Applying F(I)MEA Technique for SDN/OpenFlow Security Analysis

Green Kim

greenkim@konkuk.ac.kr

Contents

1. Introduction

1.1 Motivation

1.2 Related Works Analysis

1.2.1 OpenFlow: A Security Analysis

1.2.2 OpenFlow Vulnerability Assessment

1.2.3 Towards Secure and Dependable Software-Defined Networks

1.2.4 Evaluation of Security Vulnerabilities by Using ProtoGENI as a Launchpad

1. Introduction (1/2)

- SDN is rapidly moving from vision to reality
  - Host of SDN-enabled devices in development and production
  - The combination of separated **control** and **data plane functionality** and **programmability** in the network have found their commercial application in cloud computing and virtualization technology
- The SDN architecture can be exploited to enhance network security
  - Provision of highly reactive security monitoring, analysis and response time
  - The **central controller** is key to this system
    - Deploy traffic analysis or anomaly-detection

%SDN : Software Defined Networks

1. Introduction (2/2)

- However, the same attributes of centralized control and programmability associated with the SDN platform introduce network security challenges
  - An increased potential for Denial-of-Service attacks
    - Centralized controller and flow-table limitation in network device
  - Another issue of concern based on open programmability of the network is trust
    - Between applications and controllers
    - Between controllers and network devices

- An Analysis technique for SDN security is required

1.1 Motivation (1/3)

- OpenFlow is a standardized protocol which implements the notion of SDN
  - The separation of the network control plane from the data plane
  - A Logically centralized controller
- OpenFlow is used for the interaction between a network switch, constituting the data plane, and a controller, constituting the control plane
  - The switch performs packet forwarding using one or more flow tables
    - The flow rules are installed on the switch by the controller
  - The controller can choose to install flow rules *proactively* on its own accord, or *reactively* in response to a notification by the switch regarding a packet failing to match existing rules

1.1 Motivation (2/3)

- OpenFlow has seen widespread deployment on production networks and its adoption is constantly increasing
- Although openness and programmability are primary features of OpenFlow, Security is of core importance for real-world deployment
- A number of Security Analysis have recently been performed
  - Security Analysis have performed that the altered elements relationship between elements in the SDN framework introduce new vulnerabilities
    - Vulnerabilities were not present before SDN

1.1 Motivation (3/3)

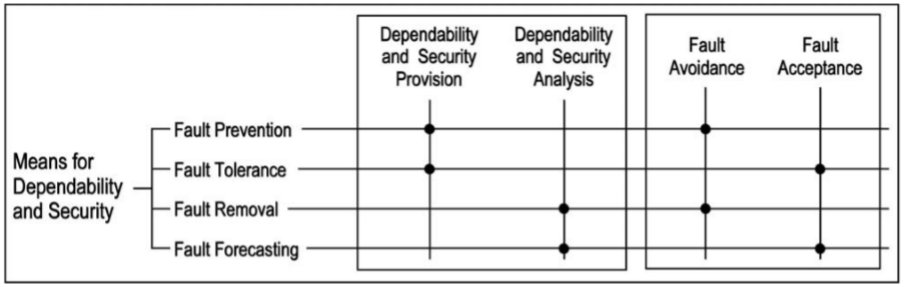


Fig. 21. Groupings of the means for dependability and security.

- When focusing on security, analysis is called security evaluation
- Fault Forecasting
  - qualitative, or ordinal, evaluation that aims to identify, classify, and rank the failure modes, or the event combinations (component failures or environmental conditions) that would lead to system failures
    - qualitative evaluation : e.g., failure mode and effect analysis
  - quantitative, or probabilistic, evaluation that aims to evaluate in terms of probabilities the extent to which some of the attributes are satisfied; those attributes are then viewed as measures

Basic Concepts and Taxonomy of Dependable and Secure Computing. 2004

1.2 Related Works Analysis

1.2.1 OpenFlow: A Security Analysis (2013)

1.2.2 OpenFlow Vulnerability Assessment (2013)

1.2.3 Towards Secure and Dependable Software-Defined Networks (2013)

1.2.4 Evaluation of Security Vulnerabilities by Using ProtoGENI as a Launchpad (2011)

1.2 Related Works Analysis

1.2.1 OpenFlow: A Security Analysis (2013)

→ Evaluation of Possibility

1.2.2 OpenFlow Vulnerability Assessment (2013)

→ Evaluation of Possibility

1.2.3 Towards Secure and Dependable Software-Defined Networks (2013)

→ High-level analysis of the overall security of SDN

1.2.4 Evaluation of Security Vulnerabilities by Using ProtoGENI as a Launchpad (2011)

→ Evaluation of Possibility

%**possibility** of any event is always 1 or 0 i.e. 'yes' or 'no'.
If an event is possible, how likely will its occurrence be, under a given situation is **probability**

1.2.1 OpenFlow : A Security Analysis (1/2)

- This research Combines two modeling techniques
  - Microsoft's STRIDE methodology
    - STRIDE methodology is used to construct a model of and OpenFlow system and enumerate its potential vulnerabilities
    - **S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, and **E**levation of Privilege
    - The result of this analysis is a set of system component and vulnerability pairs
  - Attack trees
    - Attack trees is used to explore how an identified vulnerability could be exploited
    - The root of an attack tree is an attacker's ultimate objective

1.2.1 OpenFlow : A Security Analysis (2/2)

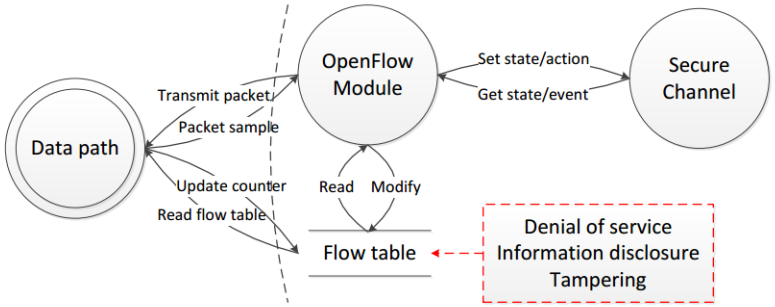


Fig. 1. Simplified DFD for an OpenFlow switch, showing relevant vulnerabilities

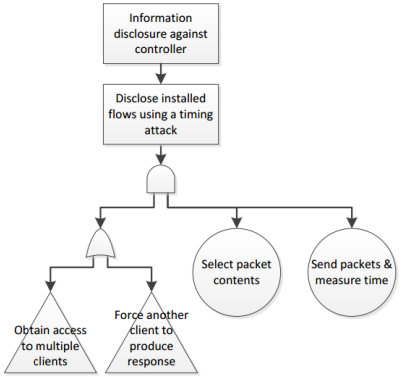


Fig. 2. Simplified attack (sub-)tree showing an Information Disclosure attack against an OpenFlow controller

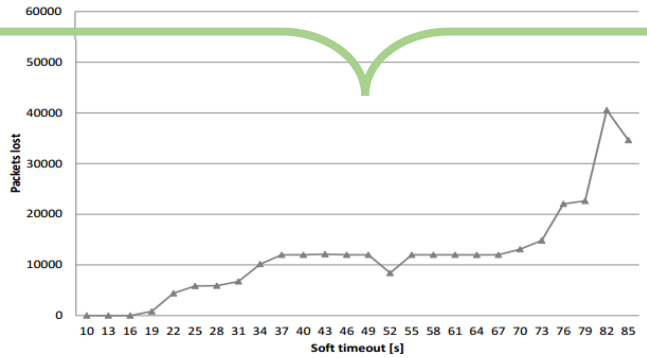


Fig. 5. Test with data link at 100 Mbps, 10 ms delay, control link at 100 Mbps, 1 ms delay

OpenFlow : A Security Analysis. 2013.

- Although a number of mitigation techniques are proposed in this paper, these techniques are not proven in the work

1.2.2 OpenFlow Vulnerability Assessment

- This research suggests the possibility of attacks

Flow Table Verification A full TLS implementation could increase security of the messages between switches and controllers; however, it wouldn't help detect switches that erroneously alter rules. Also, tracking the state changes of the flow-table for each switch by recording all of the *flow-removed* messages generated by switches requires extra logic on the controller, especially in the case of temporary network outage recovery. This mismatch between the controller's idea of the network's rule-state and the actual rule-state can lead to an access-control failure, a network outage, or other unexpected behavior. Currently, the only way to verify the rules is by dumping and inspecting the flow tables from each switch. This can be quite computationally costly for the switches and the controller(s).

OpenFlow Vulnerability Assessment. 2013.

1.2.3 Towards Secure and Dependable Software-Defined Networks

- This research presents a high-level analysis of the overall security of SDN
- They conclude that due to the nature of the centralized controller and the programmability of the network, net threats are introduced requiring new responses

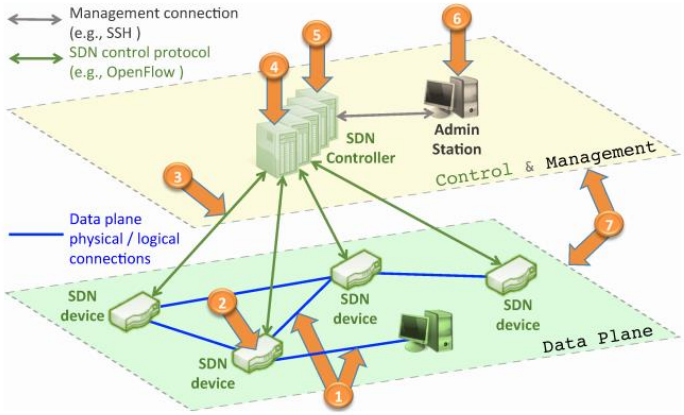


Figure 1: SDN main threat vectors map

Table 1: SDN specific vs non-specific threats

Threats	Specific to SDN?	Consequences in SDN
Vector 1	no	can be a door for DoS attacks
Vector 2	no	but now the impact is potentially augmented
Vector 3	yes	communication with logically centralized controllers can be explored
Vector 4	yes	controlling the controller may compromise the entire network
Vector 5	yes	malicious applications can now be easily developed and deployed on controllers
Vector 6	no	but now the impact is potentially augmented
Vector 7	no	it is still critical to assure fast recovery and diagnosis when faults happen

Towards Secure and Dependable Software-Defined Networks. 2013.

1.2.4 Evaluation of Security Vulnerabilities by Using ProtoGENI as a Launchpad

- The authors discovered that numerous attacks between users of the testbed along with malicious propagation and flooding attacks to the wider internet were possible when using the ProtoGENI network

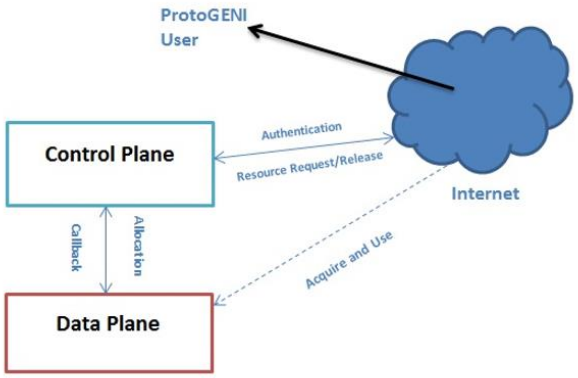


Fig. 1. ProtoGENI Control Plane and Data Plane

```
[lidawei@nodew1 src]$ sudo /usr/local/bin/netwox 33 -d ath0 -a 0C:0C:0C:0C:0C:0C
 -b 00:17:9A:C3:65:24 -c 2054 -e 2 -f 0C:0C:0C:0C:0C:0C -g "10.1.1.3" -h 00:17:9
A:C3:65:24 -i 10.1.1.2
Ethernet_____________________________________________________.
| 0C:0C:0C:0C:0C:0C->00:17:9A:C3:65:24 type:0x0806           |
|____________________________________________________________|
ARP Reply____________________________________________________.
| this answer : 0C:0C:0C:0C:0C:0C 10.1.1.3                   |
| is for      : 00:17:9A:C3:65:24 10.1.1.2                   |
|____________________________________________________________|
```

Fig. 4. Launch ARP Cache Poisoning

```
[lidawei@nodew1 ~]$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
control-router.emulab.n  ether   00:B0:8E:84:69:34   C                     eth4
nodew2-lan0              ether   0C:0C:0C:0C:0C:0C   C                     ath0
```

Fig. 5. ARP Cache after Attack

%ProtoGENI : Prototype control framework implementation of GENI (Global Environment for Network Innovations)

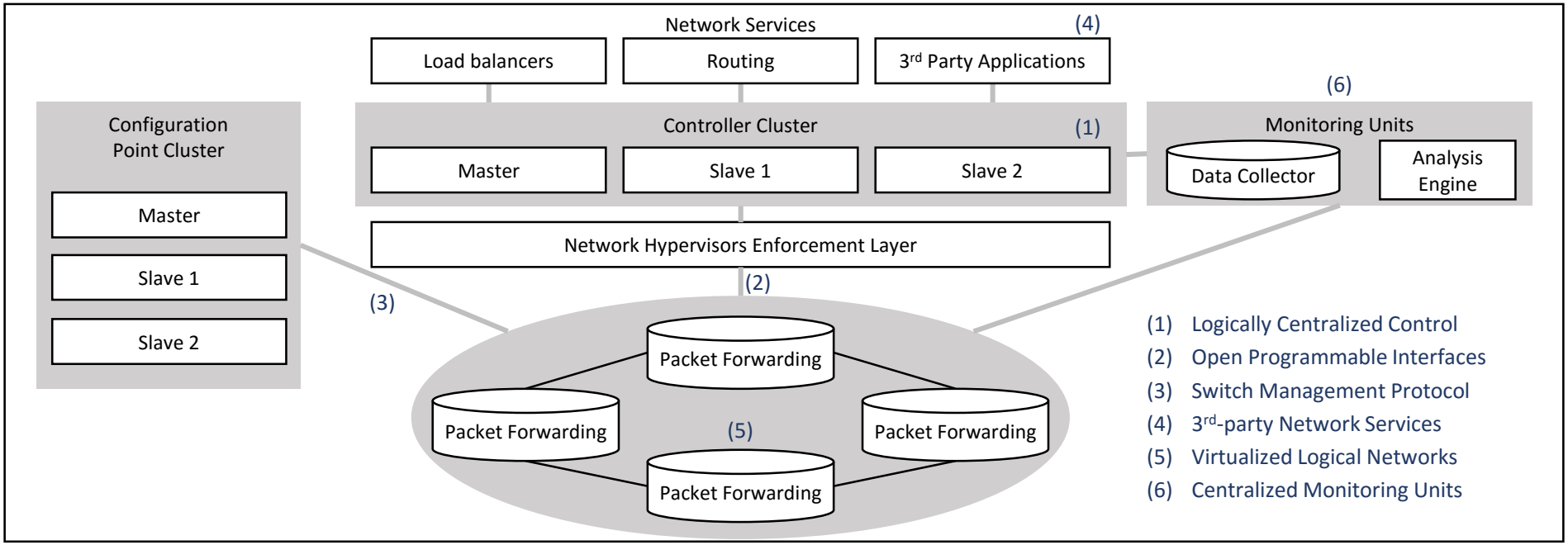
2. Security Issues associated with the SDN (1/4)

- The basic properties of a security communications network
  - Confidentiality
  - Integrity
  - Availability of information
  - Authentication
  - Non-repudiation

→ Secure data, network assets and communications transactions

2. Security Issues associated with the SDN (2/4)

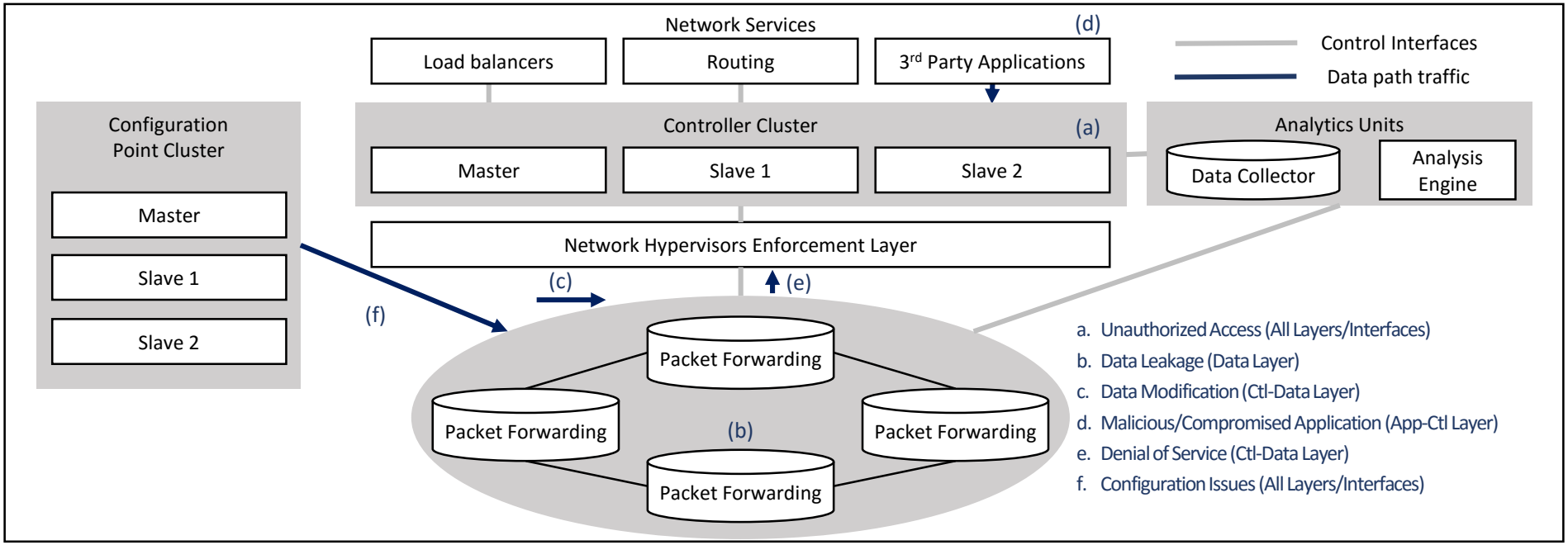
- SDN Characteristics



'A Survey of Security in Software Defined Networks', IEEE Communications Surveys & Tutorials, 2015.

2. Security Issues associated with the SDN (3/4)

- SDN Potential Attacks and Vulnerabilities



'A Survey of Security in Software Defined Networks', IEEE Communications Surveys & Tutorials, 2015.

2. Security Issues associated with the SDN (4/4)

- Categorization of Security Issues

Security Issue/Attack	SDN Layer Affected or Targeted				
	Application Layer	App-Ctl Interface	Control Layer	Ctl-Data Interface	Data Layer
Unauthorized Access e.g. • Unauthorized Controller Access/Controller Hijacking • Unauthorized/Unauthenticated Application	 X	 X	X X	X	X
Data Leakage e.g. • Flow Rule Discovery (Side Channel Attack on Input Buffer) • Credential Management (Keys, Certificates for each Logical Network) • Forwarding Policy Discovery (Packet Processing Timing Analysis)			 X	 X	X X X
Data Modification e.g. • Flow Rule Modification to Modify Packets (Man-in-the-middle attack)			X	X	X
Malicious/compromised Applications e.g. • Fraudulent Rule Insertion	X	X	X		
Denial of Services e.g. • Controller-Switch Communication Flood • Switch Flow Table Flooding			X	X	X X
Configuration Issues e.g. • Lack of TLS(or other Authentication Technique) Adoption • Policy Enforcement • Lack of Secure Provisioning	X X X	X X X	X X X	X X	X X
System Level SDN Security e.g. • Lack of Visibility of Network State			X	X	X

'SDN Security: A Survey', IEEE SDN for Future Networks and Services, 2013.

3. Failure (Intrusion) Modes and Effect Analysis

3.1 Taxonomy of issues

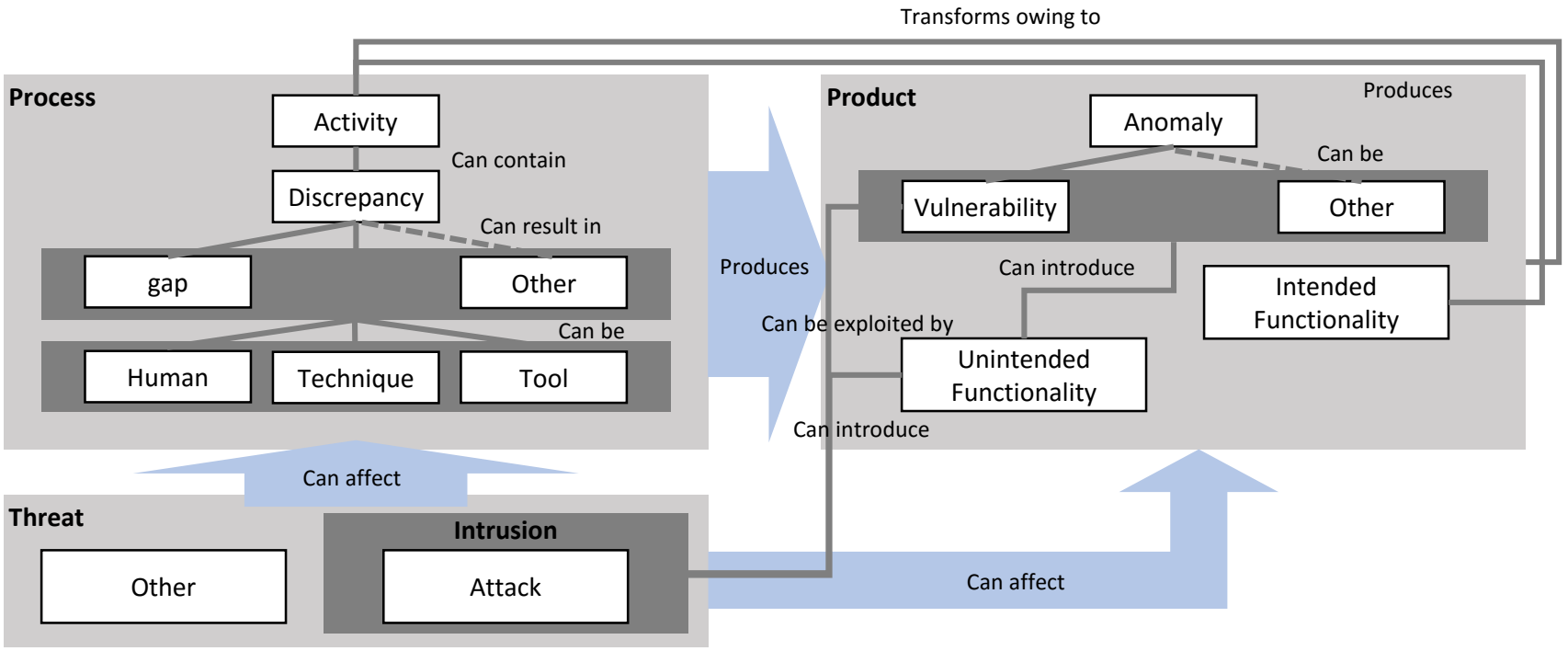
3.2 Analysis Technique

3.1 Taxonomy of issues (1/2)

- The key idea in security assessment is using **process-product approach**
  - In determining the **possible problems**, inconsistencies during **process implementation** and obtaining of the **products**
  - One of the fundamental concepts behind the idea of the approach is the concept of '**gap**'
    - 'gap' could be defined as **a set of discrepancies** of any single process that can introduce some **anomalies** (e.g. **vulnerabilities**) in a product and/or cannot reveal (and eliminate) existing anomalies in a product

3.1 Taxonomy of issues (2/2)

- Process-Product approach



"Cyber Security Lifecycle and Assessment Technique for FPGA-based I&C systems", Design & Test Symposium, 2013

3.2 Analysis Technique

- Each '**gap**' should be represented in a form of formal description
  - To perform the description, the most convenient is **IMECA** technique
    - **Intrusion Modes and Effects Criticality Analysis**
    - **Modification to FMECA** technique that takes into account possible intrusions into the system
    - During the Security Assessment, IMECA can be used in addition to standardized FMECA for **safety-related domains**
      - each **vulnerability** can become a **failure** in a case of **intrusion** into such systems
  - Each identified gap can be represented by a single local IMECA table and each discrepancy inside the gap can be represented by a single row in that local IMECA table

4. Case study of F(I)MEA Technique (1/3)

- Based on Categorization of SDN Security Issues from **'SDN Security: A Survey'**, it is possible to choose several types of intrusions
  - **Controller hijacking**
  - **Man-in-the-middle**
  - **Denial of Service**
- Following table shows application of IMECA technique for analysis of theses intrusions

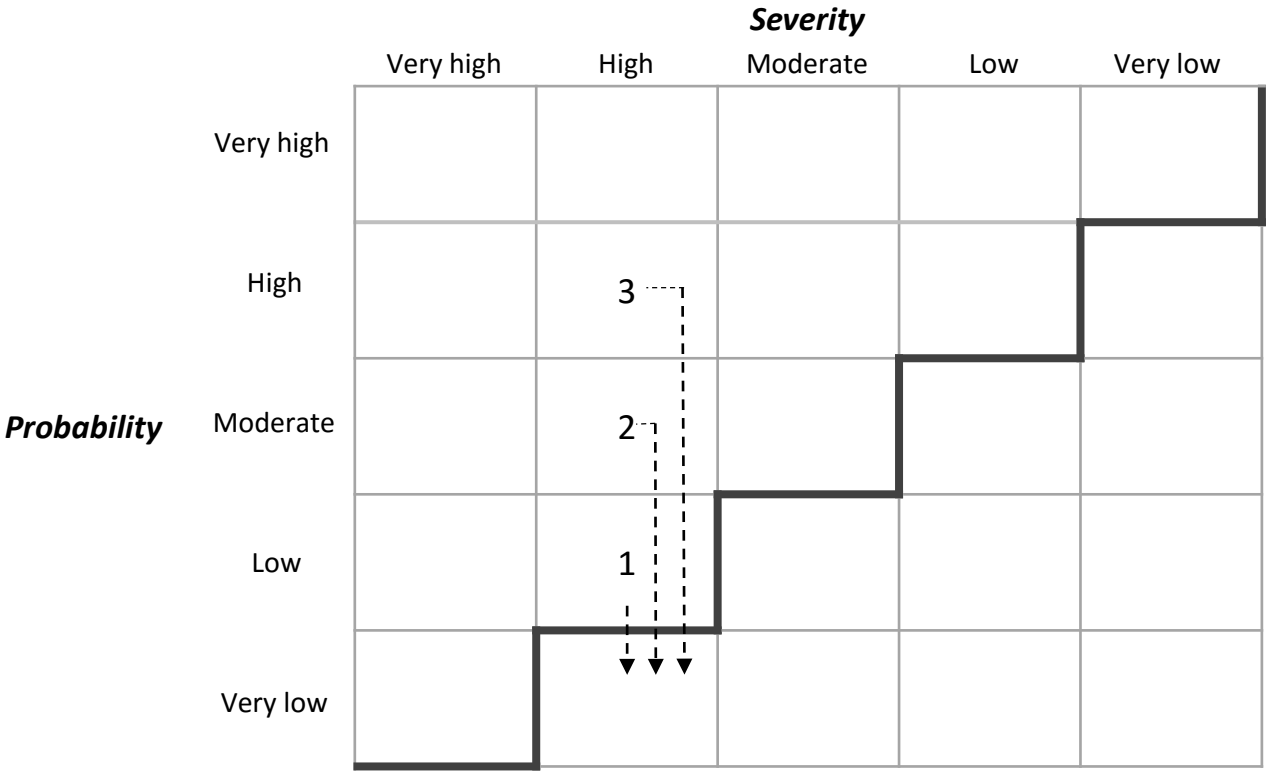
4. Case study of F(I)MEA Technique (2/3)

- Intrusion Modes and Effects Criticality Analysis

GAP No	Attack mode	Attack nature	Attack cause	Occurrence Probability	Effect Severity	Type of effects				
						Application Layer	App-Ctl Interface	Control Layer	Ctl-Data Interface	Data Layer
1	Controller hijacking	Active	• Weak authentication	Low	High	-	-	• Gain access to network resource • Manipulate the network operation		
2	Main-in-the middle	Active	• Weak Authentication • Weak confidentiality	Moderate	High	-	-	• Have control over the entire system • Insert/Modify flow rules in the network devices • Allow packets to be steered through the network to the attacker's advantage		
3	Denial of Service	Active	• Weak protection • Resource limitation of flow table	High	High	-	-	• Lead to fraudulent rule insertion and rule modification		

4. Case study of F(I)MEA Technique (3/3)

- Criticality matrix (Adapted from ISO 31000:2009)
  - Each of the numbers inside the matrix row number of IMECA table
  - Acceptable values of risks are below the diagonal



5. Conclusion

- A secure SDN does not exist
  - Hidden vulnerabilities are still possible in SDN
  - Security Assessment should be perceived as a repeatable process
- Assurance of SDN security is not possible without taking in to account all specific features of technologies in use
  - In addition to improving SDN, it is necessary to focus on developing rules and best practices that establish and maintain security of SDN

6. Future Works

- Compare the IMECA Assessment technique with other methodology such as STRIDE
- Compare SDN Security between various Controllers
  - ONOS
  - OpenDaylight
  - ROSEMARY
  - Ryu
  - SE-Floodlight
- Research and Categorize Security solutions and SDN Security Enhancement
- Recommend Best Practices

References

1. Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, Carl Landwehr. “Basic Concepts and Taxonomy of Dependable and Secure Computing”. Jan 2004.
2. M. Coughlin. “A Survey of SDN Security Research”.
3. S. Scott-Hayward, S. Natarajan, S. Sezer “A Survey of Security in Software Defined Networks”. Communications Surveys & Tutorials, IEEE, 2015.
4. S. Scott-Hayward, G. O'Callaghan and S. Sezer "SDN security: A survey", Future Networks and Services, IEEE, 2013.
5. R. Kloeti, "OpenFlow: A Security Analysis,“ Available: ftp://yosemite.ee.ethz.ch/pub/students/2012-HS/MA-2012-20- signed.pdf, 2013.
6. Kevin Benton, L. Jean Camp, Chris Small. “OpenFlow vulnerability assessment”, Proceedings of the second ACM SIGCOMM workshop on Hot topics software defined networking. 2013.
7. Diego Kreutz, Fernando M. V. Ramos, Paulo Verssimo, “Towards secure and dependable software-defined networks”, Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. 2013.
8. A. Gorbenko, V. Kharchenko, O. Tarasyuk, A. Furmanov "F(I)MEA- technique of Web Services Analysis and Dependability Ensuring", Lecture Notes in Computer Science, 2006.
9. E. Babeshko, V. Kharchenko, A. Gorbenko, "Applying F(I)MEA-technique for SCADA-based Industrial Control Systems Dependability Assessment and Ensuring", DepCoS-RELCOMEX, 2008.
10. O. Illiashenko, V. Kharchenko, A. Kovalenko, “Cyber Security Lifecycle and Assessment Technique for FPGA-based I&C systems”, Design & Test Symposium, 2013.
11. ISO/IEC 27000, Information technology-Security techniques-Information security management systems-Overview and vocabulary, International Organization for Standardization and International Electrotechnical Commission, 2009.
12. ISO/IEC 27001:2005, Information technology-Security techniques- Information security management systems-Requirements, International Organization for Standardization and International Electrotechnical Commission, 2005.
13. ISO/IEC 27002:2005, Information technology-Security techniques-Code of practice for information security management, International Organization for Standardization and International Electrotechnical Commission, 2005.
14. ISO 31000, Risk Management, Risk assessment techniques, International Organization for Standardization and International Electrotechnical Commission, 2009.

KU KONKUK UNIVERSITY

Thank You