教育部資訊安全人才培育計畫
108年度新型態資安暑期課程 AIS3 2019
Advanced Information Security Summer School
資安實務專題競賽

AIS3 2019

Network Defense Based On SDN

軟體開發安全與軟體工程
Secure Software Development and Engineering

第一小組 Group 1
陳易煒，陳靖泓，黃上豪，~~黃科皓~~

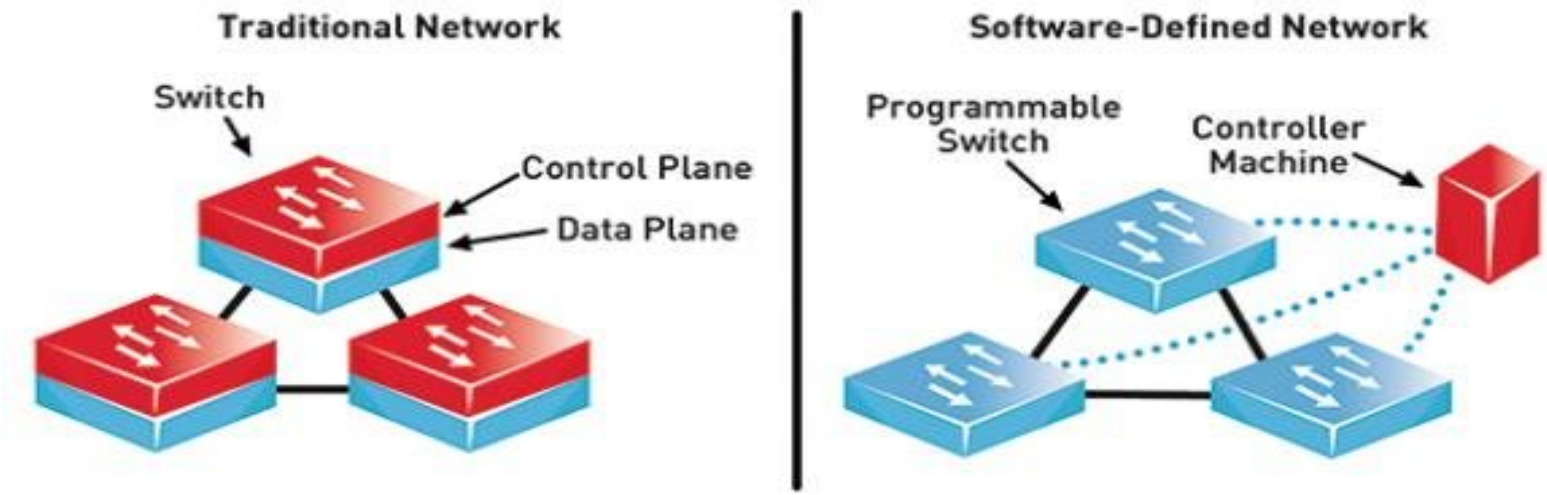
Outline

- SDN Introduction
- General SDN Architecture
- Project Designed SDN Architecture
- Project Demo

SDN Introduction

What is SDN ?

Software Defined Networking (SDN) is a new architecture of network that enables dynamic, programmatically efficient network configuration and it is opposite to the traditional network architecture.



What is SDN ?

SDN's Advantage :

- Efficiency : Optimize existing applications, services and infrastructure
- Scale : Rapidly grow existing applications and services
- Innovation : Create and deliver new types of applications and services and business models

What is SDN ?

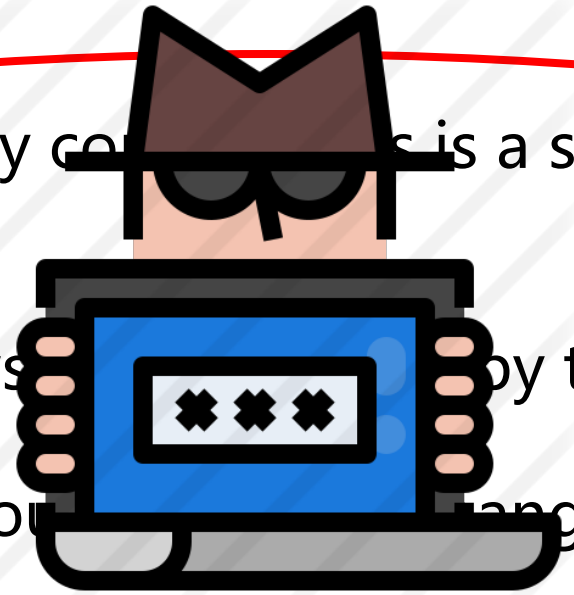
SDN's Disadvantage :

- Programmable : Every components is a software that could under the vulnerability
- Unity : All of the flows are controlled by the controller
- Elastic : Every rules could be easily changed by the controller

What is SDN ?

SDN's Disadvantage :

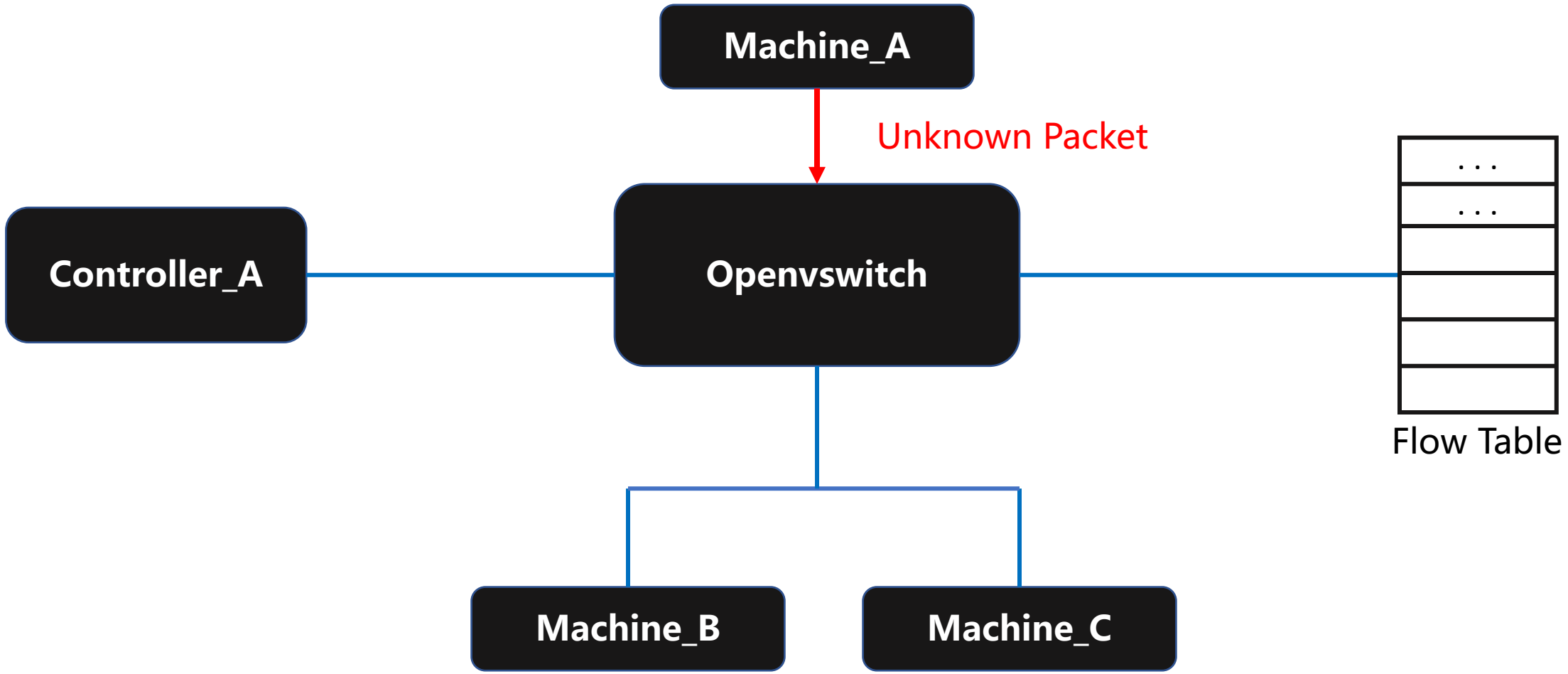
- Programmable : Every co[illegible]s is a software that could under the vulnerability
- Unity : All of the flows [illegible]by the controller
- Elastic : Every rules co[illegible]anged by the controller



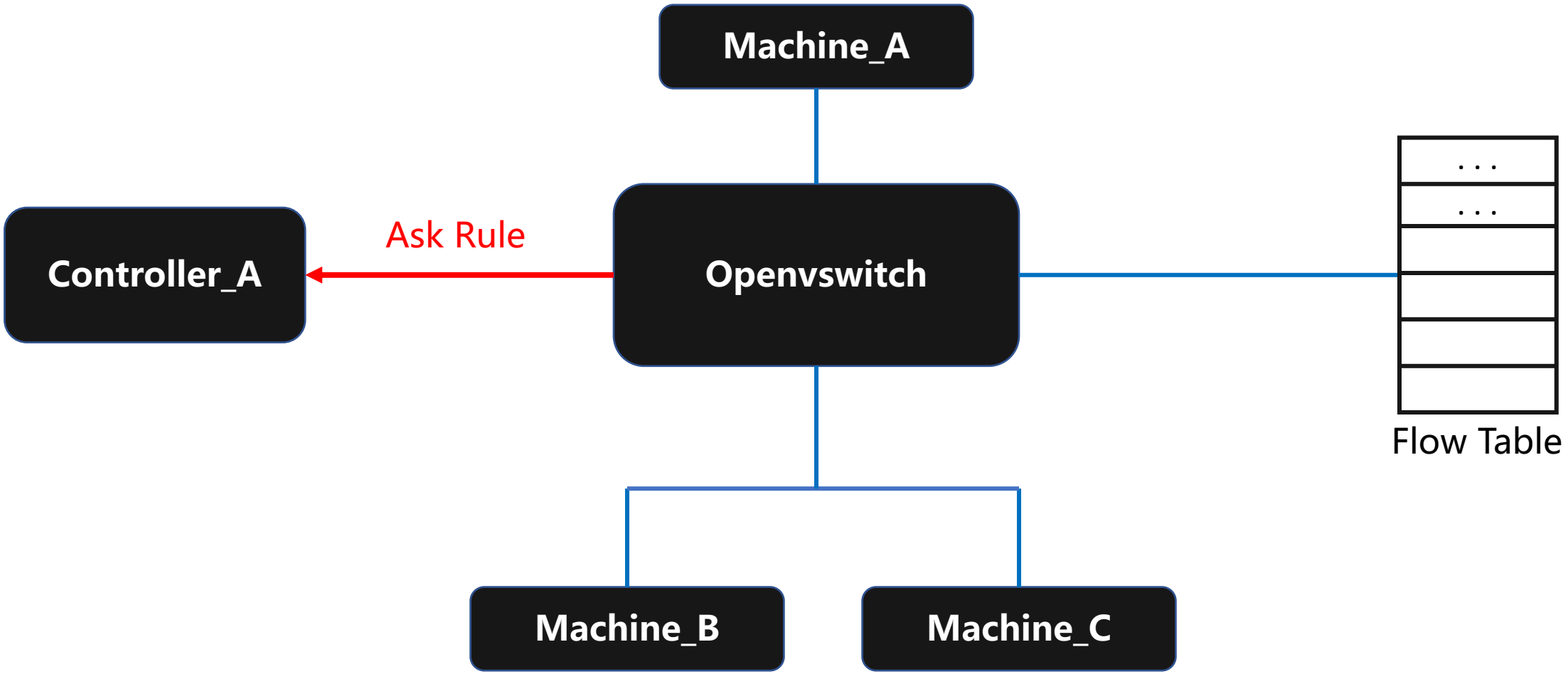
Once the controller is owned by the attacker, the whole network will be controlled.

General SDN Architecture

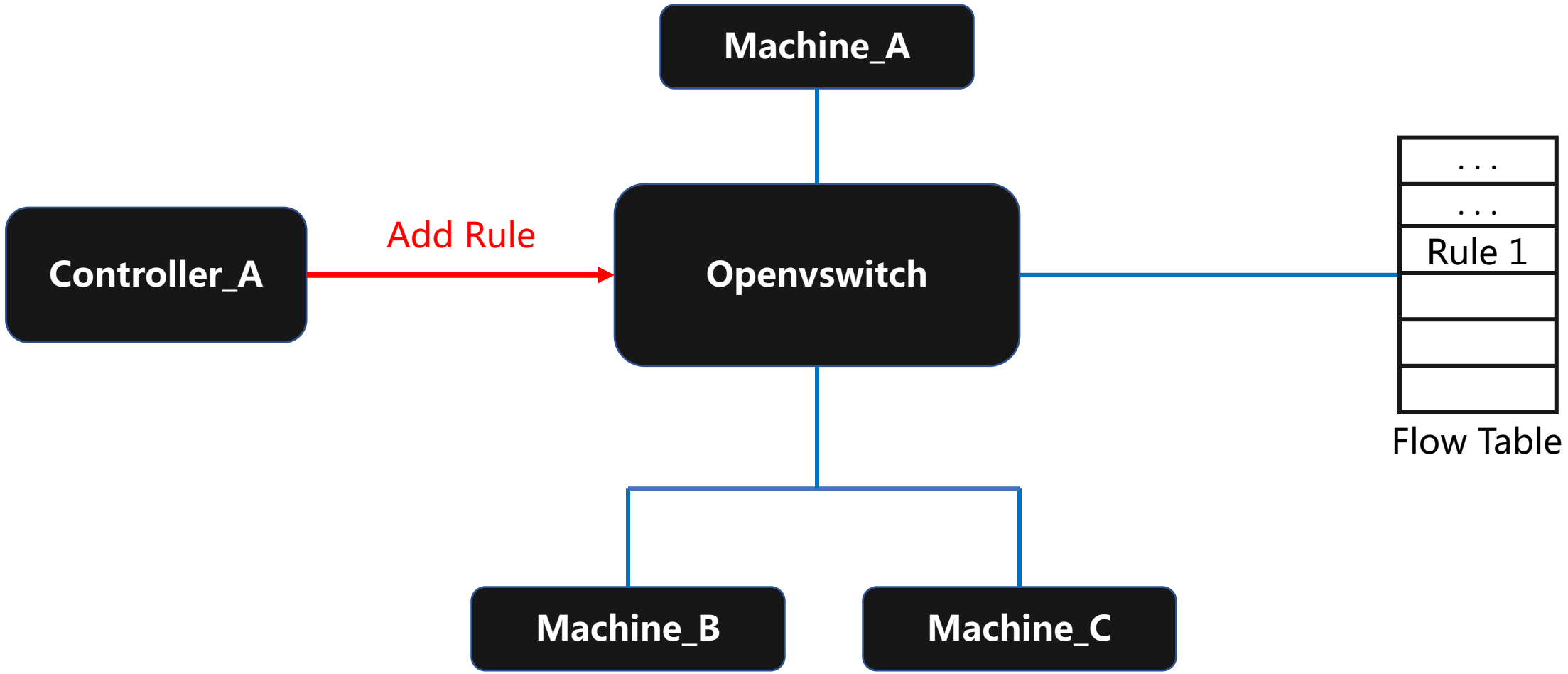
General Arch



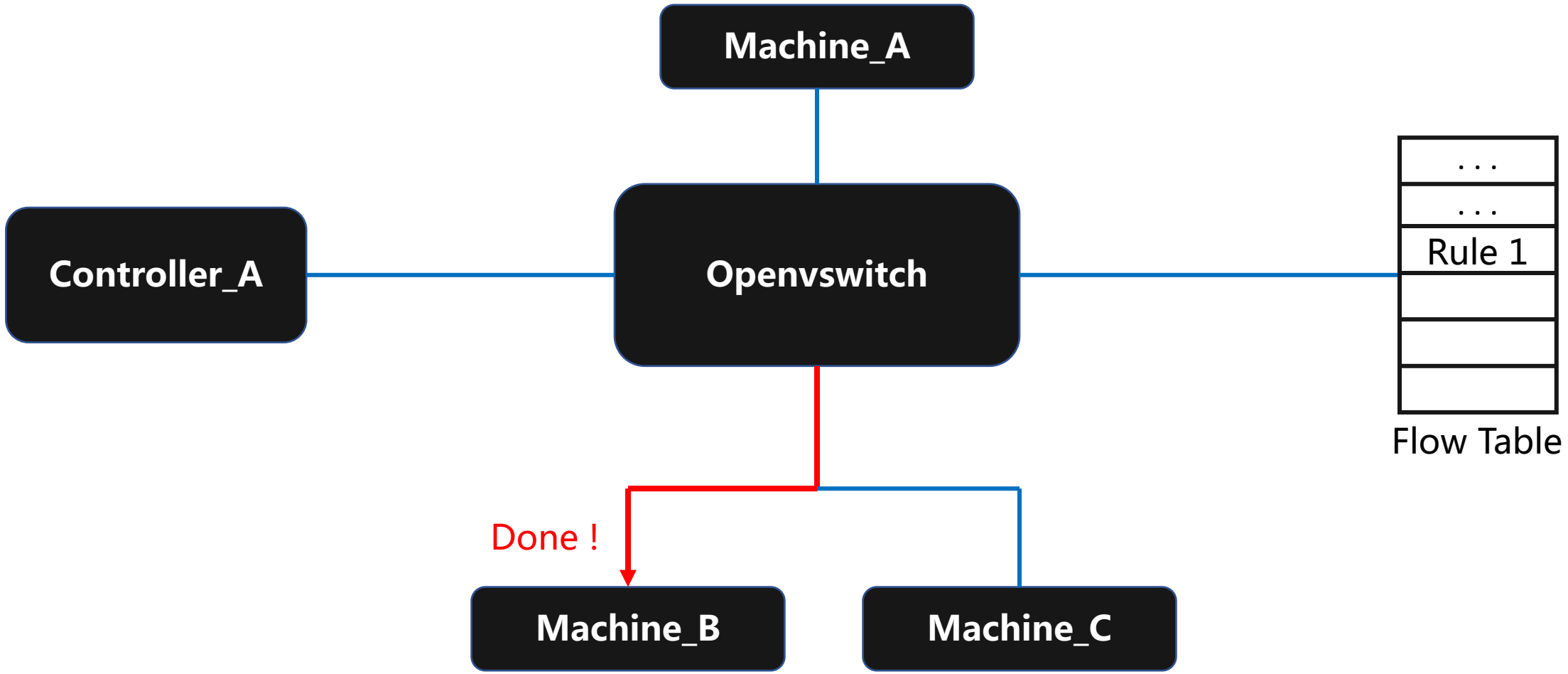
General Arch

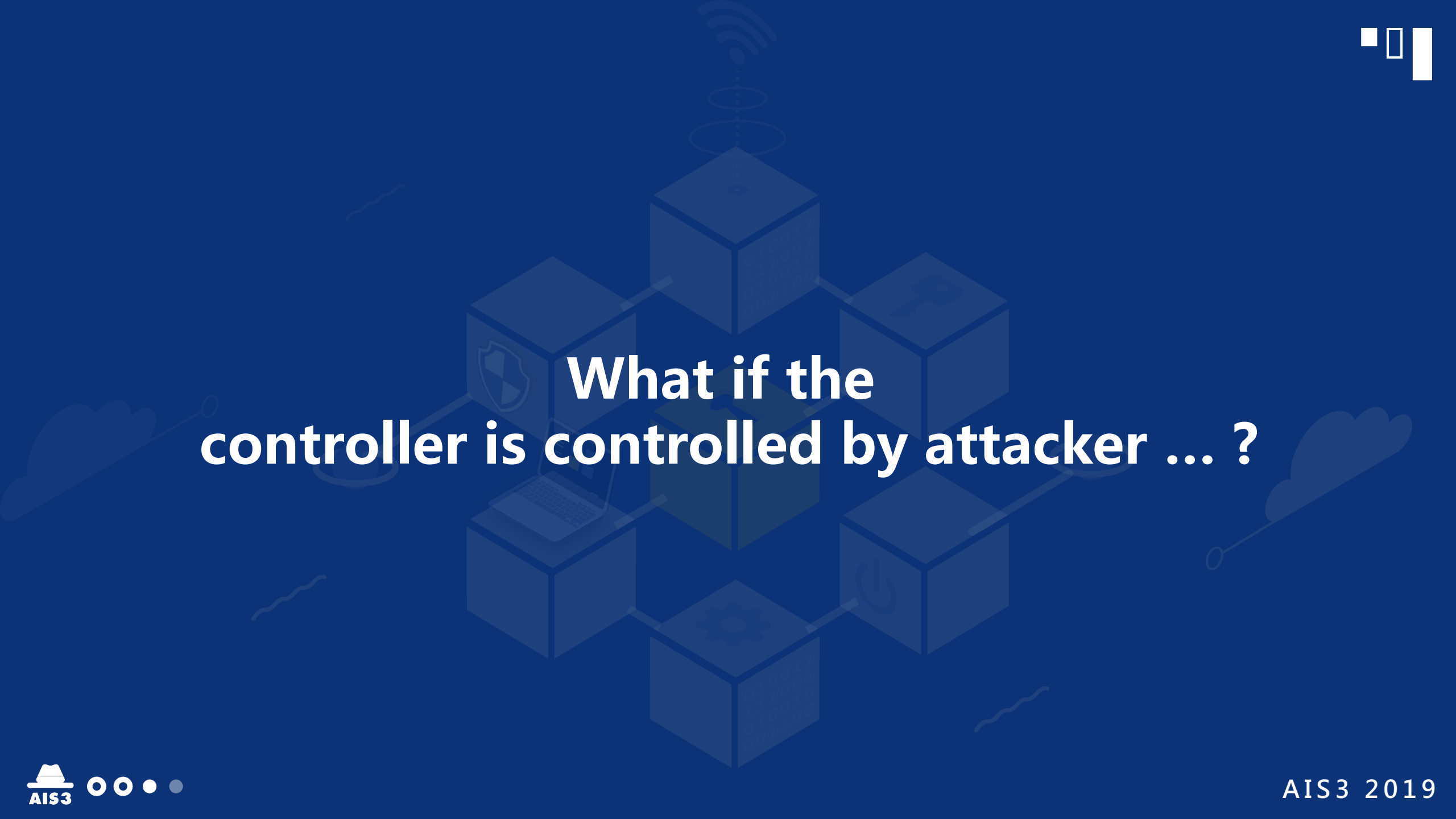


General Arch



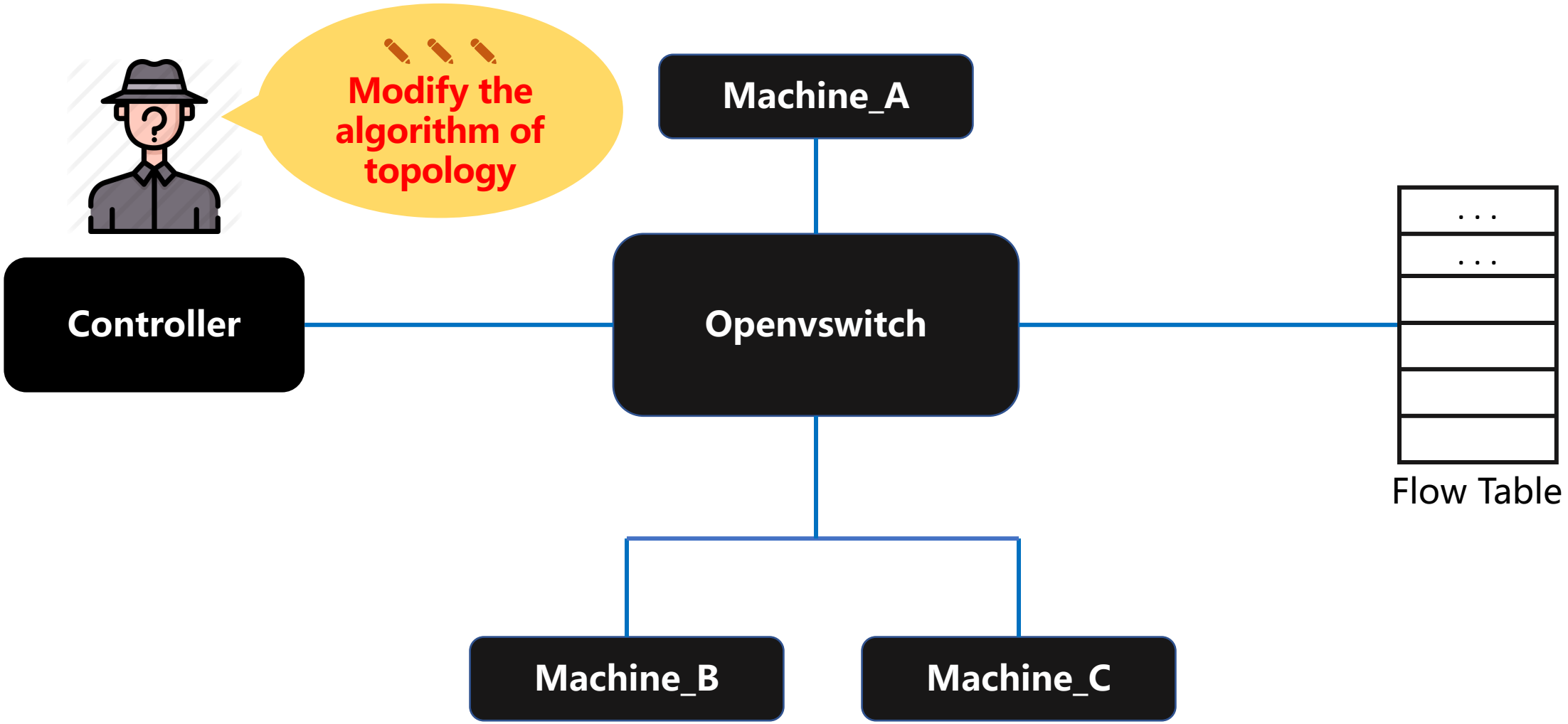
General Arch





What if the controller is controlled by attacker ... ?

General Arch



General Arch

Pwned Controller

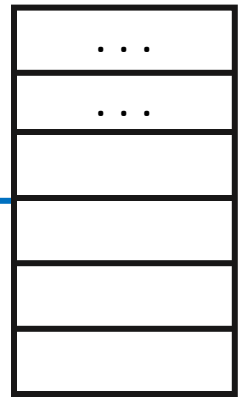
Machine_A

Unknown Packet

Openvswitch

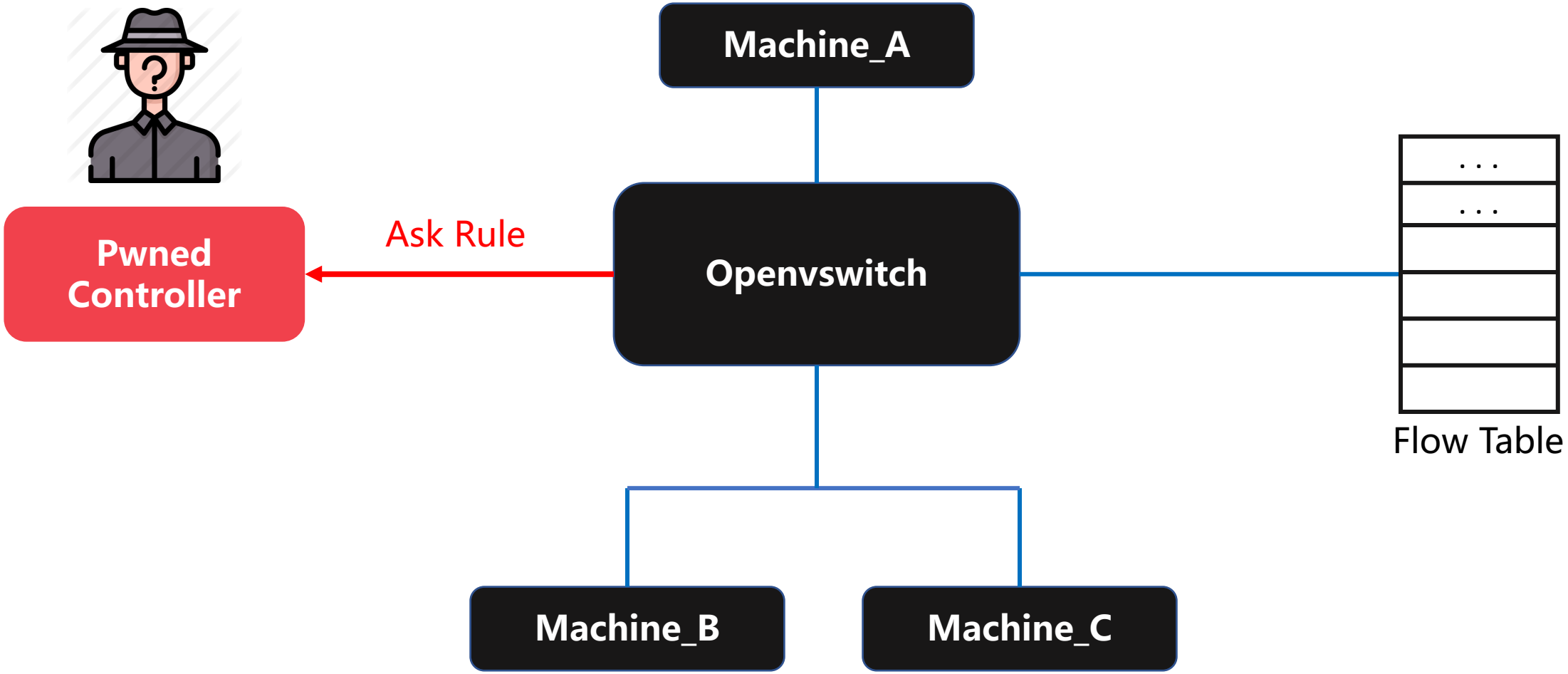
Machine_B

Machine_C



Flow Table

General Arch



General Arch

Pwned Controller

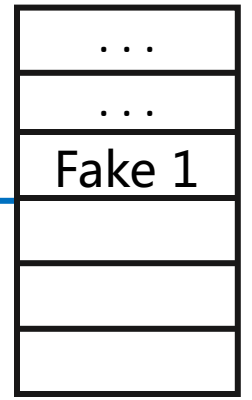
Add Rule

Machine_A

Openvswitch

Machine_B

Machine_C



Flow Table

General Arch

Pwned Controller

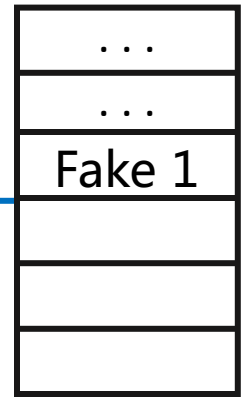
Machine_A

Openvswitch

Machine_B

Machine_C

Attacked !



Flow Table

Project Designed Architecture

Network Attack
By Pwned Controller

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Machine_A

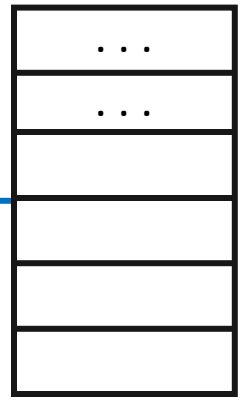
Unknown Packet

Openvswitch

Machine_B

Machine_C

Analysis



Flow Table

If recv(unknown packet) :
Ask Controller_A
Wait for a new rule
...

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Ask Rule

Controller_B
(Priority 2)

Controller_C
(Priority 3)

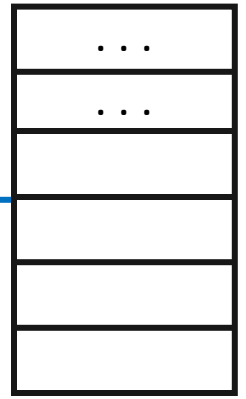
Machine_A

Openvswitch

Machine_B

Machine_C

Analysis



Flow Table

If recv(unknown packet) :
Ask Controller_A
Wait for a new rule
...

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Add Rule

Controller_B
(Priority 2)

Controller_C
(Priority 3)

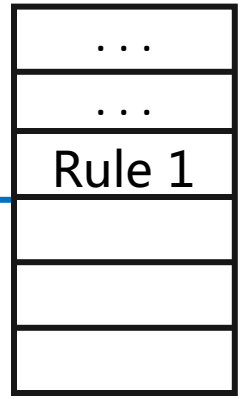
Machine_A

Openvswitch

Machine_B

Machine_C

Analysis



Flow Table

If recv(Rule_1) :
Ask Controller_B
...

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Ask Rule

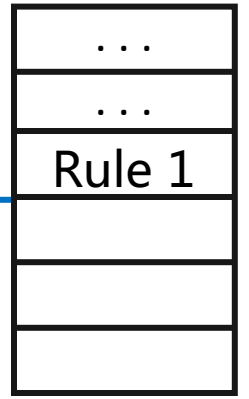
Machine_A

Openvswitch

Machine_B

Machine_C

Analysis



Flow Table

If recv(Rule_1) :
Ask Controller_B
...

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Controller_B
(Priority 2)

Add Rule

Controller_C
(Priority 3)

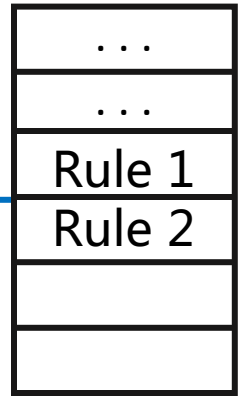
Machine_A

Openvswitch

Machine_B

Machine_C

Analysis



Flow Table

```
If recv( Rule_2 ) :
    If Rule_1 == Rule_2 :
        Forward
    Else
        Ask Controller_C
    ...
```

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

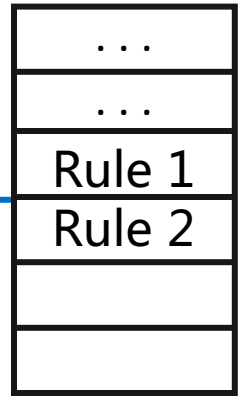
Machine_A

Openvswitch

Machine_B

Machine_C

Analysis



Flow Table

```
If recv( Rule_2 ) :
    If Rule_1 == Rule_2 :
        Forward
    Else
        Ask Controller_C
    ...
```

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Machine_A

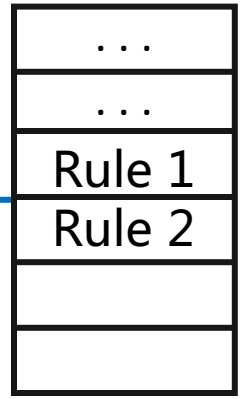
Openvswitch

Machine_B

Machine_C

Analysis

Ask Rule



Flow Table

```
If recv( Rule_2 ) :
    If Rule_1 == Rule_2 :
        Forward
    Else
        Ask Controller_C
    ...
```

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Add Rule

Machine_A

Openvswitch

Machine_B

Machine_C

Analysis

. . .
. . .
Rule 1
Rule 2
Rule 3

Flow Table

Else :
Ask Controller_C
Add new rule
...

Project Arch

Network Attack

Pwned
Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Add Rule

Machine_A

Openvswitch

Machine_B

Machine_C

Analysis

. . .
. . .
Rule 1
Rule 2
Rule 3

Flow Table

If Rule_2 == Rule_3 :
Controller_A.Priority = -1
Controller_B.Priority = 1
Controller_C.Priority = 2
...

Project Arch

Network Attack

Pwned
Controller_A
(Priority -1)

Controller_B
(Priority 1)

Controller_C
(Priority 2)

Add Rule

Machine_A

Openvswitch

Machine_B

Machine_C

Analysis

. . .
. . .
Rule 1
Rule 2
Rule 3

Flow Table

If Rule_2 == Rule_3 :
Controller_A.Priority = -1
Controller_B.Priority = 1
Controller_C.Priority = 2
...

Controller Attack

Project Arch

Controller Attack

Machine_A

Unknown Packet
With Attacking Payload

Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Openvswitch

. . .
. . .

Flow Table

Machine_B

Machine_C

Analysis

If recv(unknown packet) :
Ask Controller_A
...

Project Arch

Controller Attack

Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Ask Rule

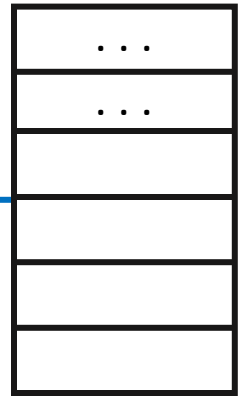
Machine_A

Openvswitch

Machine_B

Machine_C

Analysis



Flow Table

If recv(unknown packet) :
Ask each Controller(c)
...

Project Arch

Controller Attack

Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Add Rule

Machine_A

Openvswitch

Machine_B

Machine_C

Analysis

. . .
. . .
Analysis

Flow Table

If packtFromManager :
  Add Rule(Forward to Controller)
Else :
  Add Rule(Forward to Analysis)
Forward

Project Arch

Controller Attack

Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

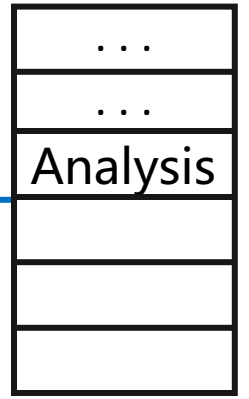
Machine_A

Openvswitch

Machine_B

Machine_C

Analysis



Flow Table

If packtFromManager :
  Add Rule(Forward to Controller)
Else :
  Add Rule(Forward to Analysis)
Forward

Project Arch

Controller Attack

Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Machine_A

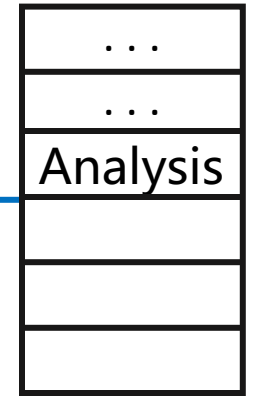
Openvswitch

Machine_B

Machine_C

Analysis

Analyze the packet



Flow Table

Project Arch

Controller Attack

Controller_A
(Priority 1)

Controller_B
(Priority 2)

Controller_C
(Priority 3)

Machine_A

Openvswitch

Machine_B

Machine_C

Analysis

. . .
. . .
Analysis

Flow Table

If contain malicious payload :
Drop packet from hacker
Else :
Repeat mechanism
...

Project Demo

Refrences

- https://en.wikipedia.org/wiki/Software-defined_networking
- https://hkitblog.com/淺談軟體定義：sdn-改變傳統複雜網路架構新趨勢/
- https://www.flaticon.com/
- IEEE-A Policy-Based Security Architecture for Software-Defined Networks


Q & A


010011
111001
010010
001100
010011
111001
010010
001100