WP4: Case Studies and Testing Environment

Tomohiro Ishihara

The University of Tokyo

Outline

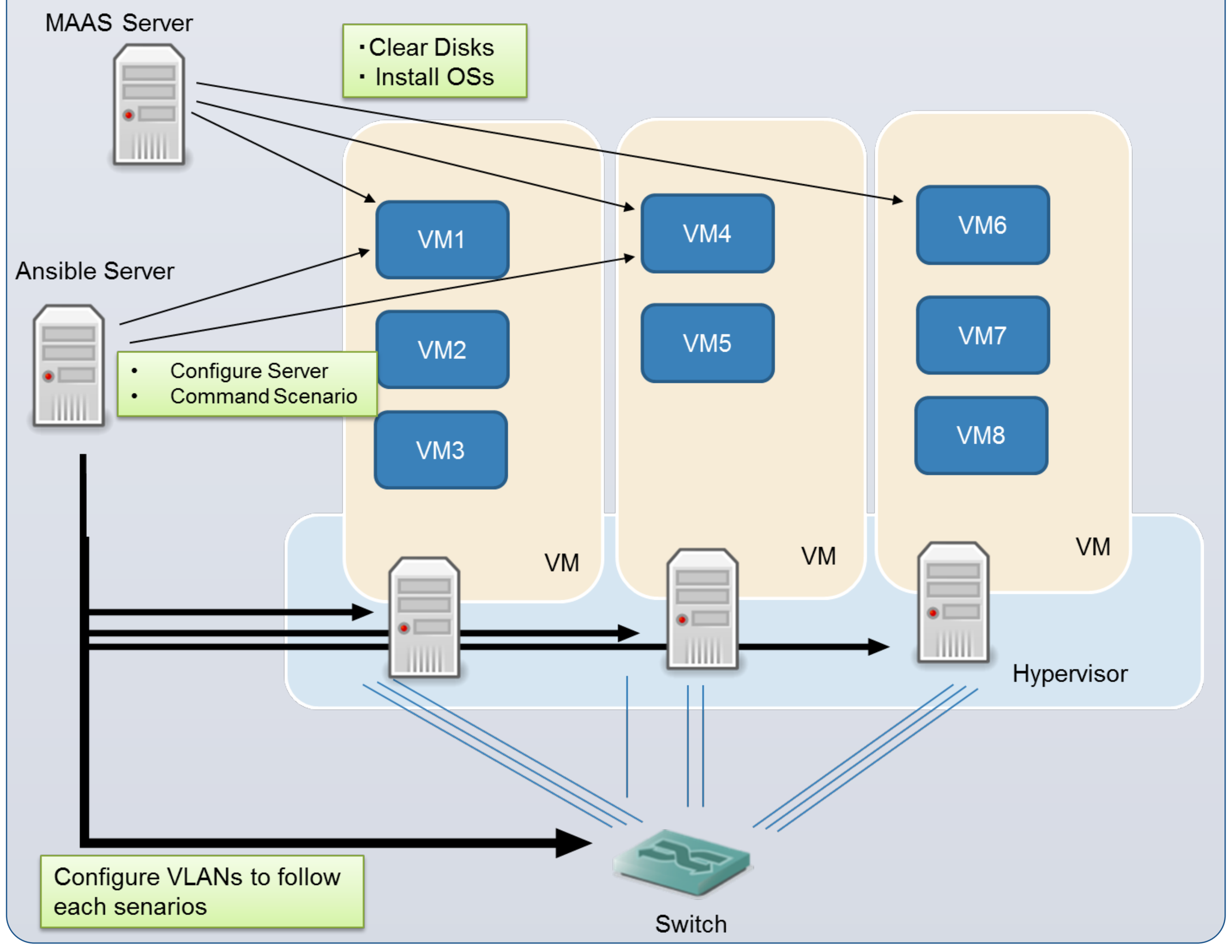
WP4 focuses on testing and demonstrating platform and methods which developed other WPs.
It contains following elements;

- Develop a testing environment which could evaluate this project's outcomes
- Demonstrate and Evaluate case studies

Testing Environment

We agreed to adapt a virtual machine environment for the demos due to the diversity of multiple scenarios on WP4. Therefore, we require flexibly in order to set up demonstration environments for each scenario. Accordingly, we chose the following tools for constructing the environments.

- virtual machine environment
  - **KVM**: a hypervisor (HV) for the virtual machine environment.
  - **virtsh**: provides controlling and management functions for KVM. The tool creates, configures and deletes guest VMs on HV machines.
- automatic deployment
  - **MAAS**: a software to install OS images on bare metal machines. It can automate the installation process of the OS (Ubuntu).
  - **ANSIBLE**: an automation tool to configure hosts based on predefined settings. The tool is used to set up software on hosts.


MAAS Server

・Clear Disks
・Install OSs

Ansible Server

- Configure Server
- Command Scenario

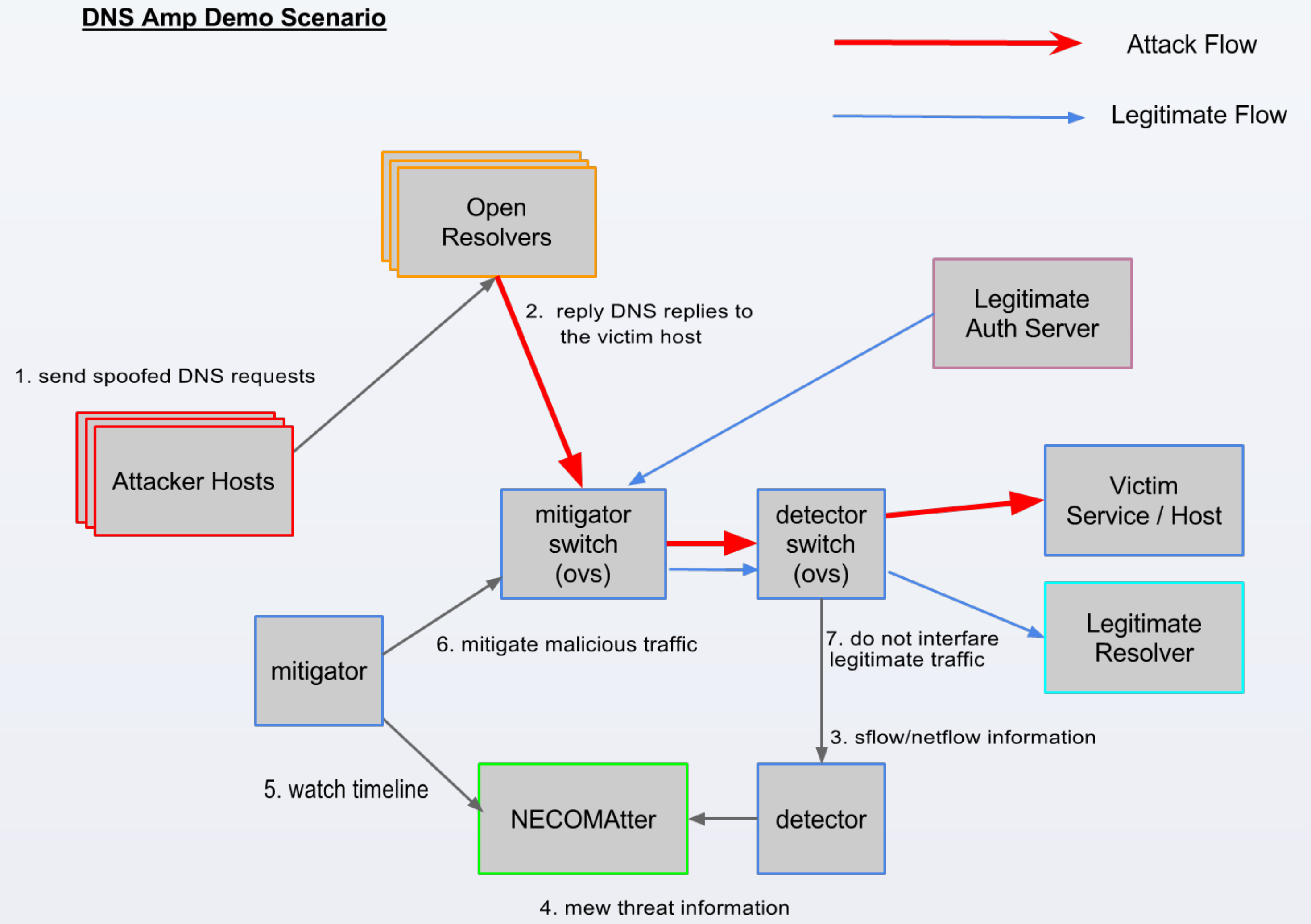
VM1
VM2
VM3
VM4
VM5
VM6
VM7
VM8
VM
VM
VM
Hypervisor

Configure VLANs to follow each senarios

Switch

Testing Scenarios

(1) DDoS Mitigation - DNS Amplification Attack


DNS Amp Demo Scenario

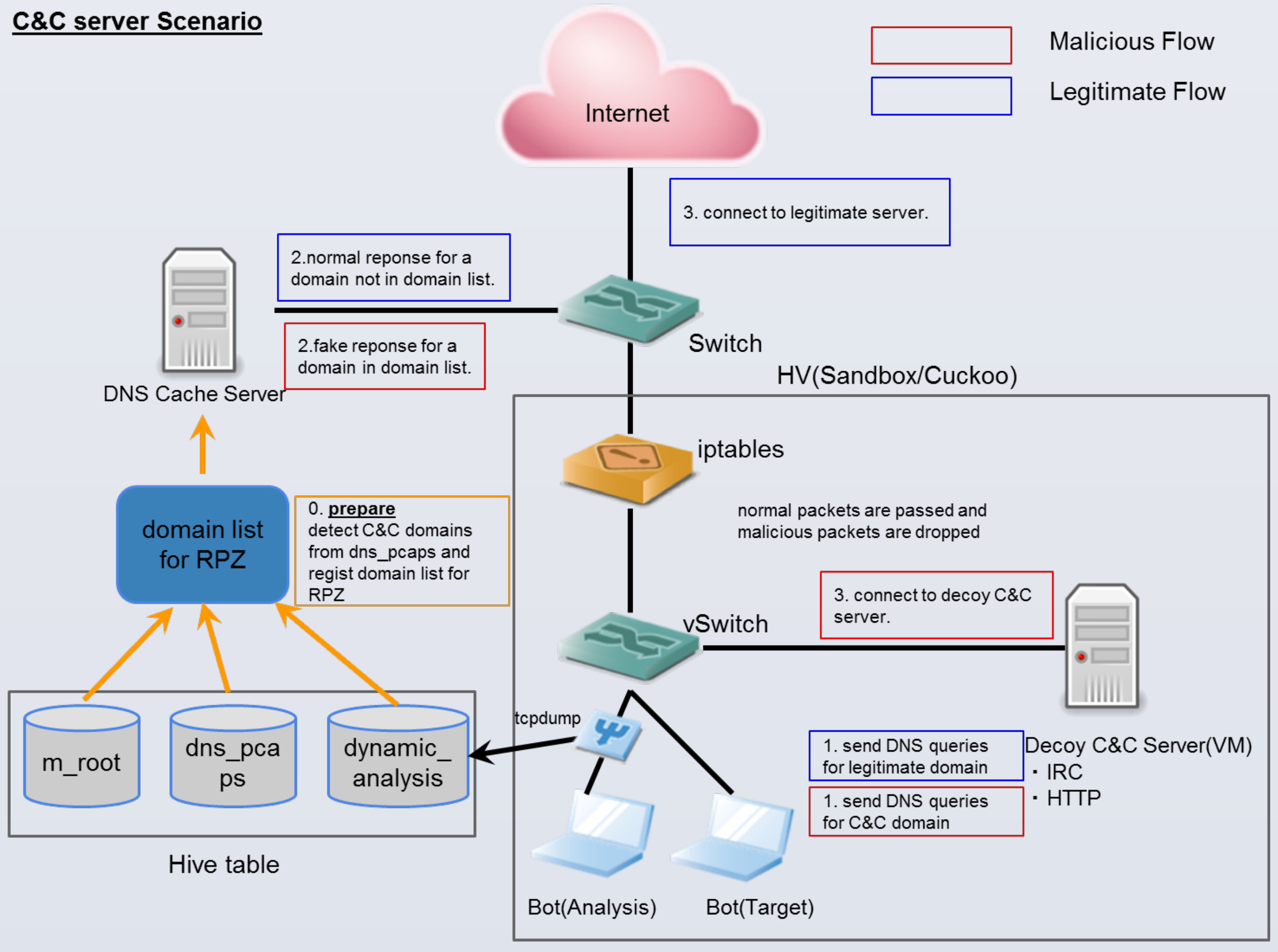
Attack Flow
Legitimate Flow

Open Resolvers
Legitimate Auth Server
Attacker Hosts
mitigator switch (ovs)
detector switch (ovs)
Victim Service / Host
Legitimate Resolver
mitigator
NECOMAtter
detector

1. send spoofed DNS requests
2. reply DNS replies to the victim host
3. sflow/netflow information
4. mew threat information
5. watch timeline
6. mitigate malicious traffic
7. do not interfare legitimate traffic

This scenario supposes a major DDoS attacking method which known as 'DNS Amplification Attack'. In this scenario, 'detector' detects series of attack and reports it through 'NECOMAtter'. 'mitigator' orders 'mitigator switch' to control these detected attacks.

(2) Botnet Introspection – detecting C&C servers


C&C server Scenario

Malicious Flow
Legitimate Flow

Internet
3. connect to legitimate server.
2.normal reponse for a domain not in domain list.
2.fake reponse for a domain in domain list.
DNS Cache Server
Switch
HV(Sandbox/Cuckoo)
iptables
normal packets are passed and malicious packets are dropped
domain list for RPZ
0. **prepare** detect C&C domains from dns_pcaps and regist domain list for RPZ
3. connect to decoy C&C server.
vSwitch
tcpdump
m_root
dns_pcaps
dynamic_analysis
Hive table
1. send DNS queries for legitimate domain
1. send DNS queries for C&C domain
Decoy C&C Server(VM)
・IRC
・HTTP
Bot(Analysis)
Bot(Target)

This scenario shows a botnet detecting and investigating mechanism which developed on this project. This system continuously analyzes traffic data to find out C&C servers and Botnet clients in their network. An analyzing method uses two kinds of network traffic – 1) DNS queries from Botnet client sandbox, 2) DNS queries from their commodity network. Using machine learning method which developed in this project, the system maintain a blacklist of C&C server. When botnet clients attempt to connect C&C servers, it will be induced to our 'Decoy' C&C server to investigate their traffic.

Contact
Tomohiro Ishihara
sho@c.u-Tokyo.ac.jp

東京大学 THE UNIVERSITY OF TOKYO

NECOMA