoked
  - Impersonation prevention, preferably through strong cryptographic mechanisms
- Analysis of the SDN architecture identifies numerous means for elements inside the system’s trust boundary to compromise the availability of the logically centralized control
- Strong authentication based on assured identity is, therefore, critical to the security of the system

Principle 3: Build Security based on Open Standards

- Proven protocols and methodologies should be implemented in favor of developing or designing new ones
- New protocols and algorithms are created as a last resort when existing requirements cannot be met
  - Transport layer protection is required to secure the OpenFlow™ communication channel for both the Transmission Control Protocol (TCP) traffic header and payload
  - Various TCP enhancement techniques have been previously proposed for this purpose and are widely deployed
- Recommended to adopt such an existing technique rather than to develop a new transport layer solution
- The concept of protocol/algorithm reuse is particularly important in the case of security functionality such as encryption, authentication, and integrity, the solutions for which require significant vetting to prove their strength
- The use of legacy protocols or algorithms (e.g., MD5, Transport Layer Security (TLS) 1.0) that have been proved to be insecure and are no longer recommended by standards organizations should be avoided

Principle 4: Protect the Information Security Triad

- Although security controls by nature should increase the confidentiality, integrity, and availability (CIA) of a system, the security posture of the control should be evaluated for its impact on the overall architecture
- An effective method for evaluating new controls is to determine whether the overall system availability might be reduced as a result
- The control should not introduce new vulnerabilities or exploits
- Any reduction in the effectiveness of the core pillars (CIA) should be identified and mitigated
  - The introduction of a centralized security server into the SDN architecture must be carefully evaluated in case the server’s potential vulnerability to denial-of-service (DoS) attacks might impact system availability
  - Suitable mitigation to this problem must be identified

Principle 4: Protect the Information Security Triad

- Security controls should be constructed in a way that they do not unnecessarily degrade system performance or impose additional system complexity which will likely introduce new security vulnerabilities
- In practice, the eventual solution of a security control is synthetically affected by security requirements, cost, and manageability

Principle 5: Protect Operational Reference Data

- The effectiveness of a security control is directly impacted by the integrity of the reference data (e.g., credentials and sequence numbers), which is a key requirement in making operational decisions
- Incorrect information can lead to unexpected system behavior that can result in a loss of confidentiality, integrity, and/or availability
- The leakage of certain sensitive reference data such as cryptographic keys will cause potential security breaches of the security control
- Operational reference data for all security controls should be clearly defined and protected to a level of continuity consistent with the security policy and the security architecture assumptions

Principle 5: Protect Operational Reference Data

- Reference data must be generated, processed, maintained, and transported securely in expected operational states, state transitions, and during the system lifecycle
    - System initialization, normal system operation, system standby, system failover and system recovery states, and during transitions between these states
- Several security protocols use monotonically increasing sequence numbers to detect replay attacks
- Any uncontrolled rollback of these numbers (particularly following system failure) must be avoided
- This is of particular importance when automated key management is not supported

Principle 6: Make Systems Secure by Default

- Security controls should provide multiple security levels to meet the requirements of all potential system use cases
  - Vary from a state in which a control is disabled to a state that can satisfy the most rigorous security requirements (e.g., deny by default)
  - The system should define a minimum level in which the majority of primary security controls are enabled by default
  - In addition to being enabled, these controls should be configured in a manner that meets minimum criteria to ensure that the control is effective
- Security controls should have the ability to be reconfigured or even disabled, but this should be a conscious decision of the system owner/operator
- For example, when implementing an authentication control, it is important to ensure that there is some form of authentication by default
  - To make the control effective, the authentication should not be set to null or disabled entirely
  - The key security properties (which could be various in different cases) of a system should be ensured across updates, recovery from failures, restarts

Principle 7: Provide Accountability and Traceability

- All security controls should be auditable for the state and actions critical to system security
- Logged data should contain sufficient information for auditing purposes
  - An auditor should be able to not only uniquely identify the entity on whose behalf an action has been carried out but also find out the relevant sequence of the action
- However, it is also important to ensure that the audited data should not contain redundant information and the actions of auditing will not lead to violation of security policy
- The security properties of logged data should be protected to a level of continuity consistent with the security policy and assumptions during its lifecycle
  - The data should be protected against unauthorized access and modifications

Principle 8: Properties of Manageable Security Controls

In addition to the seven principles specified above, when introducing new controls into an architecture or a standard, the following properties of the control should be considered:

- Prior to designing or introducing a security control, the security objectives and assumptions should be clarified
- Security controls should be scalable and designed to support installations from the smallest reference system to the largest deployment without introducing undue complexity
- When introducing new controls, the impact of the solution implementation and lifecycle management should be considered
    - New security functions should only introduce minimal complexity to the implementation
    - A good implementation should be extensible so that additional security control functions can be introduced in the future

Principle 8: Properties of Manageable Security Controls

- Security controls should be easy to implement, maintain, and operate
- Ensure that controls are backward-compatible, or provide an upgrade path that allows current and legacy controls to coexist
- Ensure that controls are well documented and based on well-defined standards
- It should always be possible to revoke and modify security credentials as part of a system’s lifecycle
- Wherever possible, all security controls should support automation to ensure that controls are properly implemented
  - In many cases, manual processes may lead to improper configuration, which may reduce the effectiveness of a control
- The ability to monitor, troubleshoot, and debug any system is fundamental to its successful adoption

Security Requirements for Protocols

- A set of security requirements are derived from each security principle introduced in the previous
- These security requirements specifically relate to the design and development of protocols
- Goal:
  - The security requirements are intended to help the designers of security mechanisms to:
    - Address or mitigate the potential for malicious exploitation of ONF protocols
    - Evaluate and control the negative effects (e.g., overheads, new security weaknesses) that may be introduced by the deployment of security mechanisms

Security Requirements for Protocols

The following issues are out of scope:

- Security issues caused by improper implementation of security mechanisms
- Security issues caused by improper implementation of security mechanisms
- Physical attacks against SDN network assets (e.g., disabling network devices or breaking the cables connecting them)

1. Clearly Define Security Dependencies and Trust Boundaries

Before designing the security solution for a SDN protocol, the application scenarios in which the protocol will be used and potential threats associated with its use must be carefully analyzed. In each scenario, authentication and authorization must be performed between network elements on each side of the trust boundary before signaling packets are exchanged. In addition, packet level security protection must be provided for signaling packets.

1. Clearly Define Security Dependencies and Trust Boundaries

A. The security solution of an SDN protocol should support mutual authentication between two SDN components running the protocol

B. The security solution of an SDN protocol should provide the authorization function for the SDN components running the protocol in the case where an SDN component is only approved (based on certain security policies) to perform a limited set of operations on the resources of another SDN component

C. The SDN protocol processing components should agree upon the security associations (e.g., key materials, algorithms etc.) for securing their communications before exchanging any protocol packets

D. In the case that a protocol exchange could be accessed by an attacker, the security mechanism should be able to provide integrity protection (and optionally provide confidentiality protection) for protocol packets

- In practice, confidentiality protection can be optional and provided only when the protected content is sensitive

2. Assure Robust Identity

A. Each entity (SDN devices or users) running the ONF protocol should have an ID that distinctly identifies the owner of the ID within a required scope. The possession of the identity should be verifiable through cryptographic methods during authentication

B. In the protocol specification, the issues related to management of IDs during their lifecycle (including generation, distribution, maintenance, and revocation) should be considered

It is not intended that a complete solution for ID management be specified in each protocol specification. However, ID management should be specified and provided as a fundamental service in ONF security solutions

3. Build Security based on Open Standards

A. Existing security protocols/mechanisms should be applied first
   - Security extensions to the base ONF protocols or new security protocols are proposed only when there is no existing security protocol meeting all the security requirements

B. Non-standard or vulnerable algorithms/protocols should not be adopted
   - Both MD5 and SHA-1 are now known to be vulnerable to collision attacks. These two algorithms are therefore not recommended for use in the security solutions proposed by ONF

C. Both MD5 and SHA-1 are now known to be vulnerable to collision attacks. These two algorithms are therefore not recommended for use in the security solutions proposed by ONF
   - Non-compliant packets or corrupted control messages should be handled correctly by the entities communicating via ONF protocol

4. Protect the Information Security Triad

A. The security solution for an SDN protocol should consider the security issues raised in multiple layers
   - For example, the packet headers and signaling messages of underlying transport protocols should be properly protected. BGP running over TLS does not solve the problem of an attacker being able to send a spoofed TCP FIN or TCP RST and causing the BGP session to go down

B. The protocol specification should provide the mechanism to manage and rate control messages initiated by activity in the control/data plane in order to mitigate potential DoS/DDoS threats

C. It is desirable for the SDN control protocol to be extensible to support additional signaling messages/options for dealing with future network attack types
   - It is common that security mechanisms/extensions for a protocol are proposed after the publication of the base protocol. Therefore, it is desirable for extensibility to be considered during the design of the base protocol such that the protocol can be extended for future security purposes

4. Protect the Information Security Triad

D. A security protocol should be defined in such a way that each protocol message consists of sufficient information to instruct the message recipient(s) to correctly process it, e.g., being able to verify the integrity of the message
   - This requirement is defined to avoid the case of a security mechanism being confused or overwhelmed by bogus packets. For example, when a security mechanism uses multiple keys to protect the communications between two network components, a key ID may need to be carried within a packet to indicate which key is used to verify the packet

E. The amplification effect should be considered
   - If a device has to generate a response that is much larger than the request, the device may be used by an attacker to perform reflection attacks

F. The proposed security mechanism should avoid the introduction of further, knock-on security issues
   - For example, if the security solution for an ONF protocol introduces new centralized servers, it is necessary to identify how to protect them from becoming new attack targets (e.g., vulnerable to DDoS)

5. Protect Operational Reference Data

- If the loss or improper/uncontrolled modification of certain reference data will result in potential security risks, such information should be securely maintained (e.g., integrity (and optionally confidentiality) protection applied when sensitive information is stored) and only be accessed by authorized entities
    - In practice, such information normally includes access control policies, certificates, private keys, service descriptions and policy, etc. Note that sometimes the uncontrolled rollback of some data such as time and counters will result in security issues, e.g., Y2K

6. Make Systems Secure by Default

A. The security solution for an ONF protocol may need to specify different default configuration and deployment plans for multiple application scenarios in order to ensure the security of network devices using the SDN protocol across updates, recovery from failures, restarts

- Such default configuration information may include default behavior, default algorithms, default key length, types of certificate, pre-defined access control policies

B. Mandatory cryptographic algorithms and security protocols should be specified

7. Provide Accountability and Traceability

A. When designing an ONF protocol, critical events or incidents should be notified and logged for auditing purposes as well as reported to the required entities for reliability purposes

B. All logging information from different SDN components should be securely stored (minimally with integrity protection). Confidentiality and integrity protection must be provided when the logs are transported to remote servers for analysis

C. Critical status and counters for different SDN components must be logged for monitoring purposes. Those logs must be regularly monitored in order to detect malicious activities in regards to different SDN components

8. Properties of Manageable Security Controls

In addition to the requirements introduced above, a well-designed security mechanism for ONF protocols should also adhere to the following requirements.

A. The security mechanism should be able to support various security algorithms so that a user can select their preferred algorithm to secure the system
   - This requirement can be used to enable users to select different levels of security protection according to different security objectives

B. The security mechanism should be extensible and support introducing new algorithms or new security functionalities when necessary

C. A security mechanism should be able to support automated key/credential management and consider the issues with generation, distribution, and revocation of security credentials
   - Key management is closely related with ID management. See also 4.2.2. This requirement does not preclude the usage of manual key management

OpenFlow Switch

- Issues, Countermeasures and Principles/ Requirements for OpenFlow Switches

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Physical Ports	A physical device can be inserted or changed on the traffic monitoring perhaps leading to a network attack.	Enable the controller to notice the modification of far-end MAC Addresses and other link layer states.	Principle 4: Protect the Information Security Triad REQ 4.4
Logical Ports	Tunnel ID is not provided in Ports Statistics messages	Enable the controller to learn the tunnel IDs associated with the logical ports	Principle 4: Protect the Information Security Triad REQ 4.4
Reserved Ports	No way for applications to collect the statistical information of reserved ports(expect LOCAL)	Enable the controller to learn such information of reserved ports	Principle 7: Provide Accountability and Traceability REQ 4
Counters	Roll-back of counters is out of control	Discuss how such conditions will not cause inconsistencies	Principle 5: Protect Operational Reference Data REQ 4.5

OpenFlow Switch

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Matching	No specification for handling malformed packets	Any non-compliant incoming packet(IEEE and/or RFC specification) should be dropped by the switch/controller. In addition, a mechanism to check malformed or corrupt OpenFlow control packets should be implemented in the switch/controller.	Principle 3: Build Security based Open Standard REQ 4.3
Flow Removal	Inconsistent flow table view at the controller	Any changes to the forwarding state (particularly flow removal initiated by non-master controller) in the switch must be communicated/notified to the controller. This ensures that the controller and switch have a consistent view of the forwarding topology.	Principle 5: Protect Accountability and Traceability REQ 4

OpenFlow Channel and Control Channel

- Issues, Countermeasures and Principles/Requirements for OpenFlow Channels

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Connection Setup	No information provided on TLS usage	Clarification on TLS usage should be provided or a pointer to specification in a companion protocol.	Principle 6: Make Systems Secure by Default REQ 4.6.1
	TLS does not provide protection of TCP headers.	Security mechanisms such as TCP-AO that provide protection to TCP headers could be considered.	Principle 3: Build Security based on Open Standards REQ 4.3.1
	No information on managing credential details (keys, certificates)	Credentials should be configured and managed by a switch management protocol like OF-Config. A pointer in OF protocol is required	Principle 5: Protect Operational Reference Data REQ 4.5.1 Principle 8: Properties of Manageable Security Controls REQ 4.8.3
Connection Interruption	Potential for reduced security level following connection interruption.	Same level of security should be maintained before and after the connection interruption. The controller should be notified of the switches current state after reconnection. In this case, a message should be generated to the controller following any transition in mode of operation (from “fail-standalone mode” to “fail-secure mode”).	Principle 6: Make Systems Secure by Default REQ 4.6 1

OpenFlow Channel and Control Channel

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Encryption	Only authentication using certificates is discussed, this implies the exclusion of message authentication based on pre-shared key Fail to discuss the cases where only integrity protection is provided	Add statements regarding support for multiple types of authentication mechanism Message integrity protection should be supported when the information transported over the OpenFlow messages is not sensitive.	Principle 8: Properties of Manageable Security Controls REQ 4.83 Principle 4:Protect the Information Security Triad REQ 4.4.4

OpenFlow Channel and Control Channel

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Multiple Controllers	Potential for conflict between multiple controllers with Equal role.	Employ policy conflict resolution mechanisms at the controller or add additional flags in the specification to detect conflict flows like CHECKOVERLAP	Principle 8: Properties of Manageable Security Controls REQ 4.8.2
	Fingerprinting is possible by Asynchronous messages being sent to all attached controllers.	Mutual authentication between controllers and switches is required regardless of controller role.	Principle I: Clearly define Security Dependencies and Trust Boundaries REQ 4.1.1.
	Malicious controller requests role change to Master, demoting the legitimate controller to Slave.	A message should be sent to the Master controller to identify a role change. A message should be sent to all controllers upon new controller connection.	Principle 4: Protect the Information Security Triad REQ 4.4.3.
	Unauthorized access or manipulation of controller connection role	Secure switch storage of controller connection information.	Principle 5: Protect Operational Reference Data REQ 4.5.1.
	Ambiguous role status event notification	Role Status Event – Reason should identify which controller requested the role change and report that in the reason to other Controllers whose role is changed; "Another controller asked to be master" is ambiguous.	Principle 7: Protect Accountability and Traceability REQ 4.7.1

OpenFlow Channel and Control Channel

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Auxiliary Connections	Lack of notification when receiving an invalid DP1D When a key is used to protect different channels, the compromise of one channel may result in the compromise of others.	An error message should be generated for an incoming packet with an invalid DP1D Different keys should be used for each connection (main and auxiliary).	Principle 7: Protect Accountability and Traceability REQ 4.7.1 Principle 4*:* Protect the Information Security Triad REQ 4.1.3

Additional Issues

- Issues, Countermeasures and Principles/ Requirements for no TLS on D-CPI

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Multiple Controllers	Fingerprinting is possible by Asynchronous messages being sent to all attached controllers	Mutual authentication between controllers and switches is required regardless of controller role	Principle 1: Clearly define Security Dependencies and Trust Boundaries REQ 4.1.1
	Integrity of role request messages	Use secure channel communication	Principle 1: Clearly define Security Dependencies and Trust Boundaries REQ 4.1.4
Auxiliary Connections	Manipulation of Controller role information across an insecure auxiliary connection	All controller-switch Connections (auxiliary and main) should use Secure channel communication	Principle 1: Clearly define Security Dependencies and Trust Boundaries REQ 4.1.4
	If the Datapath ID and auxiliary ID are not sufficiently random, an attacker may perform offline attacks on the auxiliary connections over LJDP	Extend the ID to 96 bits. The lower 48 bits are the switch MAC address, while the top 48 bits are randomly generated	Principle 2: Assure Robust Identity REQ 4.2.1

Summary of Recommendations

The above present the countermeasures or recommendations generated based on the analysis of the OpenFlow Switch Specification v1.3.4

The recommendations are separated into two strands; (1) securing the OpenFlow protocol itself, and (2) securing the data plane. For (1), it is presented the recommendations as OpenFlow bugs to be fixed. For (2), additional features are proposed which do not directly benefit the security of OpenFlow communications but could be used to enhance the capability of Network components to deal with attacks on the Data Plane

Securing the OF Protocol

1. Use and specification of TLS
2. Connection Interruption Issues
3. Multiple Controllers: Role Change and Status
4. Additional Recommendations for Securing OpenFlow

1. Use and specification of TLS

Issue: The use of TLS is currently under-specified in the document

- No information on TLS version or usage information
- Need clear specification on credential management

Recommendations:

- The specification should recommend/state the use of a secure version of TLS (i.e., 1.2 or greater) or a TLS equivalent protocol (i.e., DTLS for securing messages over UDP) for auxiliary connections

1. Use and specification of TLS

- While the use of plain TCP is understandable, the specification should explicitly callout and recommend the use of TLS for all connections
- Include the recommended mandatory cipher suite to be supported by OpenFlow switches: TLS_RSA_WITH_AES_256_CBC_SHA256
- Ability to configure different cipher settings
- Key management requirements: For instance, Different keys should be used for each connection (main and auxiliary)
- In consideration of the above points, we recommend the specification to provide a pointer to configuration protocols “It is recommended to configure and manage security credentials (cipher settings and certificates) using a switch management protocol like the OF-Configuration protocol”

2. Connection Interruption Issues

Issue:

- No notification for a transition in mode of operation (from “fail-standalone mode” to “fail-secure mode”)
- Potential for reduced security level following connection interruption

Recommendations:

Same level of security should be maintained before and after the connection interruption

- A message should be generated to the controller following any transition in the switch mode of operation. This also helps in deciding if the controller should read all flow entries after re-connection
- The mode of operation can be sent as part of switch feature reply or switch configuration message in OpenFlow

3. Multiple Controllers: Role Change and Status

Issue:

- “When a controller changes its role to OFPCR_ROLE_MASTER, the switch changes all other controllers with the role OFPCR_ROLE_MASTER to have the role OFPCR_ROLE_SLAVE, but does not affect controllers with role OFPCR_ROLE_EQUAL”. There can be only 1 Master Controller. The text should be changed to reflect this
- “When the switch performs such role changes, if a controller role is changed from OFPCR_ROLE_MASTER to OFPCR_ROLE_SLAVE, the switch must generate a controller role status event for this controller informing it of its new state”. The switch must notify role status event when a controller role is changed to either SLAVE or EQUAL

Recommendations:

- A message should be sent to all controllers upon new controller connection
- Role Status Event message: Reason should include some form of information to indicate which controller initiated the request rather than sending an ambiguous reason (e.g., Another controller asked to be master)

4. Additional Recommendations for Securing OpenFlow

- Counter updates to the controller should be set at pre-defined intervals and with acknowledgment. The rollover of counters that may cause potential inconsistency needs to be controlled
- An error message should be generated for an incoming packet with an invalid DPID
- A mechanism to check malformed or corrupted OpenFlow control packets should be implemented and strictly enforced in the switch and all controllers
- For stronger security guarantees: Consider the possibility of using a security protocol which could protect the TCP headers (e.g., TCP-AO)
- The controller should acknowledge flow removal messages from the switch

Securing the Data Plane

- MAC Address modification may be reported
- The controller should be able to learn the tunnel IDs associated with logical ports
- The controller should periodically collect the statistical information of ports
- State transition of all SDN components should be logged
- All logged information should be protected
- Design flow-control mechanism to assure reliable updates and communications between controllers and switches (more research needed)

Securing the Data Plane

- Enforce message validation and integrity to avoid unintended consequences of misconfiguration of instantiation of corrupt table entries
- Implement a PKI CA to manage trust, authenticity, revocation and repudiation
- Ensure authenticity of communications endpoints within the OF SDN fabric (802.1x)
- Employ policy conflict resolution mechanisms at the controller
- Secure switch storage of controller connection information

SDN Knowledge Summary

- Programmability can provide opportunities to enhance the security posture of networks
  - It may be possible to use SDN techniques to construct a security solution that is able to coordinate both network and security devices to detect and react to attacks in a more flexible manner
  - The implementation of new network security functionality should not be achieved at the expense of overall system integrity and security
- The objective of this document is to present a set of high-level security principles that should be applied to ensure that products based on ONF-developed standards and architectures can be implemented in a consistent, fundamentally secure manner
  - This is a foundational work of ONF security

SDN Knowledge Summary

- In order to illustrate the implementation of these principles in design and development, a set of security requirements associated with the individual security principles is presented but specifically applied to securing SDN protocols

Finally, a set of recommended corrective measures for the OpenFlow v1.3.4 protocol has been identified based on the detailed security requirements

Challenges of SDN-Based Cloud

- Performance
- Availability
- Scalability
- Security
    1. Unauthorized access
    2. Data leakage
    3. Data modification
    4. Malicious applications
    5. Configuration issues
    6. Denial of service

Main Potential Threat in SDN

- Forged or faked traffic flow
- Attacks on vulnerabilities in switches
- Attacks on control plane communications
- Attacks on and vulnerabilities in controllers
- Lack of mechanisms to ensure trust between the controller and management applications
- Attacks on and vulnerabilities in administrative stations
- Lack of trusted resources for forensics and remediation

Unauthorized access

- Attacker can easily access network resources and carry out network manipulation by disguising as a suitable network application
- Attacker is able to hijack the whole controller, it becomes the acquirement of access permissions for the entire network system
  - Leads insertion and modification of the flow strategies for network devices

Example

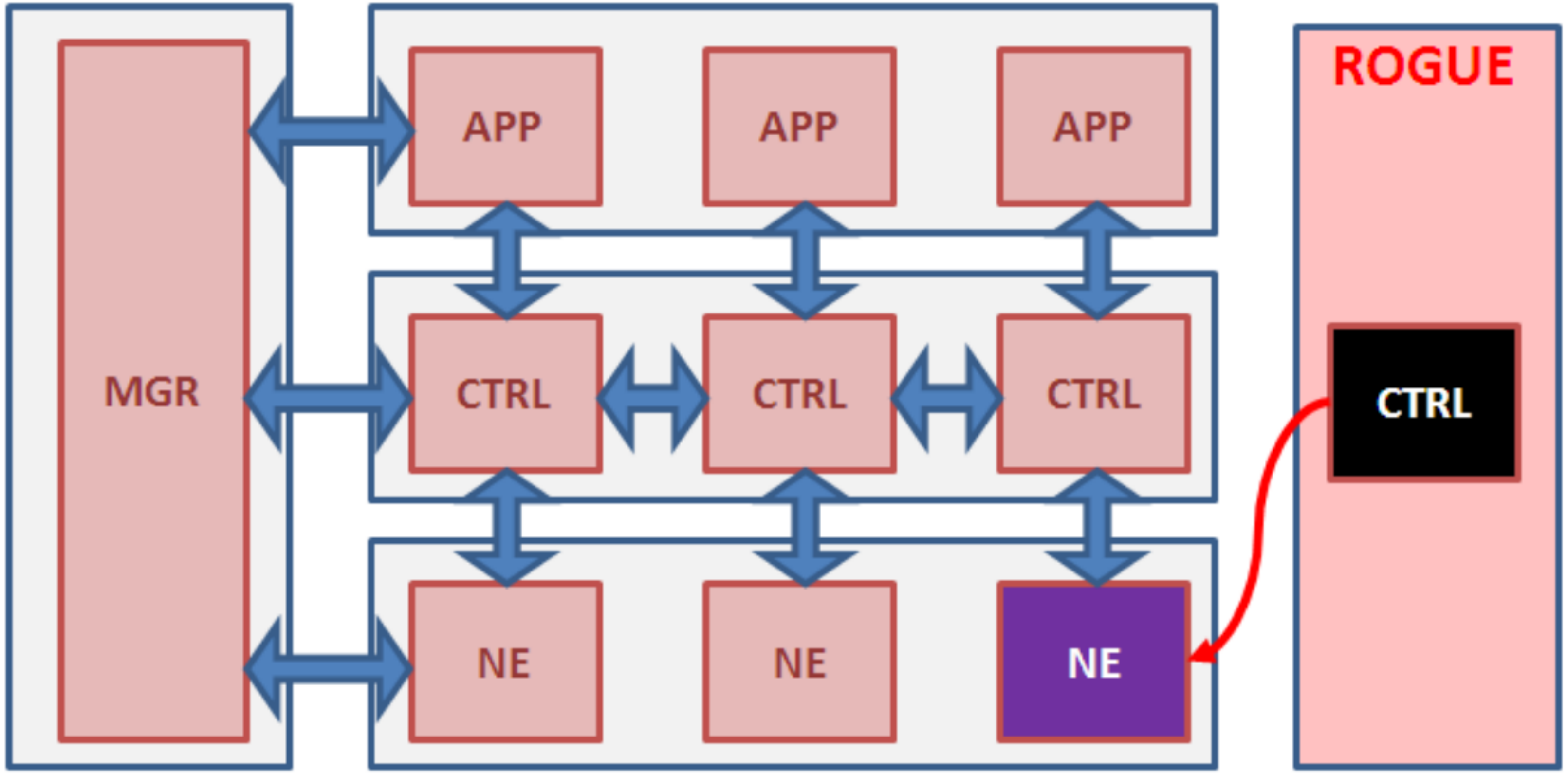
- Letting the switch drop all the incoming traffic, or making it an attack platform
- Unauthorized access using Password Brute-Forcing or Password-Guessing Attacks
- Unauthorized access using Application Exploitation Attack

Data leakage

- Applications can have vulnerabilities that result in server- side data leaks
- Securing sensitive data from applications while ensuring reasonable performance and without requiring developers to rewrite applications is challenging

Data modification

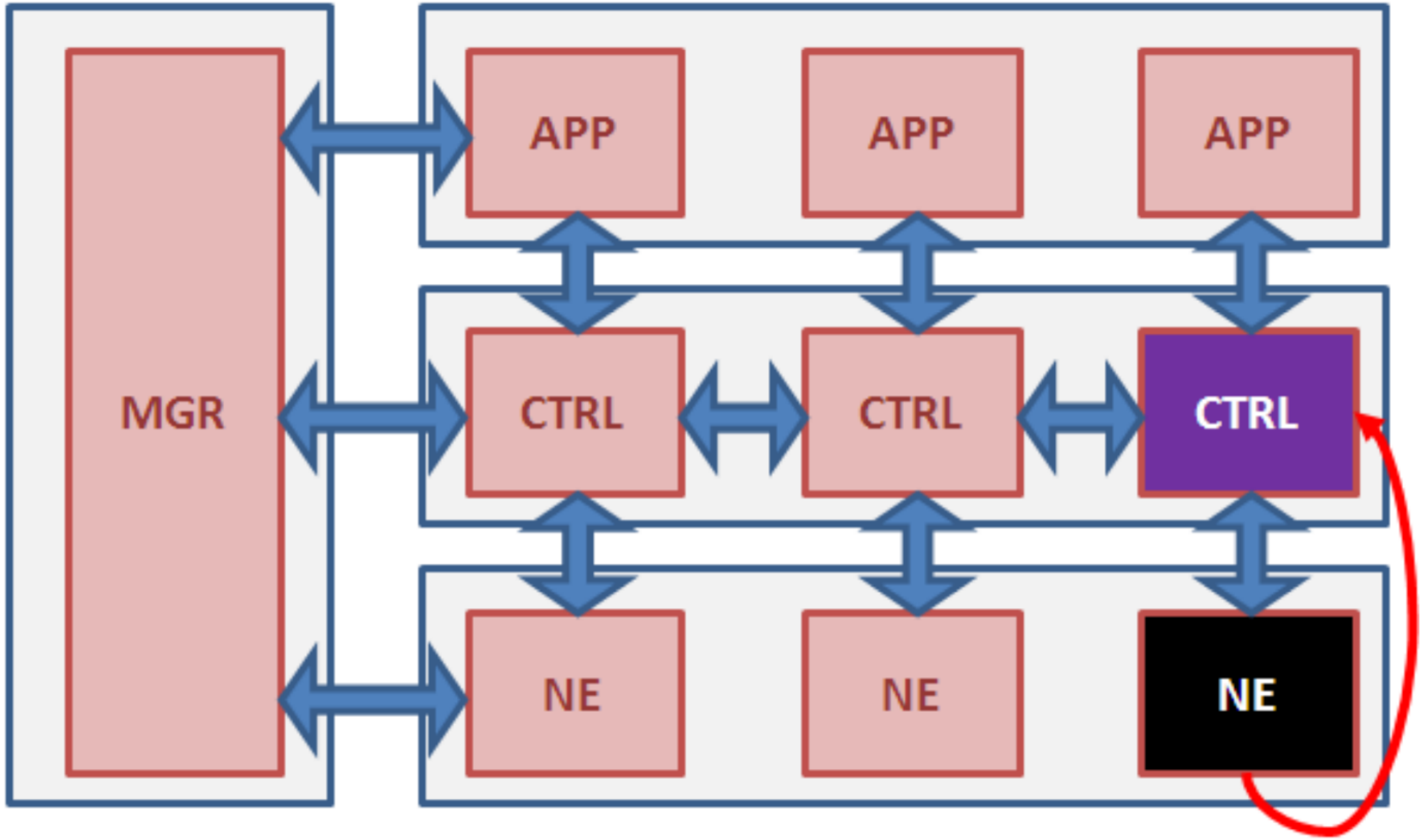
- Source alters data which it is not explicitly authorized to modify (loss of integrity)
- Attacker may spoof the identity of a legitimate controller to attempt to interact with a network element to instantiate flows into the network element's flow table



https://www.blackhat.com/docs/us-15/materials/us-15-Hizver-Taxonomic-Modeling-Of-Security-Threats-In-Software-Defined-Networking-wp.pdf

Example

- A compromised network element could forge network data to poison a controller’s view of the network topology
- This attack could be leveraged to further carry out a variety of other attacks on the network, for instance, for diverting traffic flows in the attacker’s direction for eavesdropping

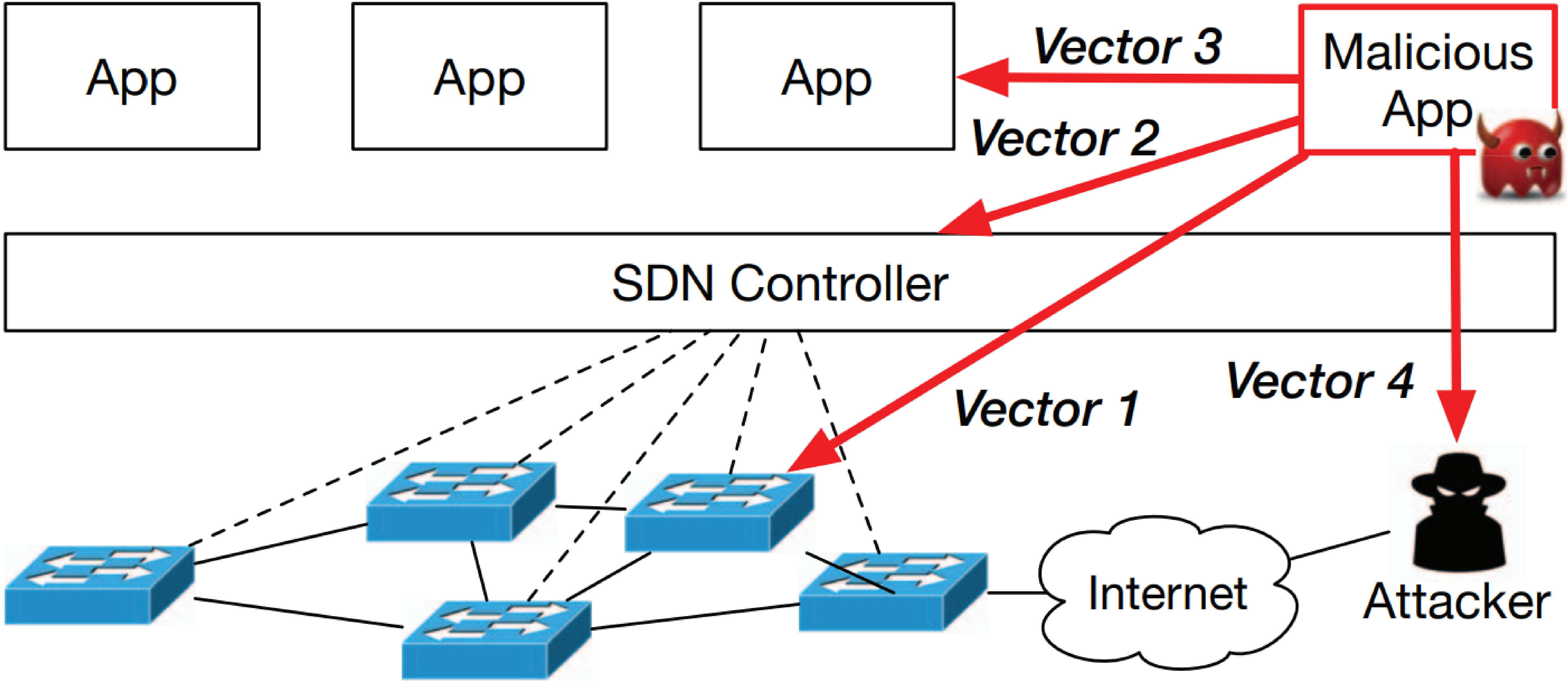


https://www.blackhat.com/docs/us-15/materials/us-15-Hizver-Taxonomic-Modeling-Of-Security-Threats-In-Software-Defined-Networking-wp.pdf

Malicious applications

- Controllers not only provide centralized control of SDN, but also implement open and programmable APIs to ultimately establish an open network environment, where anyone can develop and deliver useful SDN applications
- Can be easily developed and distributed by untrusted entities and can even possess full control of SDN

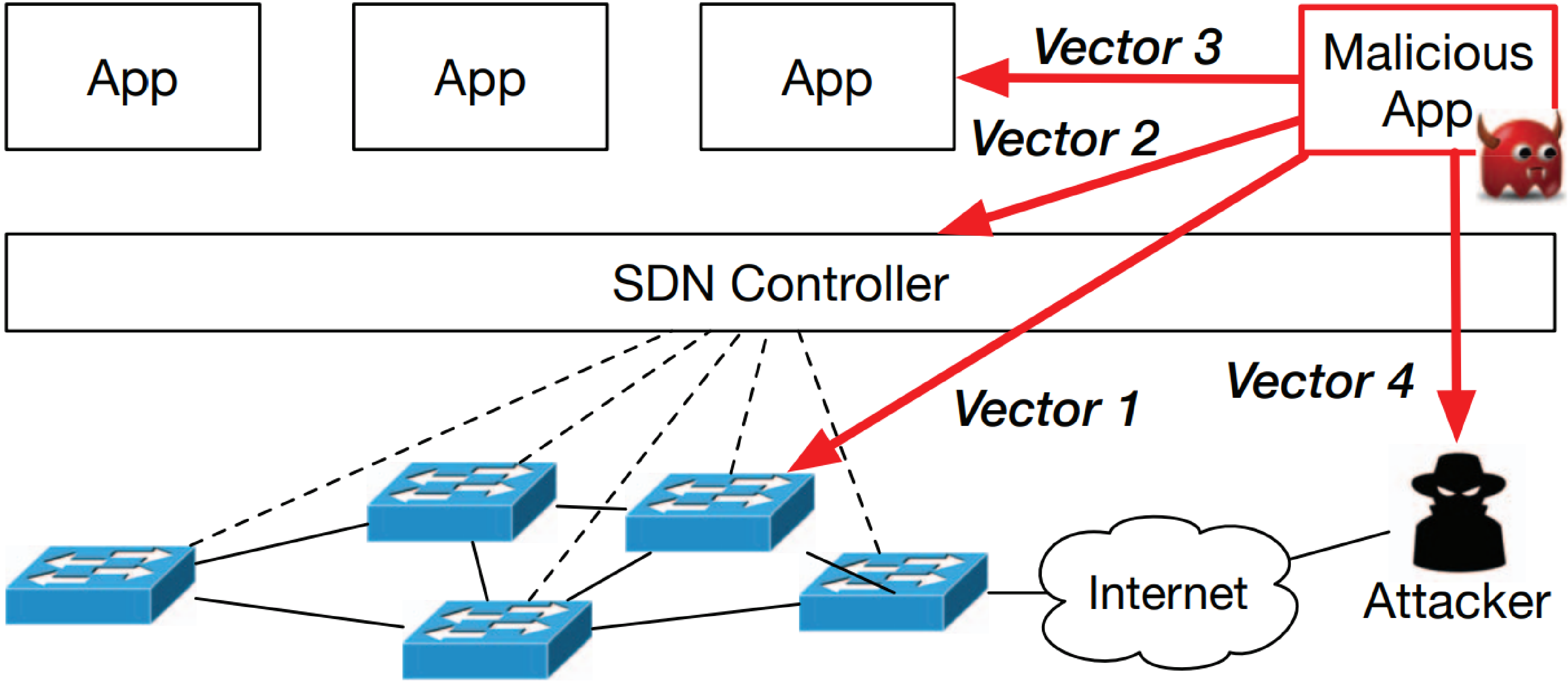
Example



Chanhee Lee, Changhoon Yoon, Seungwon Shin, Sang Kil Cha KAIST, “INDAGO: A New Framework For Detecting Malicious SDN Applications ”

Vector 1: Interference in Data Plane

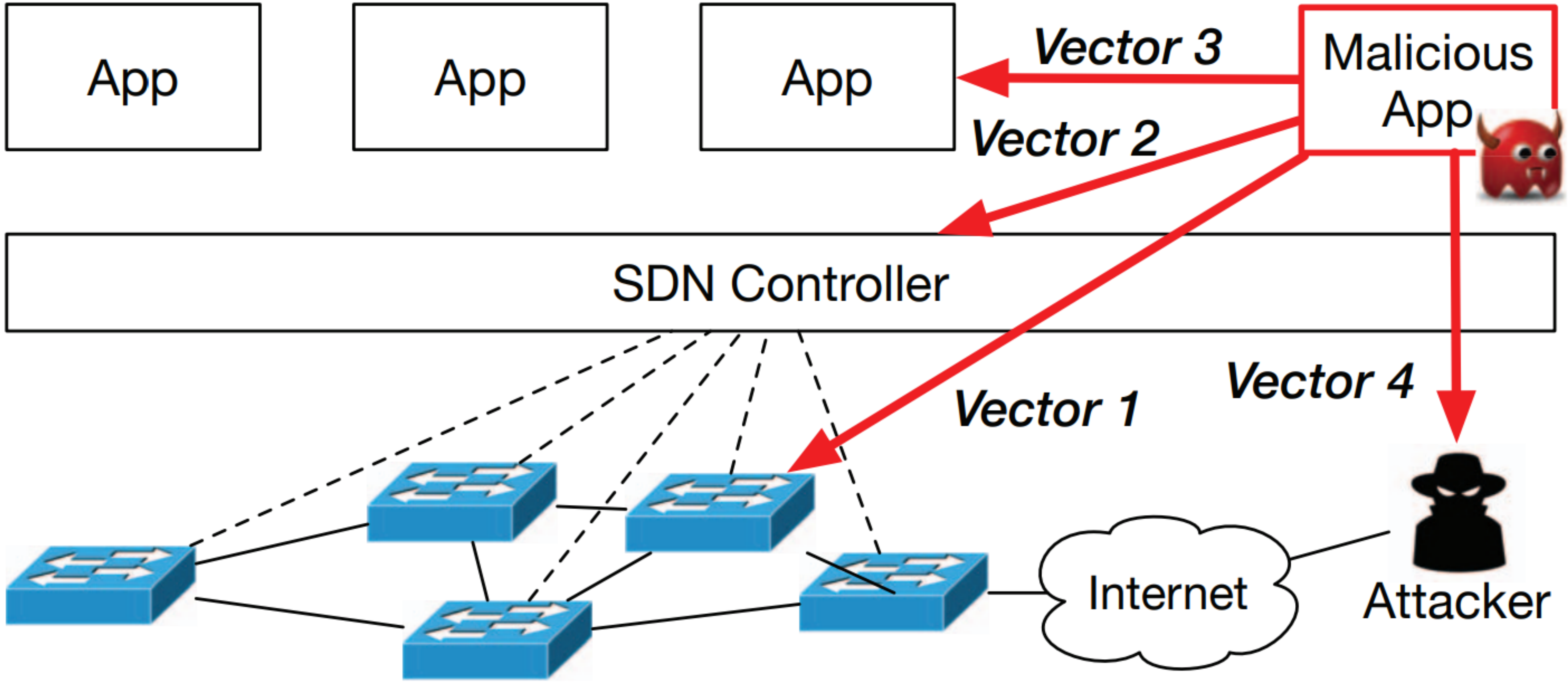
- Malware can control the data plane by manipulating OpenFlow messages
- Malicious application can sniff network packets and even reroute legitimate network traffic by putting a forged flow entry on a network device



Chanhee Lee, Changhoon Yoon, Seungwon Shin, Sang Kil Cha KAIST, “INDAGO: A New Framework For Detecting Malicious SDN Applications ”

Vector 2: Intrusion into SDN Controllers

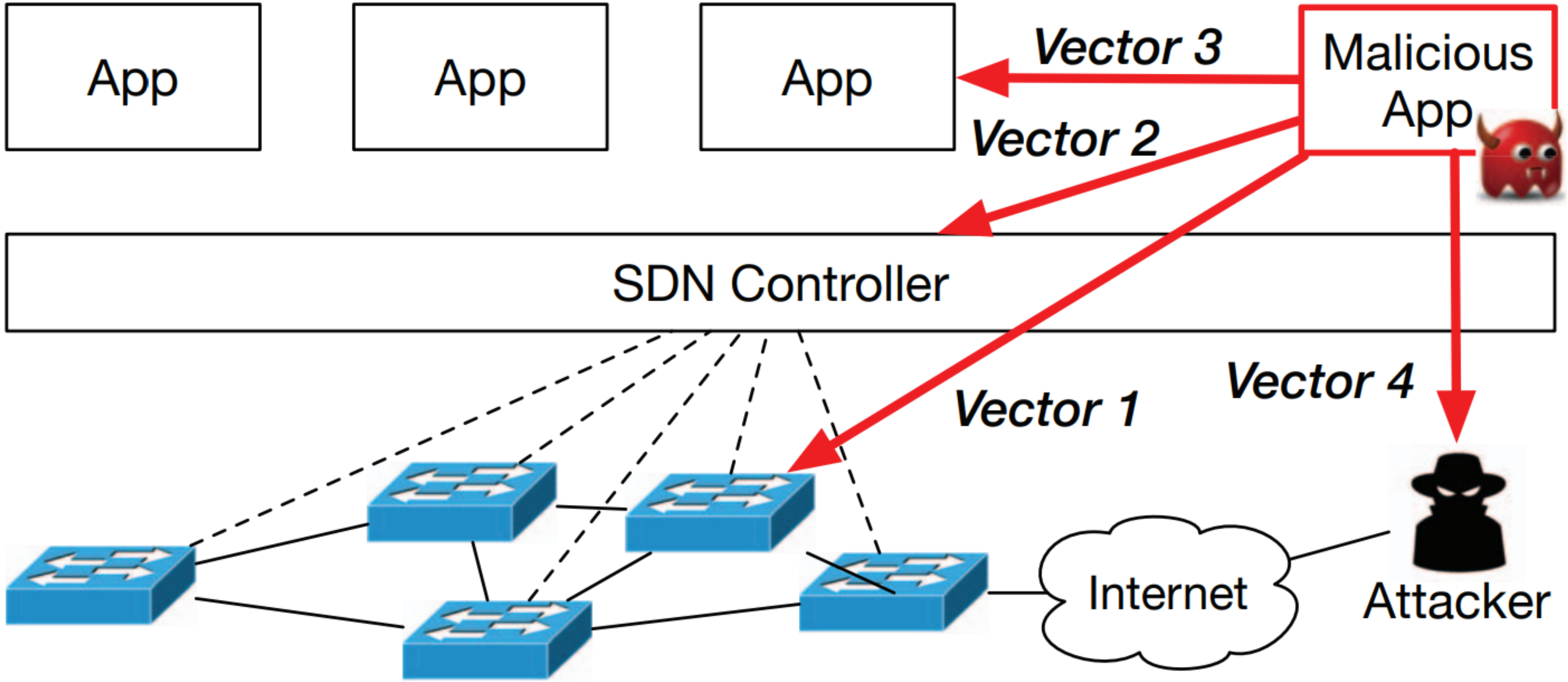
- Malware can interfere in network operations through direct access to the resources of SDN controllers by exploiting controller APIs
- Malicious application can reorder a network service chain, poison the network topology, or fabricate the statistics of network traffic



Chanhee Lee, Changhoon Yoon, Seungwon Shin, Sang Kil Cha KAIST, "INDAGO: A New Framework For Detecting Malicious SDN Applications"

Vector 3: Attack on SDN Applications

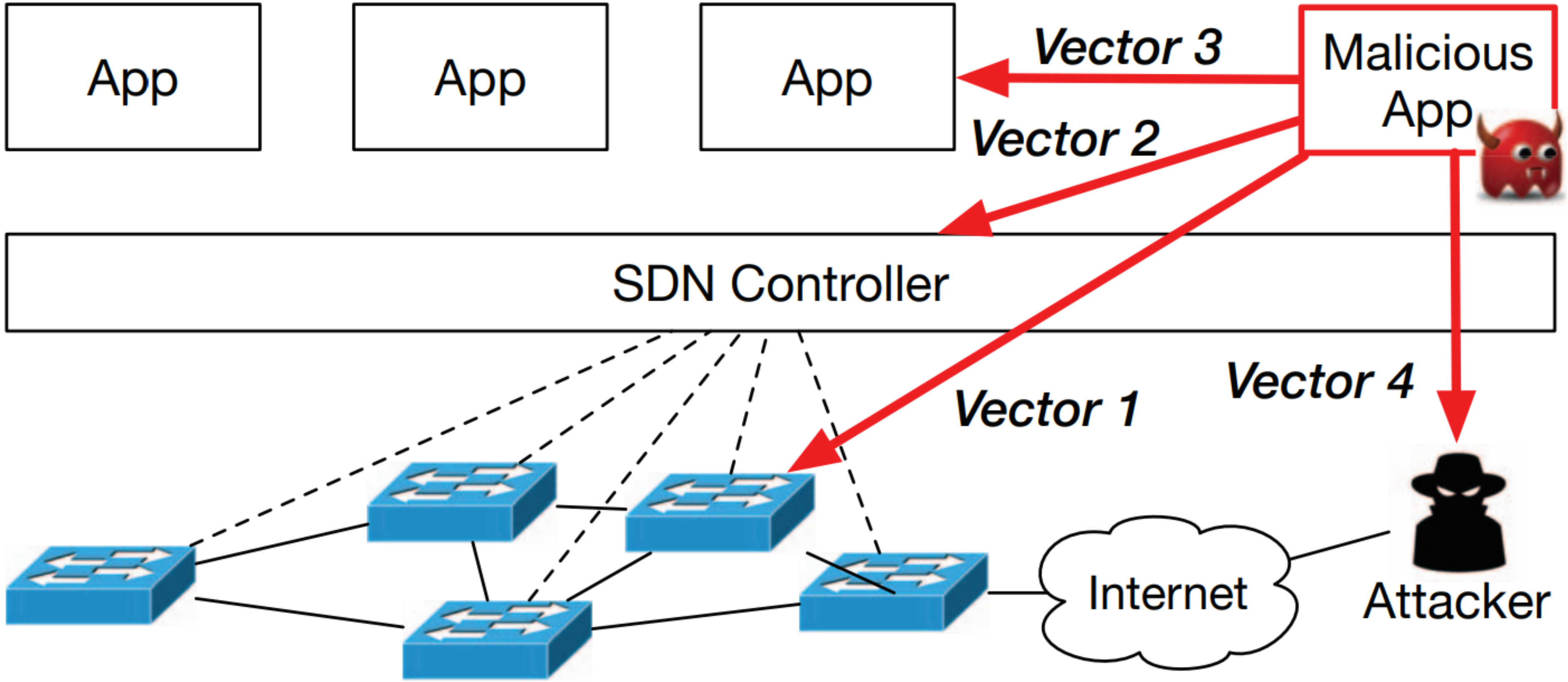
- Malware can kill or run other SDN applications
- Deactivating security applications such as firewall and IDS allows an attacker to bypass existing defenses



Chanhee Lee, Changhoon Yoon, Seungwon Shin, Sang Kil Cha KAIST, “INDAGO: A New Framework For Detecting Malicious SDN Applications ”

Vector 4: Critical Information Leakage

- Malware can leak critical information of a network to adversaries
- Attackers can obtain controller configurations, security options, network status, and flow tables, which can be used to conduct further attacks

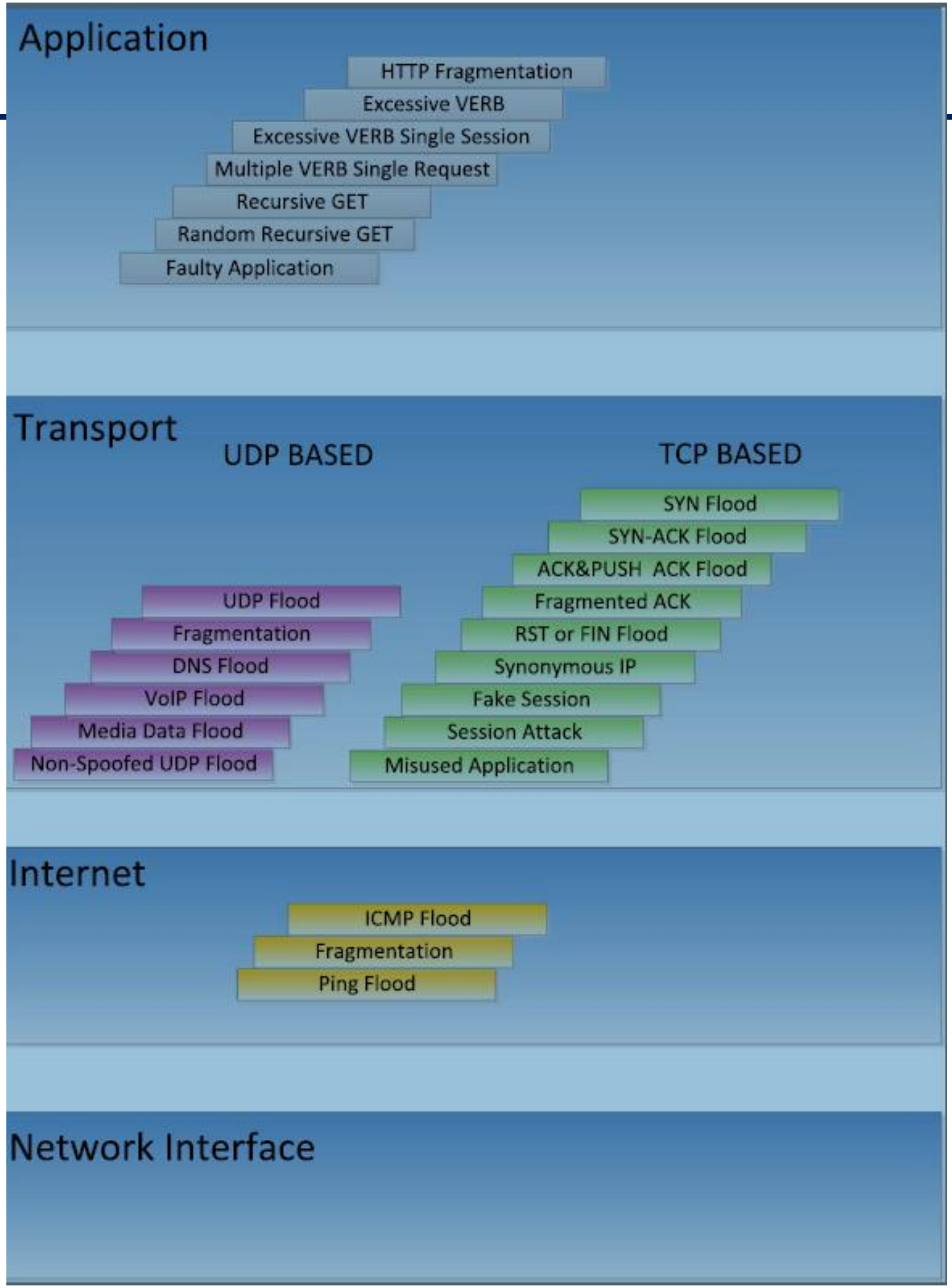


Chanhee Lee, Changhoon Yoon, Seungwon Shin, Sang Kil Cha KAIST, "INDAGO: A New Framework For Detecting Malicious SDN Applications"

Configuration issues

- Misconfigurations in an orchestrated network can lead to serious threats such as disclosure of information
- Configuration mistakes can lead the orchestrator to originate data without authorization
- Configuration issues can lead to mistaken or incorrect data sharing such as the case in which an orchestrator exposes resources which the operator does not actually own

Different DDoS Attacks



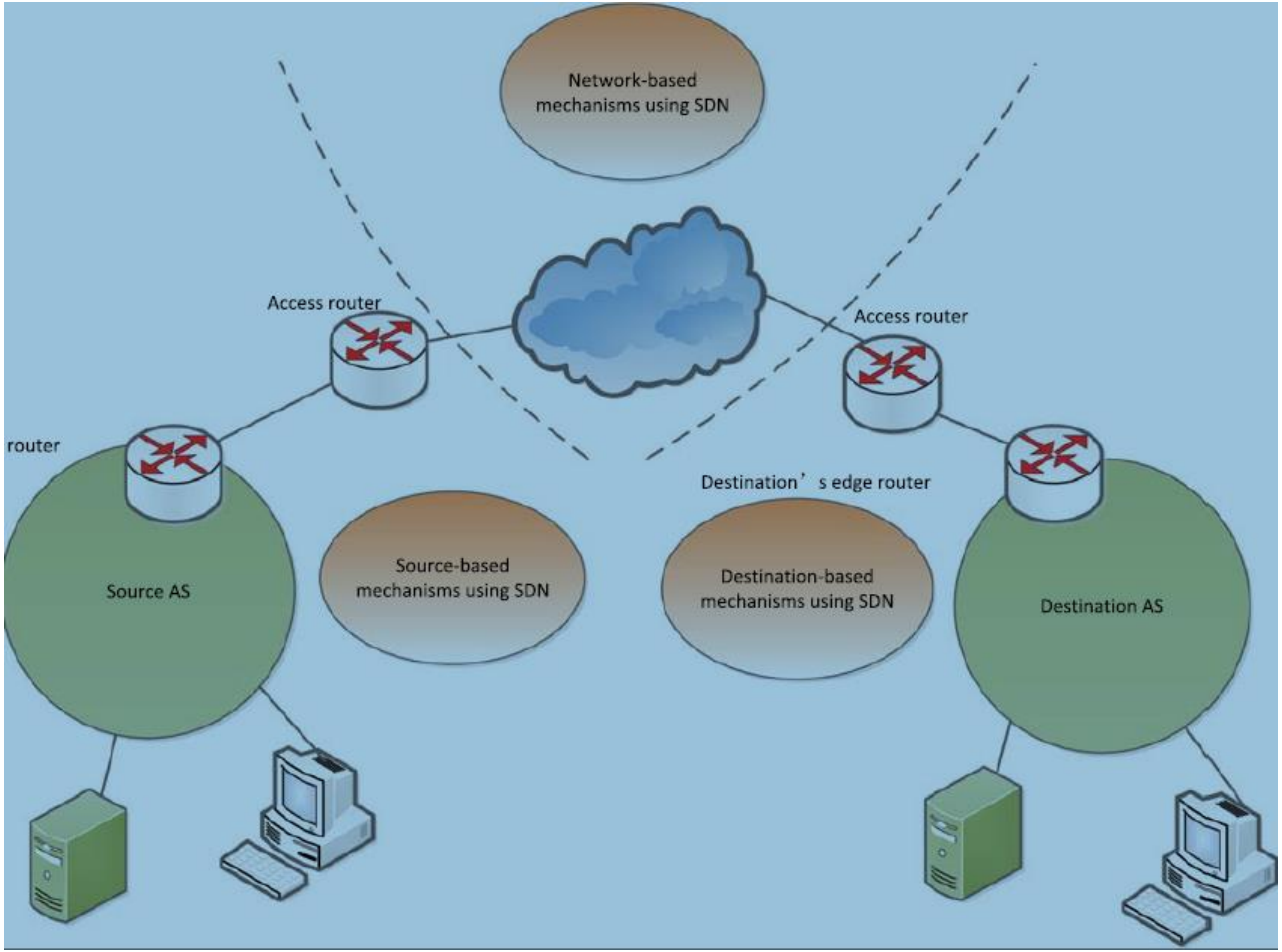
Classification of DDoS Attacks

- Network/transport-level
  - TCP, UDP, ICMP and DNS protocol packets
  - Focus on disrupting legitimate user’s connectivity by exhausting victim network’s bandwidth
- Application-level
  - Focus on disrupting legitimate services by exhausting the server resources(Sockets, CPU, memory, disk/database bandwidth, and I/O bandwidth)

Benefit of SDN

- Separation of the control plane from the data plane
- A logical centralized controller and view of the network
- Programmability of the network by external applications
- Software-based traffic analysis
- Dynamic updating of forwarding rules and flow abstraction

Using SDN to mitigate DDoS attacks



Qiao Yan, F. Richard Yu, Qingxiang Gong, Jianqiang Li, “Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges”

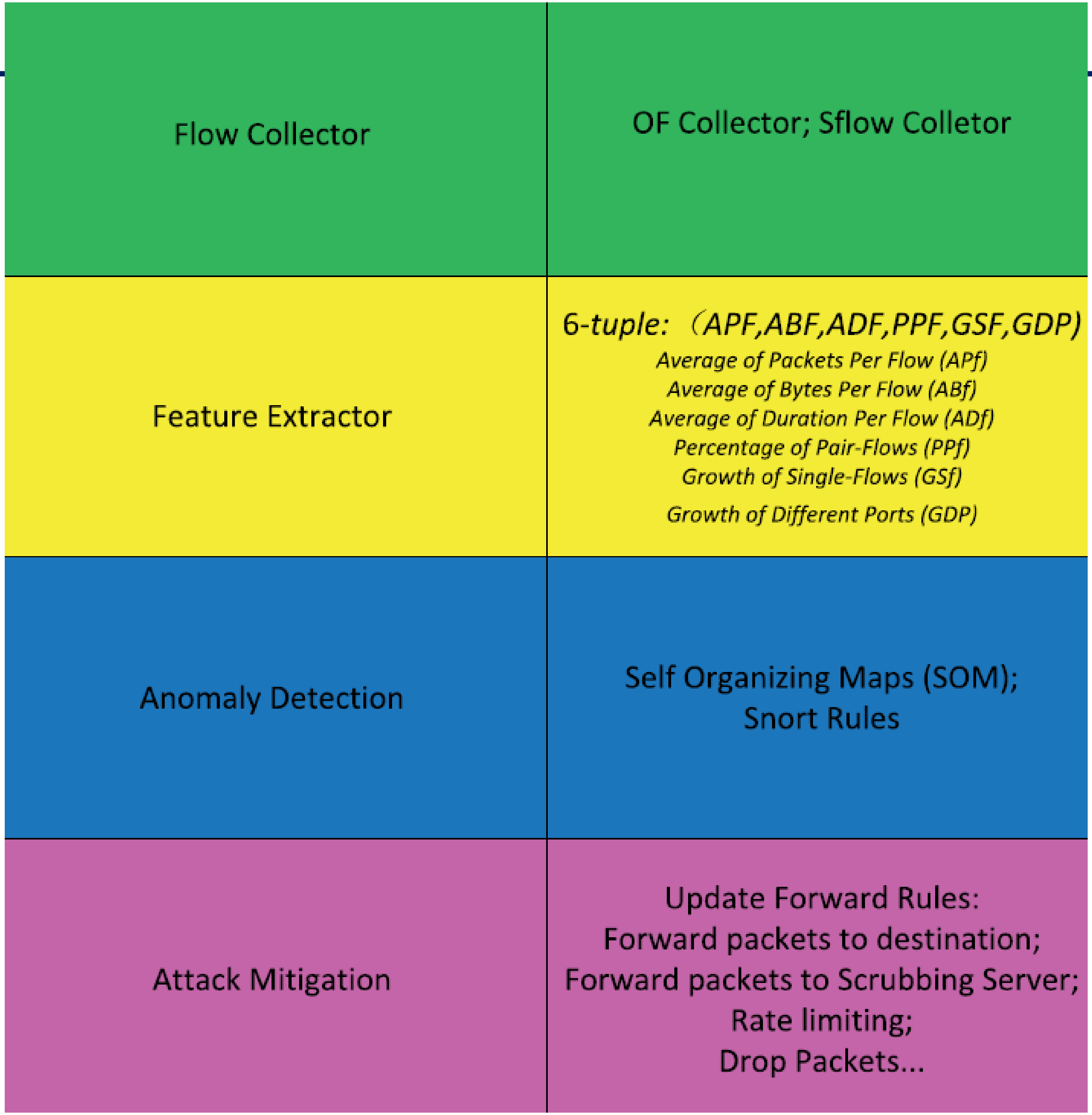
Source-Based Mechanisms

- Filter the malicious packet
- Detect anomaly traffic
- Validate the source IP address near the ingress of network

Network-Based Mechanisms

- Flow Collector module
  - Periodically requesting flow entries from all Flow Tables of Open Flow Switches
- Feature Extractor module
  - Average of Packets per flow (APf), Average of Bytes per flow (ABf), Average of Duration per flow (ADf), Percentage of Pair-flows (PPf), Growth of Single-flow (GSf), and Growth of Different Ports (GDP).
- Classifier module
  - Analyzes whether or not a given 6-tuple corresponds to a DDoS flooding attack or to legit imate traffic (Self Organizing Maps (SOMs))

Network-Based Mechanisms



Flow Collector	OF Collector; Sflow Colletor
Feature Extractor	6-*tuple:* *（APF,ABF,ADF,PPF,GSF,GDP)* *Average of Packets Per Flow (APf)* *Average of Bytes Per Flow (ABf)* *Average of Duration Per Flow (ADf)* *Percentage of Pair-Flows (PPf)* *Growth of Single-Flows (GSf)* *Growth of Different Ports (GDP)*
Anomaly Detection	Self Organizing Maps (SOM); Snort Rules
Attack Mitigation	Update Forward Rules: Forward packets to destination; Forward packets to Scrubbing Server; Rate limiting; Drop Packets...

Qiao Yan, F. Richard Yu, Qingxiang Gong, Jianqiang Li, “Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges”

Destination-Based Mechanisms

- IP traceback
  - Find the origins and paths of attacking traffic
  - Most approaches for IP traceback are hard to be deployed in the Internet because of deployment difficulties

Possible DDoS Attacks on SDN

- Application layer DDoS attacks
- Control layer DDoS attacks
- Infrastructure layer DDoS attacks

Application layer DDoS attacks

- Attack some applications
- Northbound API
- As isolation of applications or resources of SDN is not well solved, DDoS attacks on one application can affect other applications

Control layer DDoS attacks

Attacking controller, northboundAPI, southboundAPI, westboundAPI or eastbound API

1. Data plane will typically ask the control plane to obtain flow rules when the data plane sees new network packets
2. With a large volume of network traffic, sending the packet to the controller would occupy high bandwidth

Infrastructure layer DDoS attacks

- Information is transmitted to the controller
- The packet itself must be stored in node memory until the flow table entry is returned
- The memory element of the node can be a bottleneck due to high cost, an attacker could potentially overload the switch memory

05 Conclusions

Conclusions

- SDN-based cloud is still in its concept phase
- Summarized the difficulty in defeating Unauthorized access, Data leakage, Data modification, Malicious applications, Configuration issues, DDoS attacks in cloud computing environments

參考資料

[1] Hongyan Cui, Zunming Chen, Longfei Yu, Kun Xie, Zongguo Xia, “Authentication Mechanism for Network Applications in SDN Environments”

2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)

[2] Chanhee Lee, Changhoon Yoon, Seungwon Shin, Sang Kil Cha KAIST, “INDAGO: A New Framework For Detecting Malicious SDN Applications ”

2018 IEEE 26th International Conference on Network Protocols

參考資料

[3] Jennia Hizver, “Taxonomic Modeling of Security Threats in Software Defined Networking”

https://www.blackhat.com/docs/us-15/materials/us-15-Hizver-Taxonomic-Modeling-Of-Security-Threats-In-Software-Defined-Networking-wp.pdf

BlackHat Conference August 5-6, 2015

[4] Shao Ying Zhu, Sandra Scott-Hayward, Ludovic Jacquin, Richard Hill, “Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications”

參考資料

[5] Qiao Yan, F. Richard Yu, Qingxiang Gong, Jianqiang Li, “Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges”

IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 18, NO. 1, FIRST QUARTER 2016

[6] “Principles and Practices for Securing Software-Defined Networks,”ONF

https://www.opennetworking.org/wp-content/uploads/2014/10/Principles_and_Practices_for_Securing_Software-Defined_Networks_applied_to_OFv1.3.4_V1.0.pdf

參考資料

[7] SDN Security Attack Vectors and SDN Hardening

https://www.networkworld.com/article/2840273/sdn-security-attack-vectors-and-sdn-hardening.html

[8] Safeguarding Network Slices with a Rich Set of Security VNFs and Services

https://www.fortinet.com/solutions/mobile-carrier/securing-5g-innovation/network-slicing.html

參考資料

[9] 5G Network Slicing and Security

https://sdn.ieee.org/newsletter/january-2018/5g-network-slicing-and-security

[10]Network slicing security: 4 key concerns

https://www.rcrwireless.com/20180530/network-infrastructure/network-slicing-security-4-key-concerns