4th Year Professional Internship Report

BILLY Maxime
SI4-FISE
May 2024 – August 2024

Software-Defined Networking Security

Digital Science and Technology Institute

Company tutor :
Hoang Huu Duc
Lecturer at the Vietnam-Korea University of Đà Nẵng

Academic tutor :
Lippi Sylvain
Lecturer at the University of Nice Sophia Antipolis

July 2024

POLYTECH

UNIVERSITÉ CÔTE D'AZUR

DNIIT

Acknowledgements

I would like to thank my internship supervisor at VKU[1], Hoang Huu Duc, for his helpful guidance and assistance throughout my internship.

I also want to thank my academic supervisor, Sylvain Lippi, for his support throughout my internship. He advised me on writing this report and preparing for my presentation.

Likewise, I would like to thank Mr. Lê Thành Nhân for his support throughout my stay in Vietnam. He was of invaluable assistance with administrative procedures, finding accommodation, and discovering activities in Da Nang. His role as a link between Polytech and Vietnam greatly facilitated my integration.

I am also grateful to the people at Polytech who supported me during this internship. I thank Florian Bridoux, responsible for internships at SI4, for his guidance and valuable advice, and Julie Maiffret, the internship agreement manager, for her effective assistance. Their support was crucial to the success of this professional experience.

Additionally, our thanks go to the VSL[2] for the training provided through their CTF platform (vsl.ce.vku.udn.vn). The interactive challenges have greatly enhanced my cybersecurity skills.

Finally, Melanie Stanislas for her help writing this report and her support throughout my internship.

[1] Vietnam-Korea University of Information and Communications Technology
[2] VKU Security Lab

Abstract

This report presents the research and technical activities conducted during my internship at the Digital Science and Technology Institute (eSTI) within the Vietnam-Korea University of Đà Nẵng.

Collaborating with Mélanie Stanislas, under the supervision of Professor Hoang Huu Duc, our focus was on enhancing the security of software-defined networks (SDN).

Our work included in-depth research into SDN security, particularly using OpenDaylight[3] as a controller, and the development of a secure, virtualized lab environment using Mininet[4] for testing purposes.

Additionally, we wrote an article on Distributed Denial of Service (DDoS) attacks to be presented at the AWRIS[5] 2024 conference, and improved our cybersecurity skills through various certifications and practical challenges.

This report details our methodologies, findings, and the progress made in understanding and securing SDN environments.

Résumé

Ce rapport présente les activités de recherche et techniques menées pendant mon stage à l'Institut des Sciences et Technologies Numériques (eSTI) au sein de l'Université Vietnam-Corée de Đà Nẵng.

En collaboration avec Mélanie Stanislas et sous la supervision du professeur Hoang Huu Duc, nous nous sommes concentrés sur l'amélioration de la sécurité des réseaux définis par logiciel (SDN).

Notre travail a inclus une recherche approfondie sur la sécurité des SDN, en particulier en utilisant OpenDaylight comme contrôleur, et le développement d'un environnement de laboratoire sécurisé et virtualisé utilisant Mininet pour des tests.

De plus, nous avons rédigé un article sur les attaques par déni de service distribué (DDoS) à présenter lors de la conférence AWRIS 2024 et avons amélioré nos compétences en cybersécurité grâce à diverses certifications et défis pratiques.

Ce rapport détaille nos méthodologies, nos découvertes et les progrès réalisés dans la compréhension et la sécurisation des environnements SDN.

[3] https://www.opendaylight.org/
[4] http://mininet.org/
[5] ACIR+ Workshop of Research, Innovation, and Entrepreneurship among Students

Table of Contents

Introduction

1. Context of the internship

During my internship, I worked with Mélanie Stanislas, a student at Polytech Nice Sophia at University Côte d'Azur. Our joint work focused on the **technical and research aspects of software-defined network (SDN) security**. We were supervised by **Professor Hoang Huu Duc**, from the Vietnam-Korea University of Da Nang, who guided us in our research and work throughout the internship.

Communication within the team was in English, as our supervisor spoke Vietnamese and English, while Mélanie and I communicated in French and English. The teacher was available to give us advice on an ad hoc basis, while giving us considerable autonomy in our work.

2. Host organization

The **Đà Nẵng International Institute of Technology** was created in 2017 as the result of a partnership between the University of Đà Nẵng and Côte d'Azur University. The goal of this Institute is developing collaborations between universities in the fields of research, training, and innovation.

The DNIIT **focuses its research on societal themes** identified by the Côte d'Azur University Initiative of Excellence. These themes include artificial intelligence, internet of things, smart territories, sustainable development, health, well-being and aging.

2.1. Structure

2.1.1. Location

My internship took place in the **Digital Science and Technology Institute** (eSTI) located inside the **Vietnam-Korea University** of Đà Nẵng (VKU).

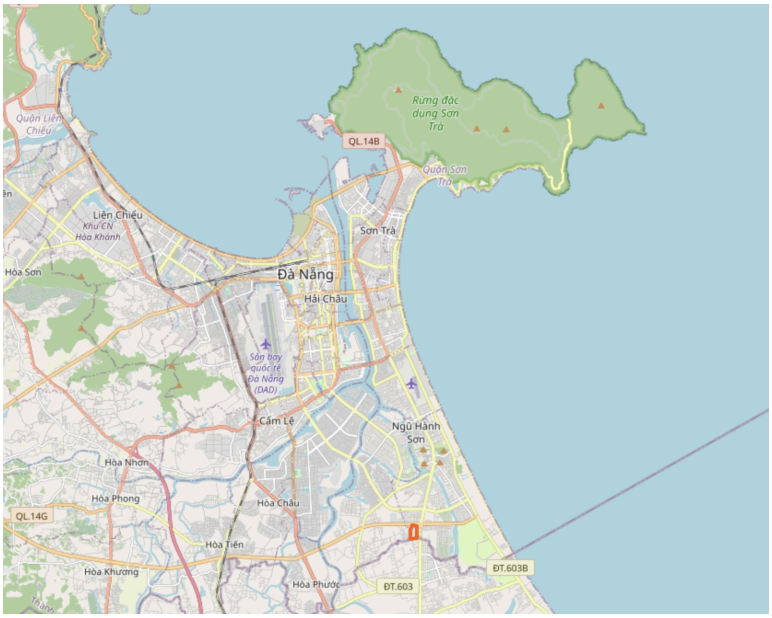

Figure 1 : OpenStreetMap Screenshot of the location of VKU in Đà Nẵng

The University is located south of Đà Nẵng in the Ngũ Hành Sơn District (cf *Fig.1*), on 470 Tran Dai Nghia.

Figure 2: 3D plan of the eSTI building

The **eSTI Institute** is composed of 3 main buildings, the main hall (at the center of *Fig.2*), the library and a study center (on the right of *Fig.2*) and **the research center where I worked** (on the left of *Fig.2*).

The Digital Science & Technology Institute (eSTI) was established as part of the Vietnam-Korea University of Information and Communication Technology (VKU) with support from the Korea International Cooperation Agency (KOICA) on December 19, 2022. Its creation aims to enhance scientific research, technology transfer, and innovation at VKU, contributing to the socio-economic development at local, national, and international levels.

2.1.2. Hierarchy

Figure 3 : List of Lecturers

The eSTI is composed of 12 Lecturers (cf Fig.3) led by Huynh Cong Phap, Senior Lecturer, followed by its deputy Nguyen Quang Vu, Main Lecturer. The Institute usually welcomes students in internships.

2.2. Duties

The duties of the Digital Science & Technology Institute (eSTI) encompass several core responsibilities. Primarily, eSTI promotes comprehensive scientific research, technology transfer, and innovation within the Vietnam-Korea University of Information and Communication Technology (VKU). It serves as a collaborative hub for researchers, policymakers, and industry leaders **to enhance the quality and practical application of scientific research**. The institute focuses on the commercialization of VKU's scientific and technological products, aiming to achieve sustainable development and elevate the university to an international standard.

eSTI is organized into various research teams focusing on key areas. These areas include Artificial Intelligence in Natural Language Processing, New Generation Communication Technology, ICT in Medicine, and Advanced Materials. Additionally, the institute has teams working on Artificial Intelligence and IoTs, Intelligent Systems, and Artificial Intelligence in the Digital Economy. The institute is actively engaged in significant research projects, including optimizing object coverage for IoT systems and exploring the intersection of technology and medicine. These projects not only advance scientific knowledge but also contribute to the **digital transformation and socio-economic development** of the Central – Central Highlands region and beyond.

Internship Duties

1. Research on the security of Software-Defined Networks

During my internship, I was tasked with carrying out in-depth research into the security of **software-defined networks (SDN)**.

1.1. What is SDN Security ?

SDN is an approach to network management that **uses software rather than network equipment**. This makes it possible to manage networks more flexibly and efficiently, bringing the **flexibility** of software development to the world of networking. But on the other hand, the SDN approach brings the security challenges of software development to the world of networking.

One part of my research focuses on **Opendaylight**, a **SDN controller**. SDN controllers are responsible for controlling the network flows and managing the network devices. This critical role makes them a **prime target for attackers**. Compromising a controller could allow an attacker to shut down a network, hide itself from the rest of the network from potential detection.

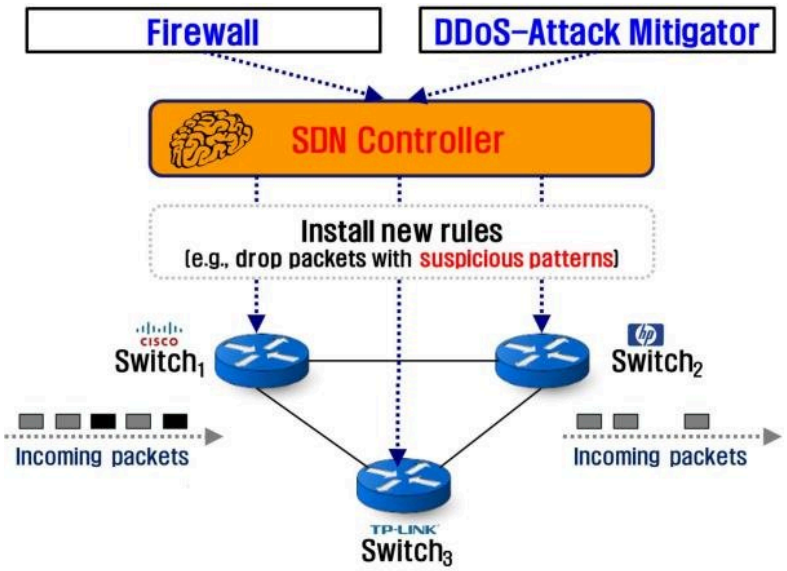

Figure 4 : SDN Security diagram

1.2. Discover the tools

The first part of the internship was dedicated to understanding how various tools and technologies in the field of Software-Defined Networking (SDN) worked. In addition to exploring **OpenDaylight**, an open-source SDN controller, the **OpenFlow** protocol for communication between the controller and network devices, we also studied **Mininet**, a network emulator that allows for the creation of virtual network environments. As we progressed in learning how to utilize these tools, we wrote detailed articles documenting their usage, which served as reference material for the later stages of the internship.

1.2.1. OpenDaylight

We began to document the development cycle of the **OpenDaylight[6] Project** and the community behind it. All collaboration and development activities for OpenDaylight are managed through Git and Gerrit on the Linux Foundation infrastructure (git.opendaylight.org). Modules and features are developed in separate repositories and reviewed by the community, which provides us with insight into the collaborative nature of **open-source** projects.

OpenDaylight is built using **Java** and utilizes the Karaf framework, a lightweight OSGi container that supports the deployment and management of multiple modules within a single Java Virtual Machine (JVM). This modular architecture allows OpenDaylight to be **easily extended with new features and functionalities.**

For instance, the **LISP (Locator/ID Separation Protocol)** plugin allows for efficient IP routing and addressing, providing scalability and flexibility in network management. Another example is the **MD-SAL (Model-Driven Service Abstraction Layer)** module, which provides a framework for data modeling and interaction between applications and the network. Additionally, the **AAA (Authentication, Authorization, and Accounting)** plugin ensures secure access control and auditing within the SDN environment, crucial for maintaining network security and compliance.

Companies like **Cisco[7]** and **Orange[8]** use OpenDaylight to manage and automate their large-scale network infrastructures, enhancing service delivery and network efficiency.

Figure 5 : OpenDaylight logo

[6] https://www.opendaylight.org/
[7] https://www.cisco.com/
[8] https://www.orange.com/en

1.2.2. Mininet

Mininet[9] is a **network emulator** that allows users to create a **virtual network on a single machine**. It provides a realistic environment for testing and developing network configurations by creating virtual hosts, switches, controllers, and links. This capability makes Mininet an essential tool for experimenting with Software-Defined Networking (SDN) technologies, enabling researchers like us and developers to simulate complex network scenarios and evaluate different configurations effectively.

Here are some key functions in Mininet :

- **addHost()** : Adds a host to the Mininet topology, representing a device connected to the network.
- **addSwitch( )** : Adds a switch to the Mininet topology to manage network traffic.
- **addLink()** : Establishes a connection between two nodes in the topology, simulating a network link.
- **pingAll()** : Tests connectivity between all hosts in the topology by sending ICMP pings.

Using Mininet, we found it to be user-friendly and easily understandable, making it accessible to users at all levels of experience.

1.2.3. MiniAttack

While researching Mininet, we discovered **MiniAttack**, a tool designed **to work with Mininet** for testing network security. MiniAttack offers a range of **pre-configured network attacks** and monitoring tools, allowing users to simulate different threat scenarios and assess the robustness of their network security measures. MiniAttack can simulate **DoS (Denial of Service) attacks** using multiple protocols and weaknesses, allowing users to assess network security robustness effectively.

This tool is particularly useful for identifying vulnerabilities and improving the overall security posture of SDN environments.

We began by simulating a DoS attack using MiniAttack, following the instructions in the project's GitHub repository README to understand the tool.

1.3. Set up the sandbox

Our primary goal during this phase was to **build a lab environment** specifically designed to test the security of OpenDaylight. We aimed to create a repeatable and isolated environment, separate from the rest of the network, where we could safely introduce potential security threats and evaluate the robustness of OpenDaylight's security measures.

[9] http://mininet.org/

1.3.1. Virtualization

Initially, we used a single physical Kali Linux machine and its suite of security tools to test and evaluate our SDN security measures, provided by the Institute. However, it quickly became apparent that to achieve a more reliable and flexible environment, **we needed to virtualize our setup**. We transitioned to a virtualized environment using **Proxmox**[10], which allowed us to create **multiple Debian virtual machines.** Proxmox is a **virtualization platform** that allows users to create and manage virtual machines and containers efficiently. Its advantages include flexible resource management, robust security features, and support, making it ideal for diverse virtualization needs.

Figure 6 : Proxmox interface

This virtualization approach enabled us to isolate different components of our network from each other and from external networks, creating a controlled environment for our security tests.

1.3.2. Automation

To ensure a smooth setup and **deployment of our SDN testing environment**, we utilized several **Ansible**[11] **playbooks** to automate the installation and configuration processes. These scripts allowed us to efficiently **deploy OpenDaylight, Mininet, and MiniAttack** across our virtual machines.

For **OpenDaylight**, we designed a playbook to handle its installation using **Docker**[12]. The playbook includes tasks such as:

- **Installing Docker** via a dedicated task file.
- Creating the necessary **directory structure** for OpenDaylight.
- Writing the **Docker Compose configuration file** from a template.
- **Starting** OpenDaylight using Docker Compose.
- Waiting for OpenDaylight to be **fully operational**.
- Displaying the latest OpenDaylight **logs** for verification purposes.

[10] https://www.proxmox.com/en/
[11] https://www.ansible.com/
[12] https://www.docker.com/

For **Mininet**, we created another playbook to handle its installation, along with optional Ryu[13] support.

Finally, for **MiniAttack**, which is essential for our security testing, we crafted a playbook to **install all necessary dependencies** and **clone the MiniAttack repository**.

These playbooks collectively ensured that our lab environment was set up efficiently, allowing us to focus on setting up attacks and defenses to identify vulnerabilities in the SDN technologies.

1.3.3. On demand Sandbox

The final step of this subtask is to create a **platform** where any user could **connect**, **request a VM**, **configure** what tools to install and **get access to the VM, controller, and a dashboard**. This would greatly help in the research and development of SDN security, as it would allow researchers to quickly test their tools and ideas in a controlled environment. This would also help in the development of new tools and techniques to secure SDN networks, as researchers would be able to test their tools in a real-world environment without having to set up their own lab.

We started **brainstorming** about the features to include in the platform. We decided that the platform should have a dashboard that would allow users to visualize the network architecture, data flows, inspect packets, and connect to any machine.

This project is still in the planning phase, but we identified the following services that we would need to implement:

- A **service responsible for creating and managing VMs** with a **Terraform**[14] integration. This service would allow users to request a VM, configure the VM, and get access to the VM. It is the basis upon which the rest of the sandbox would be run.

- A **service responsible for configuring the VMs**. This service would be responsible for installing the necessary tools and software on the VMs, as requested by the user, and giving the user access to the tools.

- A **service responsible for monitoring the network**. This service would be responsible for monitoring data flow and capturing packets on the simulated network. It would also be responsible for visualizing the network architecture.

Given these services, we would be able to create a **frontend to display this information and control the network**.

[13] https://ryu-sdn.org/
[14] https://www.terraform.io/

1.3.4. Final utilities

The **purpose of the sandbox environment** was to enable the **setup of attacks and defenses on the network**, thereby identifying vulnerabilities in the technologies we were working with. By creating a controlled and isolated lab, we could safely introduce various security threats and test the resilience of our SDN components, such as Mininet, OpenDaylight, and OpenFlow. Additionally, we aimed to leverage **AI to enhance our security measures**, using machine learning algorithms to detect and respond to security threats in real-time. This proactive approach allowed us to not only uncover potential weaknesses, but also to develop and refine strategies for mitigating these vulnerabilities effectively.

To conclude, this comprehensive approach allowed us to deepen our **understanding of SDN security concepts**, explore the interactions between these technologies, and develop **practical skills** in managing and securing SDN environments.

1.4. Security analysis of OpenDaylight

Our **security analysis** began with the **OpenDaylight codebase**. We enumerated the already known **Common Vulnerabilities and Exposures (CVE)** and their corresponding **Common Weakness Enumeration (CWE)** to gain insight into potential security weaknesses.
CVE identifies specific vulnerabilities, while CWE categorizes common types of software weaknesses that can lead to vulnerabilities.

To visualize the **evolution of these vulnerabilities over time**, we constructed a **timeline graph** of the CVEs. This analysis identified **four main categories of vulnerabilities**: SQL Injection (CWE-89), Improper Input Validation (CWE-20), Unchecked Return Value (CWE-476), and Incorrect Calculation of Buffer Size (CWE-400). This comprehensive overview guided us to focus on addressing these specific vulnerabilities.

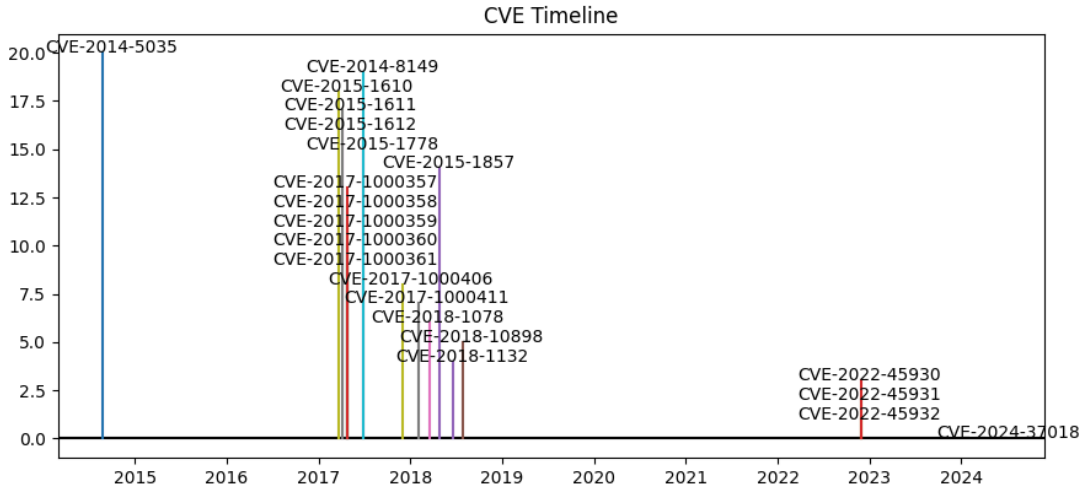

Figure 7 : OpenDaylight CVE Timeline

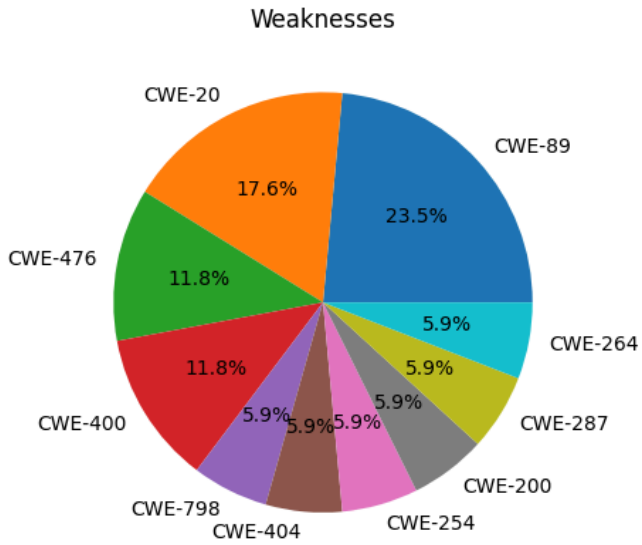

Figure 8 : OpenDaylight CWE Weaknesses

We then used a **SonarQube[15] scan** to **detect potential security weaknesses in the code**. This yielded one potential **DOS vulnerability** in the 'md-sal' module responsible for managing the data store. There was also potentially **weak cryptography** in the same module in the raft implementation. We are still investigating these potential issues.

In the future, we aim to **expand the analysis of the codebase to more ODL modules**. Due to the nature of the ODL project, plenty of modules are available to enable. These modules are not necessarily developed by the same team and are probably not under the same scrutiny as the core of ODL. This could lead to potential security issues that we would like to investigate.

1.5. Report

The final step of this project will be to **compile our findings into a comprehensive report**. This report will detail our research methodology, the tools, and technologies we used, the setup of our lab environment, the security analysis of OpenDaylight, and the results of our security testing. We will also include recommendations for improving the security of OpenDaylight and other SDN technologies based on our findings. This report will serve as a valuable **resource for organizations looking to enhance the security of their SDN environments and for researchers** interested in further exploring the security implications of SDN technologies.

[15] https://www.sonarsource.com/products/sonarqube/

2. Article on the Anatomy of a DDoS attack

2.1. Purpose

In addition to the research on SDN security, we were tasked with **writing an article** to **participate in the AWRIS[16] 2024 conference representing the Côte d'Azur University**. After some research in the field of cybersecurity, we presented potential topics to Mr. Hoang Huu. We decided to write an article on the **Anatomy of a DDoS Attack**.

We decided to write an article on this topic because it is a very common attack, and it is closely related to our research on SDN security. As such, we decided to write an article that would be accessible to a wide audience, including non-technical people, on the inner workings of a DDoS attack. As a title, we chose "**Anatomy of a DDoS Attack: From Host Infection to Service Denial**".

The goal of the article is to **explain how a DDoS attack works**. We identified three major steps in a DDoS attack: the **infection of the hosts**, the **command and control of the botnet** and the **attack** itself. We also wanted to make sure that this paper wouldn't be able to be used to perform a read DDoS attack, as such we concentrated on explaining principles, not technical details.

2.2. Building the virus

The first part of the paper explains the concept of a **Command and Control Server (C2)** and how it works with a **Remote Access Trojan (RAT)**.
A **C2 server** is a centralized **server used by attackers to send commands** and receive information from compromised devices in a botnet or malware network.
A **RAT** is malicious software that allows **remote access and control** of a computer or device without the user's knowledge or consent. Well-known types of RATs in the realm of cybersecurity include **backdoors**, **keyloggers**, and **remote administration tools**, each designed to enable unauthorized access and control over compromised systems.

To demonstrate this principle without using real malware, we built a simple **python script** that would **connect to a C2 server and execute commands** (cf Fig.9). In addition to this, we also built a simple C2 server that would serve commands to the botnet. This allowed us to demonstrate the **principle of a C2 server without using real malware**.

[16] ACIR+ Workshop of Research, Innovation, and Entrepreneurship among Students

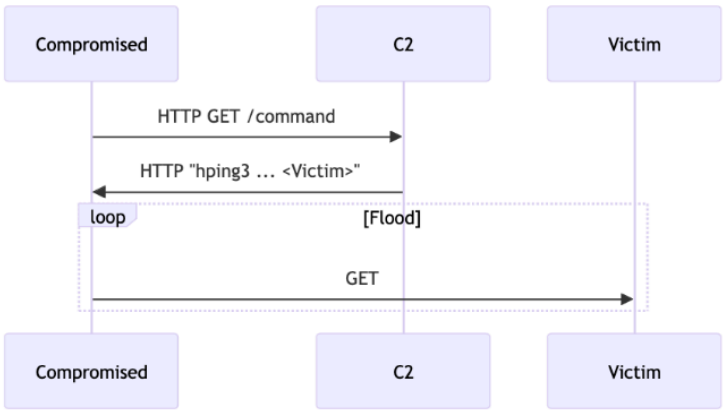

Figure 9 : Sequence diagram

2.3. Host compromise

The second part of the article explains different ways a host can be infected with our fake RAT, such as **phishing**, **weak passwords/misconfigurations** or **vulnerable software**.

So for **phishing**, we demonstrated the use of uncensored **LLMs**[17] to **automatically write emails to potential victims**. This produced a realistic phishing email that could be used to infect a host. We then presented potential ways to protect against phishing attacks, such as verifying the sender's email address and not clicking on links in emails.

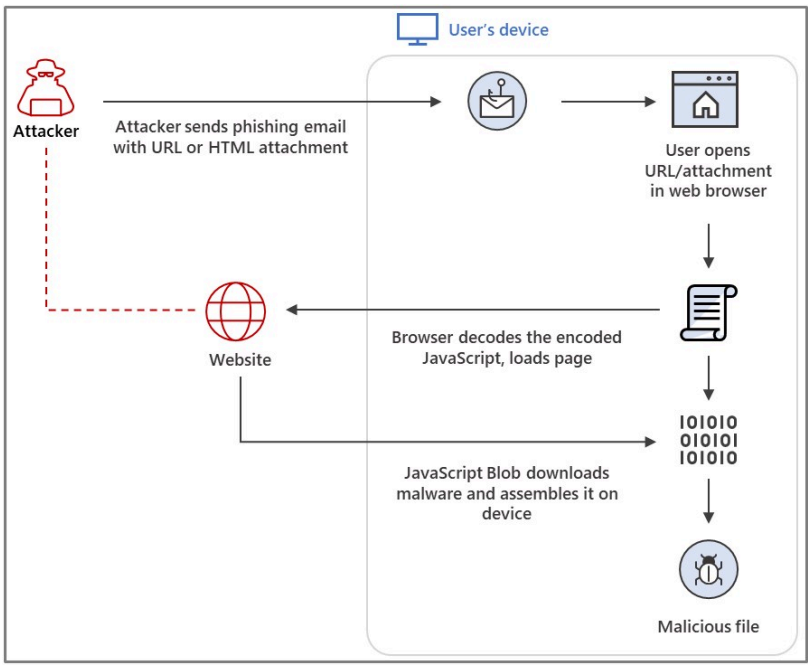

Figure 10 : Schema of phishing attack

[17] Large Language Models

For **weak passwords/misconfigurations**, we built a **SSH brute force attack** script that would try to connect to a server using a list of common passwords. This allowed us to demonstrate how an attacker could gain access to a server using weak passwords and a misconfigured SSH server. We then presented potential ways to protect against this type of attack such as using strong passwords, disabling password authentication and using a tool like Fail2Ban[18] to block brute force attacks.

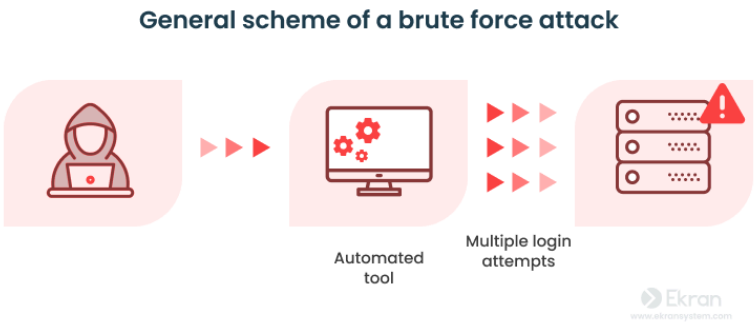

Figure 11 : Schema of a brute force attack

For **software vulnerability**, to write this section of the article, we choose to deal with the security vulnerability in poorly secured web applications that allow **file uploads**. We described how an attacker could exploit this vulnerability by **uploading a PHP file designed to execute arbitrary commands on the server**. We provided a practical example, demonstrating how to upload a PHP file that uses the 'system' function to run commands from URL parameters, thereby compromising the server. Finally, we discussed **mitigation strategies**, emphasizing the importance of keeping applications updated, applying security patches, conducting regular audits, and implementing measures such as file type checks, file size limits, and filename sanitization to prevent such attacks.

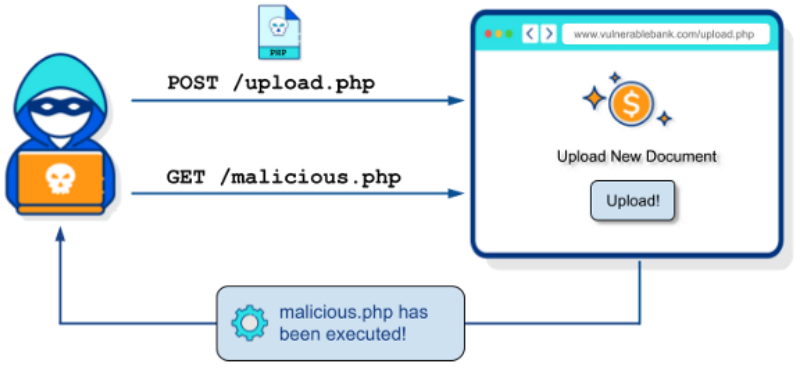

Figure 12 : Malicious file upload vulnerability scheme

[18] https://github.com/fail2ban/fail2ban

2.4. The attack

We then presented the attack itself, using the **hping3[19] tool** to perform a **SYN flood attack** with large packets on a server with the goal of saturating the server's network connection. We described how a Command and Control (C2) server can instruct these hosts to generate a flood of traffic to overwhelm the target server.

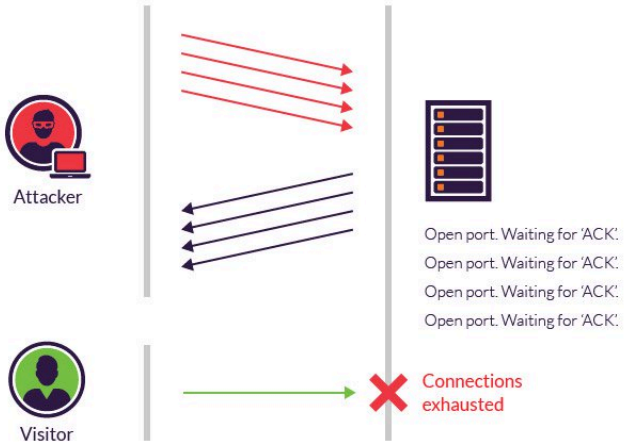

Figure 12 : SYN flood attack scheme

To better demonstrate this attack, we built a **simple network using Mininet[20]** *(cf Fig. 13)* and automated the steps to **perform the attack with a Python script**. This allows students and readers to reproduce the attack in a safe environment easily.

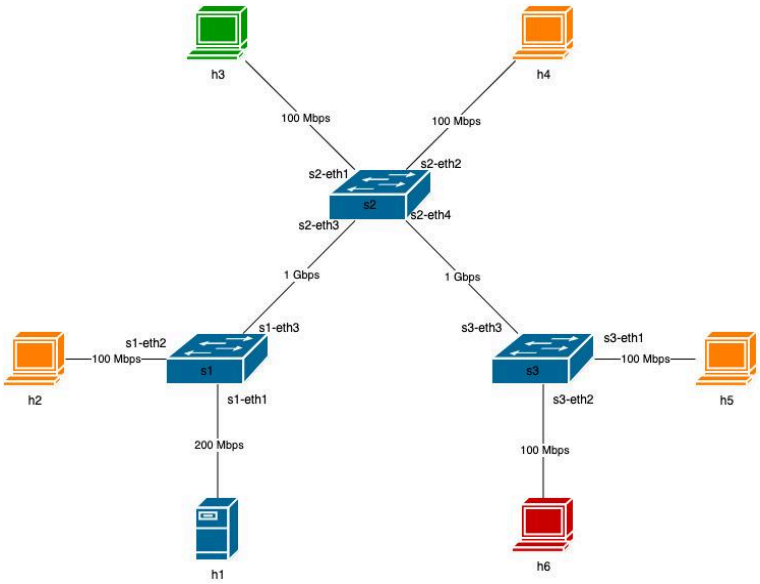

Figure 13 : Network Topolog

[19] https://www.kali.org/tools/hping3/
[20] http://mininet.org/

2.5. Forensics and analysis

An important step of the article was to explain how to **detect and analyze a DDoS attack**. As such, we introduce two famous tools, **tcpdump**[21] and **bwm-ng**[22].

- **tcpdump** is a tool that allows us to **capture packets** exchanged on a network interface. We used it to capture the packets exchanged during the attack and **analyze them** to understand how the attack was performed.

- **bwm-ng** is a tool that allows us to **monitor the network traffic on a server**. We used it to monitor the network traffic during the attack and **show the effects of the attack** on the server's network connection.

We built these tools into our Demo Lab, starting the packet capture and bandwidth monitoring on each network interface before launching the attack, stopping them after the attack and exporting the data at the end.

Furthermore, we **show ways to analyze the data** captured by these tools, such as looking at the number of packets exchanged, the size of the packets, the source, and destination IP addresses, and the protocols used. To achieve this, we used **Wireshark**[23] to analyze the packets captured by tcpdump.

To better visualize the data from a whole network perspective, we wrote some **python scripts to graph the network traffic over time between the hosts**[24]. We included these graphs in the article to show the effects of the attack on the network bandwidth.

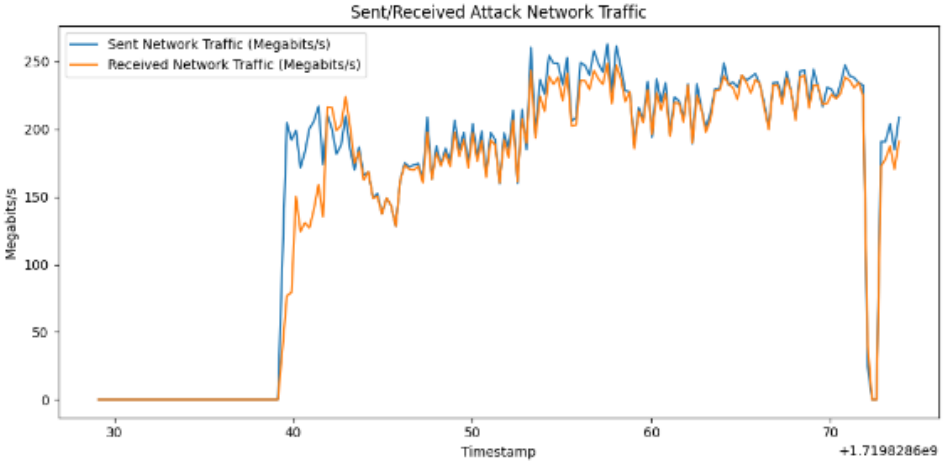

Figure 14 : Example of a network traffic graph on the target server

[21] https://www.tcpdump.org/
[22] https://linux.die.net/man/1/bwm-ng
[23] https://www.wireshark.org/
[24] https://github.com/ozeliurs/SDN-Security/tree/main/papers/.project-files/ddos-attack/forensics

2.6. Prevent DDoS attack

In our efforts to **mitigate potential attacks**, we explored several defensive strategies, particularly focusing on using an **SDN controller**.

One primary solution was to **grant more bandwidth** to the server under attack, helping to manage increased traffic loads. However, leveraging an **SDN controller offers a more dynamic and effective defense**.

An **SDN controller** can detect and **block malicious activities in real-time** by **monitoring traffic patterns** and identifying anomalies indicative of an attack. For instance, it can automatically reroute or drop malicious packets when detecting abnormal traffic spikes, neutralizing the threat before it impacts the network. Additionally, SDN controllers can implement **advanced security mechanisms** such as rate limiting, ensuring only legitimate traffic reaches critical resources. By integrating **machine learning algorithms**, the SDN controller can predict and respond to threats more effectively, continuously **adapting to new attack patterns**.

Overall, using an **SDN controller** provides a **flexible and powerful solution** for protecting against network attacks, enhancing the security and resilience of our network infrastructure.

2.7. Corrections and reviews

The first versions of the article were reviewed by Mr. Hoang Huu, and we are currently working on the final version of the article.
We presented the article to a board of professors at the Institute, and they chose us to **represent the Côte d'Azur University at the AWRIS[25] 2024 conference**.

2.8. Presentation

You can read the article before it's published by downloading the latest version of the PDF from the GitHub releases under the assets section, the file named "Anatomy.of.a.DDoS.Attack.From.Host.Infection.to.Service.Denial.pdf", https://github.com/ozeliurs/SDN-Security/releases/latest.

In the near future, we plan to finalize the article and start working on the presentation. This presentation will include a **live demonstration of the attack and the defenses against it**.

[25] ACIR+ Workshop of Research, Innovation, and Entrepreneurship among Students

We also planned a final review of the article and the presentation before the conference, and we will give access to our repository to the attendees so they can reproduce the attack and the defenses against it.

Mr. Hoang Huu encouraged us to **apply for a conference** such as the Journal of Theoretical and Applied Information Technology[26] (JATIT) **to publish our article**.

3. Improving our security skills

We felt the need to **level up our skills in cybersecurity**. Following Mr. Hoang Huu's advice, we took the "**Introduction to Cybersecurity**"[27] **Cisco course and got certified**. The "Introduction to Cybersecurity" certification provides **foundational knowledge on cybersecurity concepts**, common threats, and protection strategies, making it an essential starting point for anyone interested in entering the cybersecurity field or safeguarding their personal online presence.

We followed this certification with the "**Fortinet Certified Fundamentals Cybersecurity**"[28] and "**Fortinet Certified Associate Cybersecurity**"[29] certifications. The "Fortinet Certified Fundamentals in Cybersecurity" certification validates entry-level technical knowledge and skills required to understand today's threat landscape and fundamental cybersecurity concepts. And the "Fortinet Certified Associate in Cybersecurity" certification validates ability to perform high-level configuration and monitoring of devices, covering fundamental operating procedures for their most common features.

These certifications helped us to understand the basics of cybersecurity and to be able to better understand the security issues of SDN.

Figure 15 : Validated certifications

[26] https://www.jatit.org/
[27] https://www.netacad.com/courses/cybersecurity/introduction-cybersecurity
[28] https://training.fortinet.com/local/staticpage/view.php?page=fcf_cybersecurity
[29] https://training.fortinet.com/local/staticpage/view.php?page=fca_cybersecurity

We also got to know the **VKU Security Lab**[30], a lab that is dedicated to teaching students about cybersecurity. We were able to use their resources in the form of CTF challenges to improve our skills in cybersecurity. For example, I personally solved several **forensic challenges** such as "Logs Network Forensic 7," which involved analyzing network traffic logs to identify malicious activities, and "Ram Level 3," where I performed memory forensics to uncover hidden data.
Solving these CTF challenges significantly **improved my security skills** by providing hands-on experience in **analyzing network traffic**, performing **memory forensics**, and identifying and mitigating various **cyber threats**.

Figure 16 : VSL logo

Furthermore, we also got invited to participate in the **Digital Dragons CTF Challenge**[31] by the VKU. The **Digital Dragons CTF Challenge** is a prestigious **cybersecurity competition** that tests participants' skills in various areas such as cryptography, network forensics, and web security, fostering a collaborative and competitive environment that is highly valued in the cybersecurity community. We are in the process of forming a **team of 5 students of Côte d'Azur university** to compete in the qualification round on the 10th of August.

By participating in these activities and achieving these certifications, we have significantly **enhanced our cybersecurity skills**, making us better equipped to understand and address the security challenges.

[30] https://vsl.ce.vku.udn.vn/
[31] https://digitaldragonsctf.com/

Planning

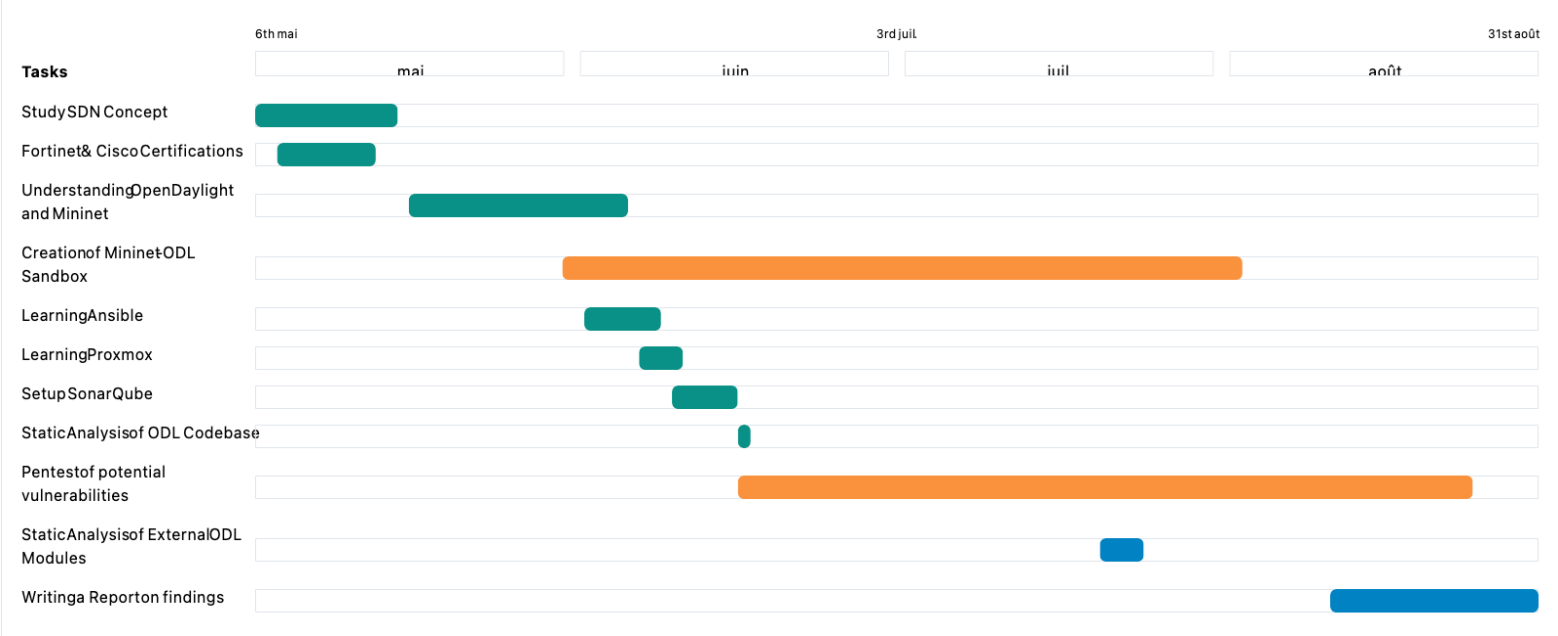

Figure 17 : Planning of Duty 1 - Research on the security of Software-Defined Networks

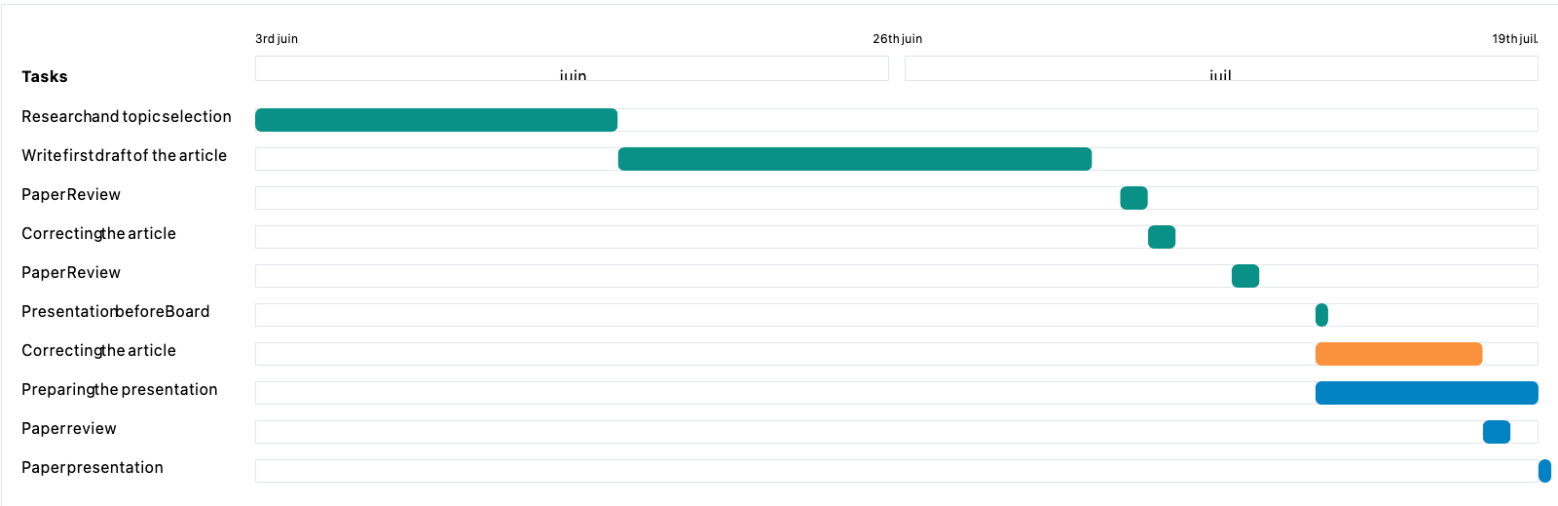

Figure 18 : Planning of Duty 2 - Article on the Anatomy of a DDoS attack

As of 12/07/2024, we have nearly finished the task of writing the article and I plan to finish it before the end of July.

As for the research task, it is difficult to predict when it will be finished, as it is an ongoing project. However, we are making good progress and hope to have some results by the end of the internship to present to the Institute.

Bibliography

Introduction

- Official DNIIT Website : “Presentation Institute”
- Official VKU Website : “Digital Science and Technology Institute”
- Official VKU Website : “Goc phai”
- Official VKU Website : “VKU: Opening the Research, Innovation Space and Digital Science & Technology Institute (eSTI)”
- “Software Defined Networking with OpenFlow”, Siamak Azodolmolky.

Internship Duties

- ResearchGate : "SDN-based security services"
- GitHub: "Anatomy of a DDoS Attack: From Host Infection to Service Denial"
- Ivan Stechynskyi : "Brute Force Attacks: How to Detect and Prevent Them"
- Imperva Website : "What is a TCP SYN Flood"
- Mininet’s GitHub : "OpenFlow Tutorial"
- Brandon Heller : "OpenFlow/SDN Introduction with Brandon Heller"
- Scott Shenker: "The Future of Networking and the Past of Protocols"
- Martìn Casado : "List of OpenFlow Software Projects"
- OpenDaylight Official Website : "Welcome to OpenDaylight Documentation"
- OpenDaylight Official Website : "Security Considerations"
- OpenDaylight Official Website : "OpenDaylight Security Advisories"
- OpenDaylight Official Website : "OpenDaylight Security Documentation"
- Official Ryu Website : "RYU SDN Framework"
- Network Heresy Website : "Is OpenFlow/SDN Good at Forwarding?"
- GitHub : "Offensive ONOS Repository"
- GitHub : "GAR-Project Repository"
- Narmox Spear : "Mininet Demo"
- GitHub : "Project DELTA Repository"
- OpenDaylight’s Gerrit : "List of Repositories"
- ResearchGate : "Topology poisoning attack scenario"
- Pratiknawale111’s Medium: "Malicious File Upload Vulnerability"

Planning

- Project Plan Maker Website : “Project Plan Maker”

This article has partly been translated with ChatGPT.

Future directions and Conclusion

1. Future directions

After writing this report, we still have more than a month of internship left, so we plan to look into other topics to enrich the research on SDN Security.

1.1. Explore GAR-Project

One of our planned tasks is to explore the **GAR-Project**[32], which **focuses on detecting Distributed Denial-of-Service (DDoS) attacks using AI**. This project leverages machine learning algorithms to identify and respond to DDoS threats in real-time. By integrating GAR-Project's AI-driven detection capabilities with our SDN environment, we aim to enhance our network's resilience against DDoS attacks, ensuring robust and proactive security measures.

1.2. Attack

We also plan to **attack OpenDaylight** by reproducing the vulnerability **CVE-2024-37018**. The CVE-2024-37018 vulnerability affects the OpenDaylight 0.15.3 controller and allows for **topology poisoning through API requests**. An application can manipulate the path that discovery packets take, potentially leading to **malicious control over network topology**. This type of attack could enable unauthorized users to **redirect or alter network traffic**, creating significant security risks in software-defined networking environments.
This task involves simulating the conditions to exploit this vulnerability, helping us understand its impact and potential exploitation methods.

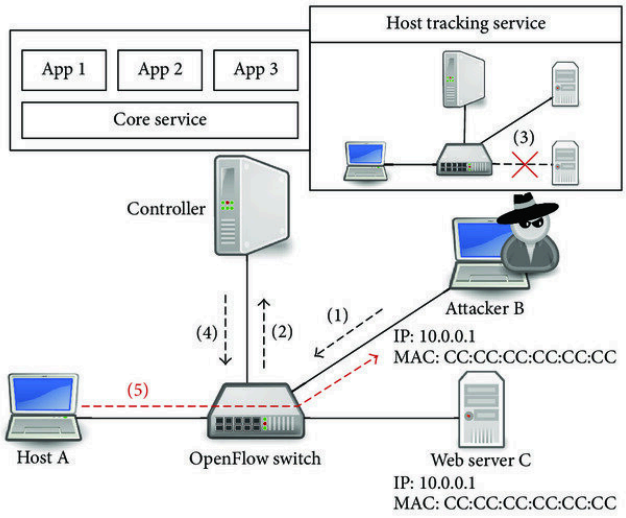

Figure 19 : Topology poisoning attack scheme

[32] https://github.com/GAR-Project/project

1.3. Performance testing

We plan to conduct **performance testing** on OpenDaylight. This involves evaluating the performance impact of various features by **simulating the same attack under different OpenDaylight configurations**. By comparing the results, we aim to understand how different settings impact the controller's ability to handle and mitigate attacks, ultimately optimizing its performance and resilience.

2. Conclusion

To conclude, during these ten weeks of internship at the Institute of Research, Innovation, and Digital Science and Technologies, I had the exceptional opportunity to put into practice my theoretical skills. These skills, in **cybersecurity**, **system administration**, and **networking** were acquired during my studies at the Computer Science department of Polytech Nice Sophia. Each day was an opportunity to **learn and grow**, both **professionally and personally**. I was able to confront my academic knowledge with real-world situations, which allowed me to better understand the current issues and challenges in the field of computer security.

This internship allowed me to discover **the world of research** and to work on concrete projects related to **network security**. I was able to deepen my knowledge of SDN technologies and DDoS attacks, as well as the tools and methods of defense against these attacks.

However, this experience also made me realize that **I prefer working in a team on concrete projects rather than focusing on writing research papers**. While research is essential for advancing knowledge, I realized that I am more fulfilled in an environment where I can directly see the impact of my work. Working on concrete projects allows me to leverage my technical skills and collaborate closely with other passionate professionals. I feel more comfortable in a technical role, and I am determined to pursue a career in this field, where I can continue to learn and grow.

I also had the honor of **participating in numerous events and conferences** organized by the Institute, which allowed me to **meet students and researchers** passionate about cybersecurity and to exchange experiences and projects with them.

Overall, satisfied with my internship experience, I am confident in my choice of a **minor in Cybersecurity next year** and I intend to direct my SI5 internship in this same field. This experience not only reinforced my academic choices but also gave me a clear vision of my professional future. I am convinced that cybersecurity is a field of the future that offers many opportunities for those who, like me, are **passionate about it**.