



**University of
Crete**

**Computer
Science
Department**

Bachelor Thesis

Mini-Internet Extention

Christos Papastamos

Advisor: Mr. Xenofontas Dimitropoulos

Herakleion, March 2024

ABSTRACT

The Mini-Internet platform is a network simulation platform, mainly used for teaching how networking works, developed at ETH Zurich. The platform is also used by our University as a project for the classes: HY335B (Computer Networks) and HY436 (Software Defined Networks). The platform uses **Docker containers** to simulate network devices (such as routers, switches, hosts, etc) and **Open vSwitch** links for the virtual connections between devices.

As part of my Thesis assignment, I focused on the development of two things, thus this report will be split into two parts: ***Web-Server Extension*** and ***BGP Monitoring using ARTEMIS Detector***.

Web-Server Extension

The platform's Web-server runs as a separate docker container using python's Flask library to implement the server. Adding to the original Web-Server the following features were implemented:

- Student Server
 - Login System
 - Password Changing
 - Rendezvous System
- Admin Server
 - Server Resources Monitor
 - Students and Teams Management
 - Docker Logs Access
 - Configuration Panel

BGP Monitoring using ARTEMIS Detector

As BGP Hijacks are still a major issue for world wide networking the best course of action is to use tools like ARTEMIS in order to detect and mitigate that kind of attacks. Using the Mini-Internet platform's simulated topology we can simulate BGP Hijack scenarios and use ARTEMIS detector in order to see how the network behaves.

Using EXA-BGP Monitors, each router logs the BGP Advertisements it receives so that ARTEMIS detector can use this information as input data for its detection algorithm. If a hijack is detected, ARTEMIS can automatically mitigate the attack so that the damage caused by the AS being hijacked is minimized.

Keywords: Docker, Open vSwitch, Web Development, Python, Flask, Jinja, HTML, BGP, BGP Hijacks, BGP Mitigation, ARTEMIS

DISCLAIMER

In this thesis report the mini-internet platform [1] is thoroughly explained. The mini-internet platform was made by Mr. Thomas Holterbach and his team at ETH in Zurich.

As part of this thesis several changes in the mini-internet platform were implemented. These changes are explained in their respective chapters: Chapter 2 and Chapter 3.

TABLE OF CONTENTS

ABSTRACT	i
DISCLAIMER	ii
TABLE OF CONTENTS	iii
1 Introduction	1
1.1 The Mini-Internet Platform [1]	1
1.1.1 Structural Elements	1
1.1.2 Network Devices	1
1.1.3 Platform Setup	2
1.1.4 Configuring the platform	2
1.2 BGP and BGP Hijacks	2
1.2.1 ARTEMIS	2
1.3 Trying the platform for yourself	3
1.3.1 Platform Setup	3
1.3.2 Connecting to the platforms devices	4
1.3.3 Default IP Addressing	4
2 Web Server	5
2.1 Introduction	5
2.2 Main changes in the Web-Server	7
2.3 Changes on the Students web-server	8
2.3.1 Log In System	8
2.3.2 Change Password Capability	8
2.3.3 Rendezvous System	9
2.4 The Admin web-server	10
2.4.1 Resource Monitor	10
2.4.2 AS Teams Management	11
2.4.3 Project Configuration	11
2.4.4 Docker Container Logs	15
3 BGP Hijacks	16
3.1 Introduction	16
3.2 Integrating ARTEMIS and EXA-BGP in the mini-internet platform	16

3.2.1	EXA-BGP Monitor	16
3.2.2	ARTEMIS Detector	18
3.3	The platform in action	20
3.3.1	Hijacking a prefix	20
3.3.2	Mitigating the hijack	23
3.3.3	Using ARTEMIS detector to mitigate the hijack	25
3.4	Use Case: BGP Hijacks Assignment	28
3.4.1	Assignment's Platform	28
3.4.2	Assignment's Execution	29
3.4.3	Assignment's Results	31
3.4.4	Acknowledgements	31
4	Summary	32
	BIBLIOGRAPHY	33
	LIST OF ABBREVIATIONS	34

1 Introduction

1.1 The Mini-Internet Platform [1]

The mini-internet platform is a teaching platform designed by Thomas Holterbach and his team of engineers at ETH Zurich university. This network simulation platform is designed to teach students how to configure local networks, ASes and inter-AS connections to achieve connectivity with the rest of the class.

The platform can be accessed and configured using simple ssh connections. For each AS, a proxy container handles the first layer of access including a script that allows for easy connection to each device. Every device can be configured using the basic bash CLI, while routers use the FRRouting CLI as default configuration tool.

The platform is used as the main project for the HY335b Course since the 2020 winter semester.

1.1.1 Structural Elements

Docker Containers

The platform uses Docker containers to simulate network devices (routers, switches, hosts, IXPs etc.). Every docker container inherits from a simple alpine docker image while having software installed to simulate each device. Specifically, the router containers are running FRRouting suite to simulate router functions while using standard routing protocols, the switches are running Open vSwitch that can simulate layer 2 functionality like Vlan tagging, trunk ports etc.

Open vSwitch Connections

Similar to the switches, the platform itself also uses Open vSwitch to simulate the links between devices. For each type of networks (internal, external, layer2 etc.) an Open vSwitch bridge exists with a port for each interface. Using OpenFlow rules each interface is connected to a docker container and is paired with the other end of the virtual connection that the platforms needs to create.

1.1.2 Network Devices

Hosts

The hosts of the platform can be found connected to each router of the AS (assuming the router has a host specified in the topology configuration). The hosts can be used to launch ping, traceroute and iperf3 tests. Python scripts can also be executed, making socket programming possible in the platform.

Switches

Switches can be found only in the layer 2 network (if exists) of an AS. The switches can be configured by changing the Open vSwitch bridges configuration (to add trunks or VLAN tags).

Routers

The Routers of the platform will be the most used device throughout this report since the configuration needed on them is much greater than any other device. Routers are configured through the FRR CLI, allowing the user to configure interfaces, assign IP addresses, use protocols like OSPF and BGP and more.

Other Devices

The platform also contains a variety of other devices such as the **Measurement** container, **VPN** plug in points, **DNS** servers, and a **Web-Server** whose structure and extension will be discussed in chapter 2.

1.1.3 Platform Setup

Setting Up the platform can easily be done with the help of the startup script provided in the platform's folder. Some of those scripts were modified to achieve the router monitoring ability.

1.1.4 Configuring the platform

The topology that the platform creates can be configured by editing the configuration files in the platform. These files include AS information, the routers configuration in each AS, the connectivity between them, the switches and hosts of the Layer 2 network and their connections etc. More information can be found in the mini-internet wiki.

1.2 BGP and BGP Hijacks

The complex network of the internet relies on a fundamental protocol known as the Border Gateway Protocol (BGP) to function seamlessly. BGP acts as the Internet's invisible traffic controller, precisely guiding data packets towards their designated destinations. However, lurking beneath this seemingly innocuous system lies a potential vulnerability: **BGP hijacking**.

BGP hijacking exploits the trust embedded within the protocol, enabling malicious actors to manipulate routing information. This manipulation essentially tricks the internet into rerouting traffic, creating a scenario similar to a meticulously planned detour on a crucial highway.

1.2.1 ARTEMIS

ARTEMIS is an open-source tool, that implements a defense approach against BGP prefix hijacking attacks. It is based on accurate and fast detection operated by the AS itself, by leveraging the pervasiveness of publicly available BGP monitoring services, and it enables flexible and fast mitigation of hijacking events. Compared to existing approaches/tools, ARTEMIS combines characteristics desirable to network operators such as comprehensiveness, accuracy, speed, privacy, and flexibility.

For the purpose of the integration of ARTEMIS in the Mini Internet platform, a new lightweight implementation in GO had to be created which provides BGP Hijack detection as well as mitigation for an AS.

1.3 Trying the platform for yourself

1.3.1 Platform Setup

The official platform of the mini-internet can be found in the ETHZ's GitHub repository: <http://mini-inter.net/>

The changes that were made as part of this thesis can be found in my personal GitHub repository: https://github.com/papastam/mini_internet_extention

For the rest of this thesis report, the locations, files and scripts that are mentioned can be found in my personal GitHub repository under the directories specified when used. A Linux operating system is required to execute the platform (preferably Ubuntu) containing the packages for *openvswitch-switch* and *docker*.

After cloning the repository, you can navigate in the files of the platform. The main platform files exist in the */platform* directory. Under this directory the following folders can be found:

- *cleanup* The folder containing the scripts needed to shut down and clean up the platform.
- *examples* The folder containing example configurations.
- *setup* The folder containing the scripts needed to start up the platform.
- *config* The folder containing the configuration of the topology that the platform will setup.
- *docker_images* The folder containing the docker files for the containers that are being used.
- *utils* The folder containing utility scripts that can be used once the platform is running.

Configuring the platform

Inside the *config* folder, the following files can be found:

- *AS_config.txt* The file containing information about each AS that will be setup.
- *aslevel_links(_student).txt* The file containing information about the links between ASes.
- *internal_links.txt* The file containing information about the router links inside an AS.
- *l2_(hosts,links,switches).txt* The files containing information about the Layer 2 topology.
- *router_config.txt* The file containing information about the routers of the AS.

When the platform is cloned from my GitHub repository, the default topology is a 4-AS topology, explained briefly in section 3.3.

Starting up the platform

Also contained in the platform files, the startup script can be used to start up the platform. Executing the script with administrator rights (*sudo ./startup.sh*) will begin building the platform. This might take some time, especially for bigger topologies.

1.3.2 Connecting to the platforms devices

The platform can be accessed through ssh using the port number to specify the AS to connect. The connection method that will be used for this report is by using docker's exec functionality. Docker's exec functionality allows you to execute commands inside a running container which makes it possible to run a complete bash (or VTYSH [2] for routers) session from the host of the platform. By using the -it flag we request an interactive terminal where the CLI will be executed.

Routers and Hosts

In order to access the router's CLI we need to execute the following command, specifying the container name with the following format: `<AS#>_<ROUTER_NAME>router`. For example, connecting to NORTH router of AS3, the container name should be `3_NORTHrouter`.

In the VTYSH [2] shell (Figure 1.1) we can execute commands such as `show run` and `show ip route`. More commands can be found in VTYSH's documentation: <https://docs.frouting.org/projects/dev-guide/en/latest/vtysh.html>.

The command we need to execute is: `docker exec -it 3_NORTHrouter vtysh`.

```
chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/platform$ docker exec -it 3_NORTHrouter vtysh
Hello, this is FRRouting (version 8.2.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
NORTH_router#
```

Figure 1.1: Connecting to a router

As for the platform's hosts we need a basic bash shell through a docker interactive terminal. For the hosts docker name, we can use the following format to connect to the specified router's host: `<AS#>_<ROUTER_NAME>host`. For example, connecting to AS1's EAST router host, the container name should be `1_EASThost`. When connected to a host (Figure 1.2) a simple bash shell is used to execute commands inside the host.

The command we need to execute is: `docker exec -it 1_EASThost bash`.

```
chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/platform$ docker exec -it 1_EASThost bash
root@EAST_host /> ls
bin dev etc home lib media mnt opt proc root run sbin srv sys tmp usr var
root@EAST_host />
```

Figure 1.2: Connecting to a host

1.3.3 Default IP Addressing

The format of the IP addresses in the platform when the AS are pre-configured follows the format:

- Each AS has the prefix `A.0.0.0/8` where `A` is their AS#. For example AS3 owns the prefix `3.0.0.0/8`.
- Each router has a router ID starting from 1. The loopback address of each router has the format `A.150+R.0.1` where `A` the AS# and `R` the router ID.
- Each host has the address `A.100+R.0.1` where `A` the AS# and `R` the router ID where the host is connected.

2 Web Server

2.1 Introduction

The platform's web-server runs in a docker container, separating it from the rest of the platform's topology. It uses python's flask [3] module to handle the back-end part of the server. Flask uses Jinja [4] web template engine to create the HTML files and tailwind CSS [4] for the styling of the elements of the front-end.

Coming out of the box, the platform has its own web-server, the functionalities of whom will be discussed in the rest of this introduction (section 2.1). The additions to the platform's web-server will be discussed from section 2.2 to 2.4 This Web-server has the following functionalities:

Matrix

The matrix (also used as the default index of the web-server) serves as a connectivity verification tool between ASes. In the matrix displayed, the connectivity between an AS in the vertical and an AS in the horizontal axis is specified with a cell having one of the following colors:

- Green Cell: The ASes are connected successfully.
- Yellow Cell: The ASes are connected but the Gao-Rexford [5] rules.
- Red Cell: The ASes are not connected.

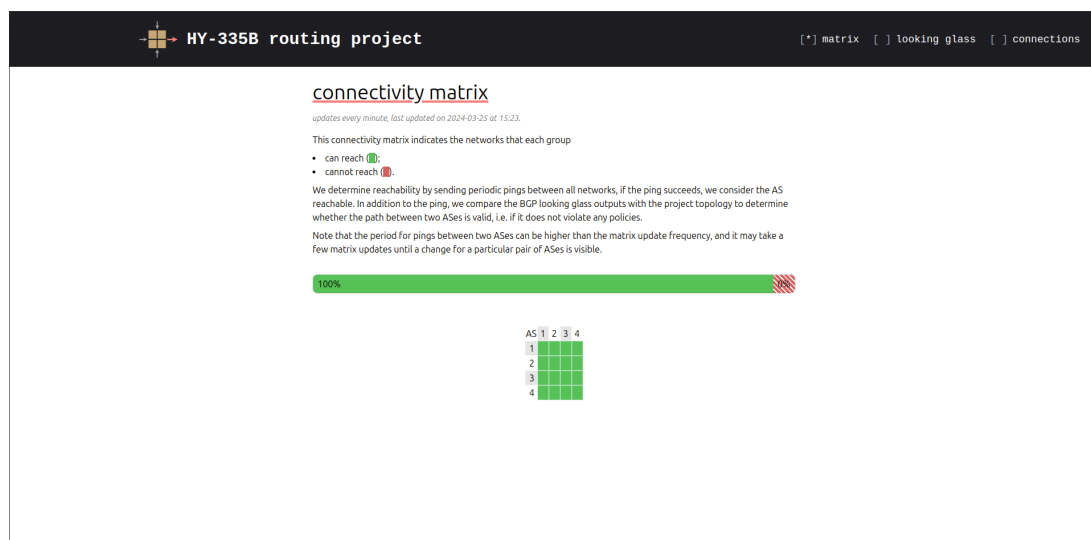


Figure 2.1: Connectivity matrix

In Figure 2.1 the connectivity matrix is displayed for a platform containing 4 ASes where every AS has complete connectivity with all other ASes.

Looking Glass

The BGP Looking Glass provided in the web-server serves as a BGP route debugging tool. When an AS and a router of the topology is selected, a snapshot of the routing table of the selected router can be seen in the looking glass output. This tool is useful mostly for routers the students have no access to.

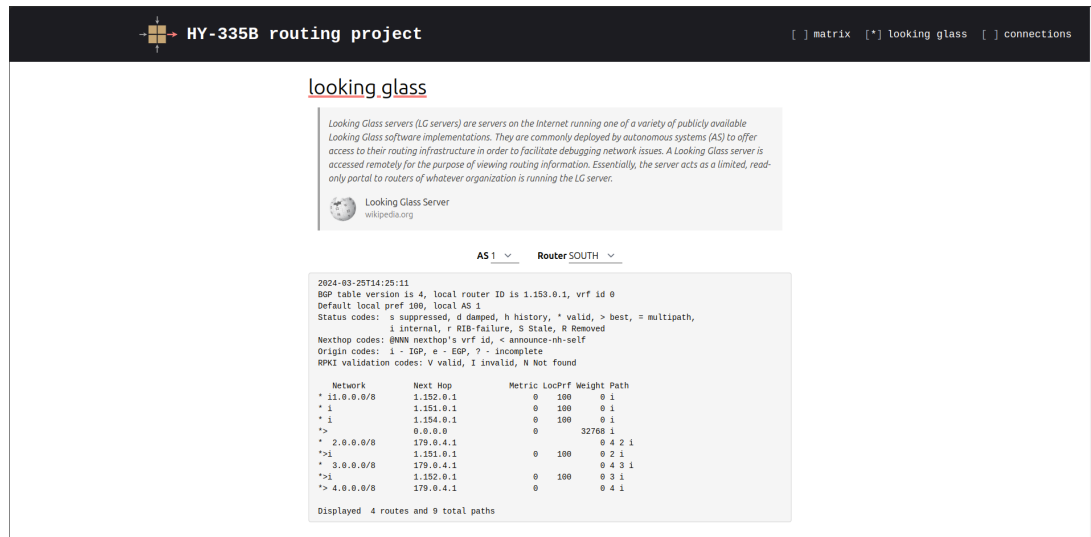


Figure 2.2: Looking Glass

In Figure 2.2 the routing table of SOUTH router from AS1 can be viewed.

Connections

In the connections tab, the students can distinguish the IPs they need to assign to their interfaces connected to external ASes. This is a helpful tool since it allows the user to filter through the (sometimes) big list of connections and help him find connections between specified ASes.

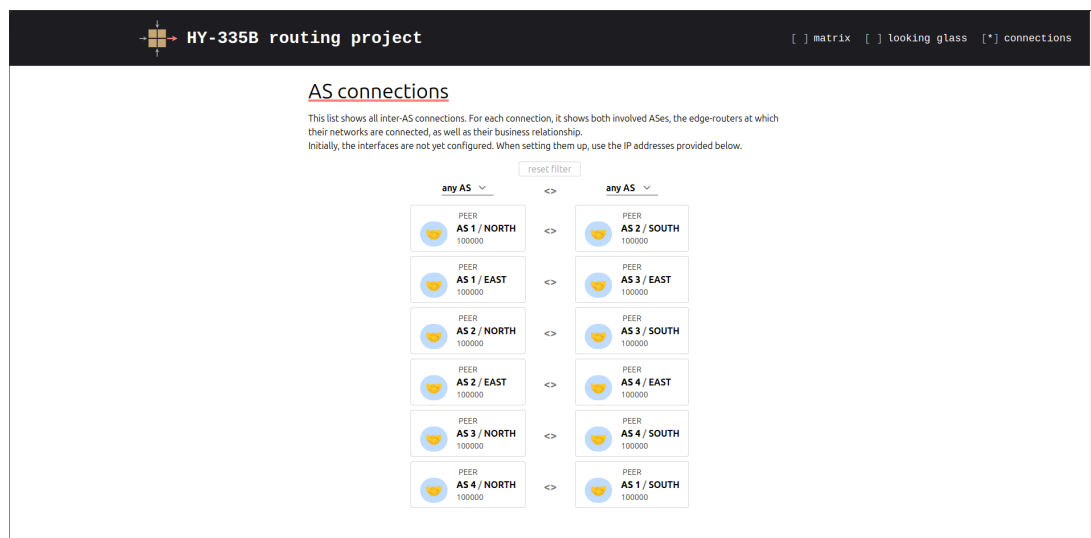


Figure 2.3: Connections

2.2 Main changes in the Web-Server

Database

A database running on Flask's integration of SQL Alchemy [6] was implemented alongside the web-server to contain information needed by other modules. Specifically, the systems that use the database are the following:

- Log-in system (Section 2.3.1)
- Resource Monitor (Section 2.4.1)
- AS Teams Management (Section 2.4.2)
- Rendezvous (Section 2.4.3 and 2.3.3)
- Project Configuration (Section 2.4.3)

Docker Pipe

A docker pipe is a named pipe created in the servers file system which is also mapped as a virtual file in the web-server docker container. This way the docker can communicate with the server and request commands executions like docker logs for other containers, password changes for the ssh containers, file editing for the passwords file that stores all current passwords etc. This file is a crucial part of the docker container since it serves as the only way of executing (regulated) administrative commands on the server as a docker container.

In Linux, a named pipe (FIFO) acts as a special file on the file system, distinct from temporary anonymous pipes. It allows multiple processes to communicate by reading from and writing to the same named location, following a first-in, first-out order. This enables programs to exchange data even if they aren't running concurrently.

On the other end of the pipe, a listener script is constantly checking for requests from the named pipe, and when a command is requested, it executes the according actions taking into account the arguments given by the docker container.

2.3 Changes on the Students web-server

2.3.1 Log In System

The login system is used to authenticate the teams based on their AS assigned to them during the start of the project. The authentication is possible through the password the team uses to log in to their platform's proxy, used to access all of the devices that need to be configured. In the web-server, the teams can access the rendezvous page when they are logged in and book or manage their reservations as well as the change password page.

The web-server uses Flask's Login Manager in association with Flask's Bcrypt implementation to store securely all the AS teams passwords.

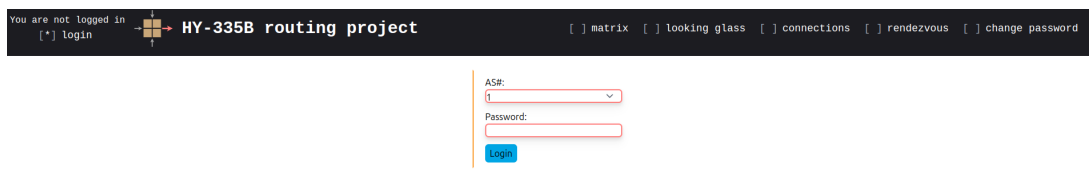


Figure 2.4: Students Login Page

2.3.2 Change Password Capability

As the password generated by the platform is a random 16-character string containing numbers and letters, the teams have the ability to change this password to something more memorable in order to access their AS proxy and log in to the web-server faster.

The students must be logged in to access this page!

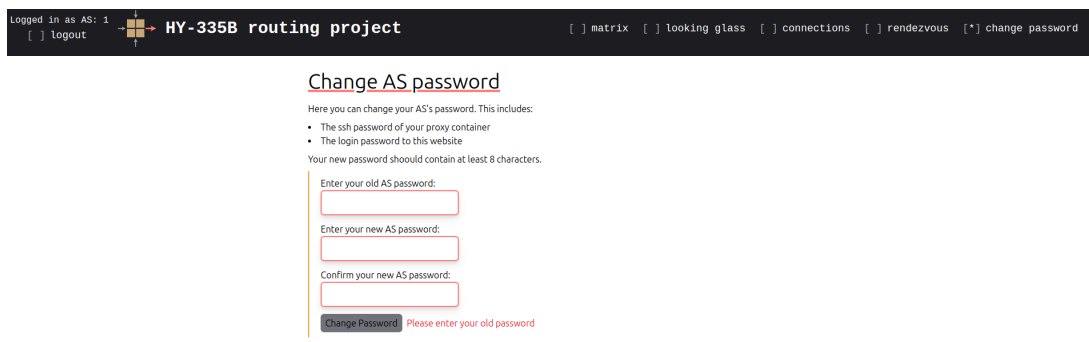


Figure 2.5: Change Password Page

When a password change is requested from a team, the server updates the database with the new password and requests a password change by the platform's server through the docker pipe (section 2.2). When the listener script receives the password change command, it updates the passwords file and changes the login password of the AS's proxy.

2.3.3 Rendezvous System

The rendezvous system was a system influenced by the way the project is implemented and graded for the course HY335b (Computer Networks). After finishing their project assignments, the students need to be orally examined by the courses Teaching Assistants. The scheduling of these exams need to be in alignment with all teams and the TAs need to know the exact schedule of the oral examination day. This rendezvous system can serve for other duties as well, like Helping Sessions planning etc.

The students need to be logged in to access this page. The rendezvous slots declaration will be discussed later on since the managing interface can only be found in the admin server (section 2.4.3).

In order for a team to book a rendezvous, they need to select the period for which they want to book (since there can be more than one booking events in one time). When the period is selected, the available rendezvous appear grouped by day. If a slot is booked by another team, the color of this slot's tile becomes red (for any other team that tries to book it) and a message informing the user that the rendezvous is already booked appears. Each team can book only one rendezvous per period.

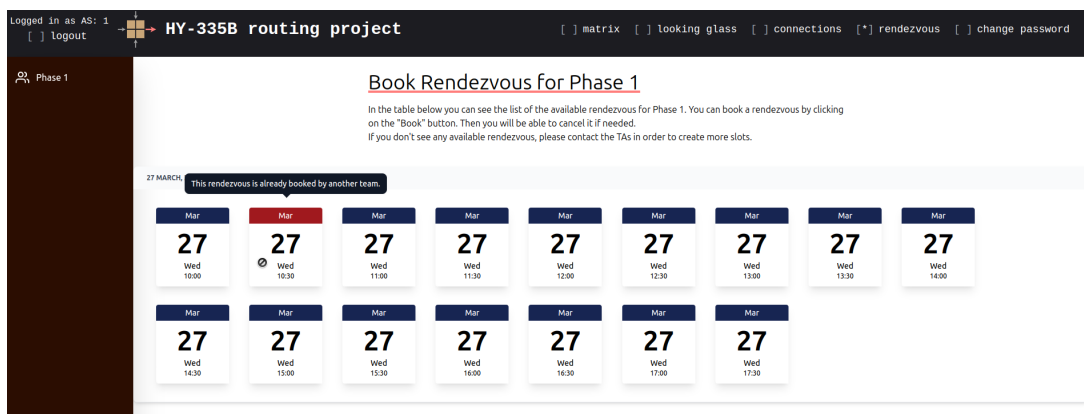


Figure 2.6: Rendezvous Page

Since many students may need to be reminded of their booking, when the rendezvous is booked, accessing the period informs them of the currently booked rendezvous. If they need to reschedule, by clicking the red Cancel button the booking is cancelled and a new one from the available rendezvous can be booked. The cancelled rendezvous slot also becomes available again.

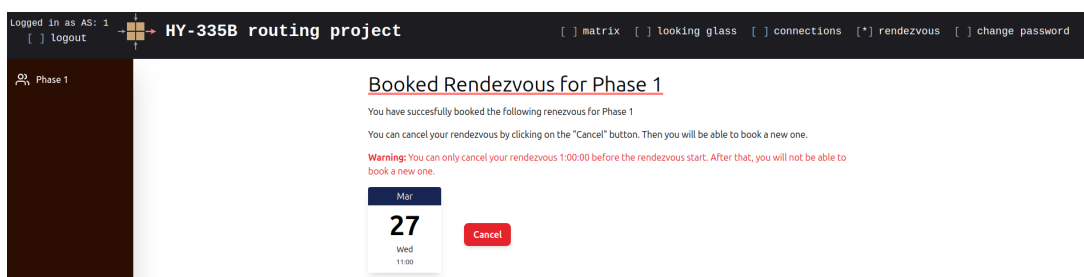


Figure 2.7: Booked Rendezvous

2.4 The Admin web-server

The administrators' website is running alongside the main web-server and can be accessed on port 8010. In order to access any of the administrators pages, the user must be logged in.

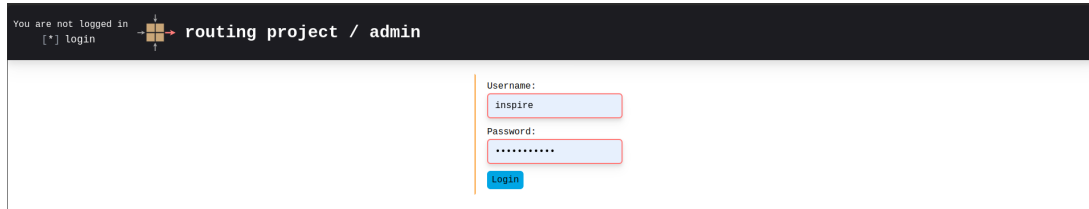
The image shows the Admin Login Page. At the top, it says "You are not logged in" and "login". Below this, there is a header "routing project / admin". The main content area contains a login form with two input fields: "Username:" with the value "inspire" and "Password:" with a masked password "*****". There is a "Login" button below the password field.

Figure 2.8: Admin Login Page

2.4.1 Resource Monitor

When an administrator is logged in he is greeted by the Resource Monitor. In this page the administrator can access the values of the CPU, RAM and DISK usage overtime by specifying the date range using the date pickers. Using the "Pull latest statistics" the display will fetch data from the selected start date till the most recent measurement.

By default the server measures the resources values in an interval of 1 minute.

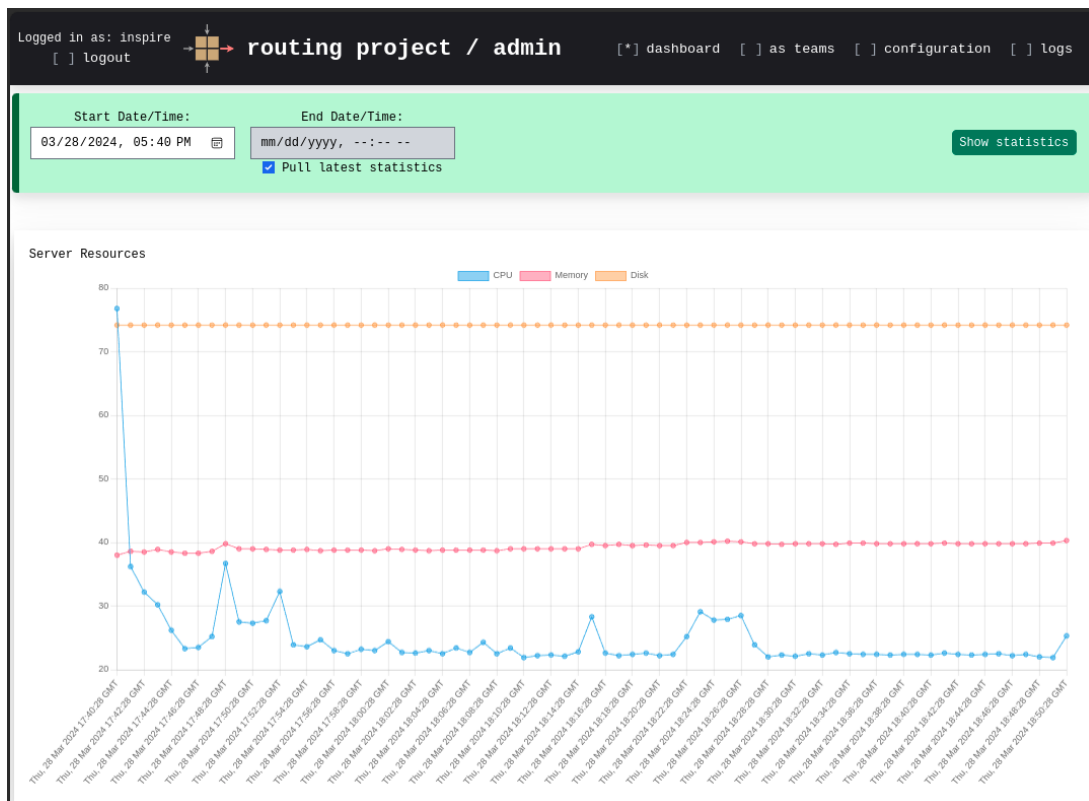
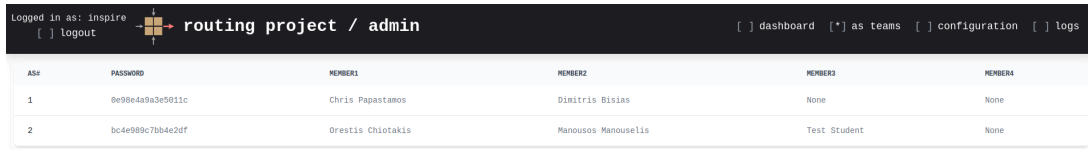


Figure 2.9: Admin Dashboard

2.4.2 AS Teams Management

Since the web-server can serve as a course management software, in the AS teams page, the administrator can quickly view the current information about each team's password and the team's members.



AS#	PASSWORD	MEMBER1	MEMBER2	MEMBER3	MEMBER4
1	0e98e4a9a3e5011c	Chris Papastamos	Dimitris Bisias	None	None
2	bc4e989c7bb4e2df	Orestis Chiotakis	Manousos Manouselis	Test Student	None

Figure 2.10: AS Teams Information

2.4.3 Project Configuration

The project configuration index is the main interface where the administrator can change the way the project's web-server behaves. If more modules get added to the web-server, their settings can also be easily added in this panel for easy and on-the-fly change.

On the left side panel, the user can select the category of configuration they would like to make. The categories are the following:

General Configuration

In this index general configurations about the web-server can be made. The implemented configuration options are the following:

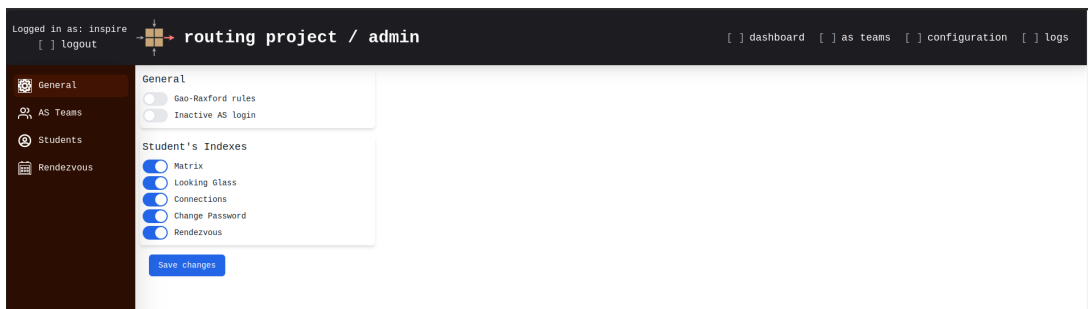


Figure 2.11: General Configuration

General

- Gao-Rexford rules [5]: Toggle whether or not the matrix will display yellow cells if an AS has connectivity but does not obey the Gao-Rexford rules [5].
- Inactive AS login: Enable login for inactive teams (*active/inactive teams will be explained in the AS Teams Configuration*).

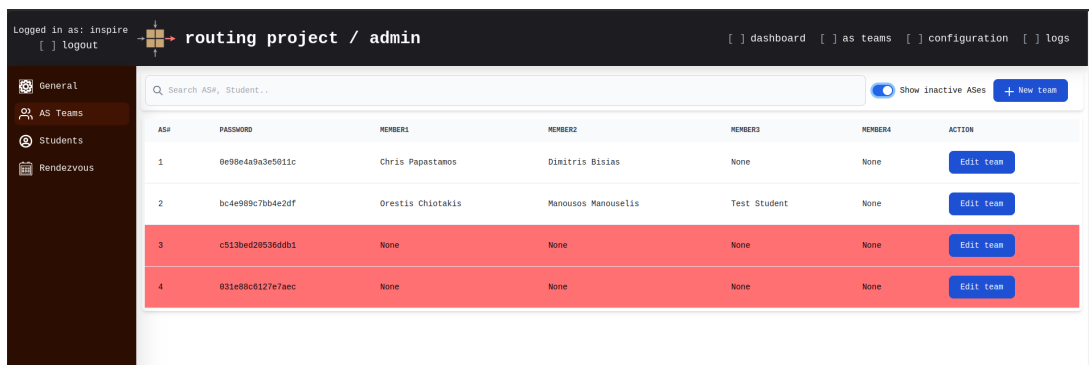
Student's Indexes: This section enables or disables the following indexes for the students web-server.

- Matrix
- Looking Glass
- Connections
- Change Password
- Rendezvous

AS Teams Configuration

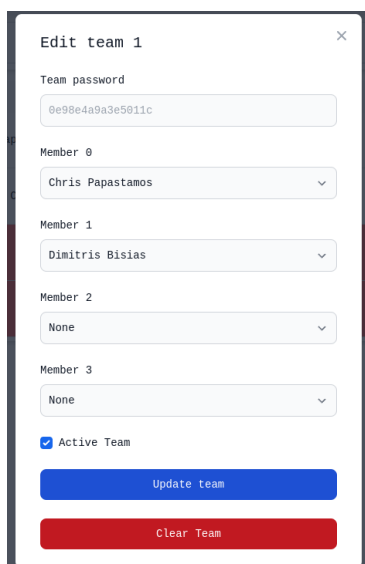
In the AS teams configuration the administrator can configure the teams that are currently practicing on the project. The teams can be active or inactive. When the page is initially loaded only the active AS teams can be viewed, but by using the "Show inactive ASes" switch (next to the search bar) the inactive ASes can be managed as well. An inactive AS team is used to signify an AS with no students working on it. This AS will not appear on the team tables and the login for this AS will be disabled (if specified by the option in the general configuration). When no students are assigned to an AS this AS becomes inactive. Another way of an AS becoming inactive is if an administrator deactivates it through the AS team edit panel.

Using the search bar the administrator can search for a team using the AS number or the assigned students.



AS#	PASSWORD	MEMBER1	MEMBER2	MEMBER3	MEMBER4	ACTION
1	0e98e4a9a3e5011c	Chris Papastamos	Dimitris Bisias	None	None	Edit team
2	bc4e989c7bb4e2df	Orestis Chiotakis	Manousos Manouselis	Test Student	None	Edit team
3	c513be420536db1	None	None	None	None	Edit team
4	031e88c0127e7aec	None	None	None	None	Edit team

Figure 2.12: Teams Configuration



Edit team 1

Team password

0e98e4a9a3e5011c

Member 0

Chris Papastamos

Member 1

Dimitris Bisias

Member 2

None

Member 3

None

☒ Active Team

[Update team](#)

[Clear Team](#)

By clicking the blue "Edit Team" button the Edit team panel comes up as can be seen in Figure 2.13. Through this panel the admin can change the team's password and the team's members. By using the "Active Team" button the admin can deactivate the team without changing the assigned students. The students will remain assigned to the team and cannot be assigned to another team unless they are removed from their initial team.

By using the "Clear Team" button the admin can clear all the students from the team making it inactive.

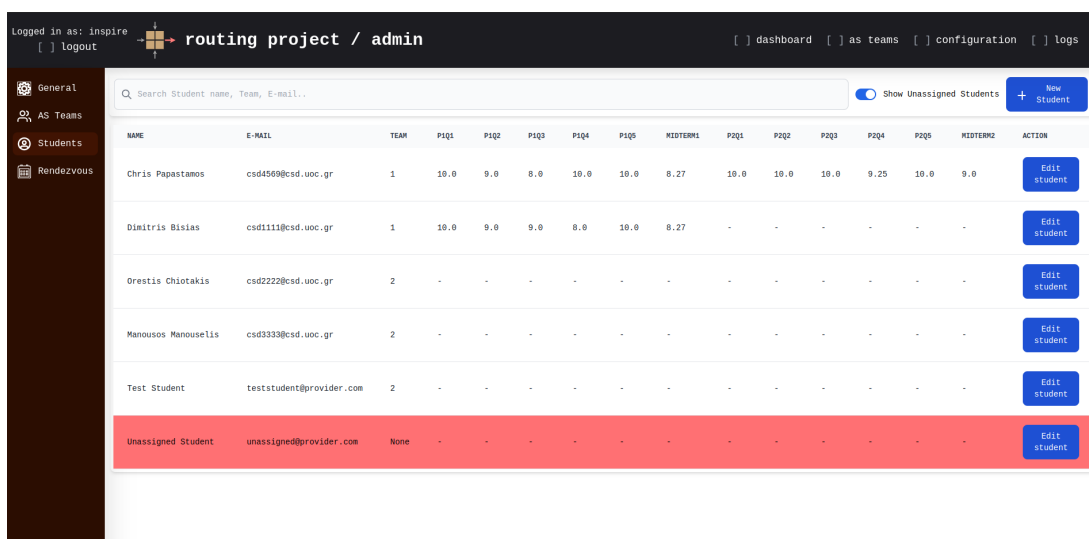
Figure 2.13: Edit AS Team Panel

Students Configuration

Through the Students Configuration index the administrator can manage individual students and grade them in each question of the project's assignment. When the index is initially loaded only the students assigned to a team can be viewed, but by using the "Show Unassigned Students" switch (next to the search bar) the unassigned students can be managed as well.

For the course of HY335b the grading is calculated based on 2 phases, each one with 5 questions and 2 midterms. The admin can grade the student and save the grades in the web-server for future use.

Using the search bar the admin can search for a student using the student's name, the assigned team or the student's e-mail.



NAME	E-MAIL	TEAM	P1Q1	P1Q2	P1Q3	P1Q4	P1Q5	MIDTERM1	P2Q1	P2Q2	P2Q3	P2Q4	P2Q5	MIDTERM2	ACTION
Chris Papastamos	csd4569@csd.uoc.gr	1	10.0	9.0	8.0	10.0	10.0	8.27	10.0	10.0	10.0	9.25	10.0	9.0	Edit student
Dimitris Bisias	csd1111@csd.uoc.gr	1	10.0	9.0	9.0	8.0	10.0	8.27	-	-	-	-	-	-	Edit student
Orestis Chiotakis	csd2222@csd.uoc.gr	2	-	-	-	-	-	-	-	-	-	-	-	-	Edit student
Manousos Manouselis	csd3333@csd.uoc.gr	2	-	-	-	-	-	-	-	-	-	-	-	-	Edit student
Test Student	teststudent@provider.com	2	-	-	-	-	-	-	-	-	-	-	-	-	Edit student
Unassigned Student	unassigned@provider.com	None	-	-	-	-	-	-	-	-	-	-	-	-	Edit student

Figure 2.14: Edit Student Panel



Edit Dimitris Bisias

Student's name
Dimitris Bisias

Student's E-mail
csd1111@csd.uoc.gr

p1q1 p1q2 p1q3 p1q4 p1q5
10.0 9.0 9.0 8.0 10.0

midterm1
8.27

p2q1 p2q2 p2q3 p2q4 p2q5
/10 /10 /10 /10 /10

midterm2
/10

Add Student

By clicking the blue "Edit Student" button the Edit Student panel comes up as can be seen on Figure 2.14. Through this panel the admin can change the student's name, email and assign grades to each part.

Figure 2.15: Students Configuration

Rendezvous Configuration

From the Rendezvous Configuration the administrator can configure the rendezvous available for booking through the students web-server.

First, the administrator needs to create a new period using the green "New Period" button. After the period is created, rendezvous can be added to it using either the manual "New Rendezvous" button to create a single rendezvous at a time or the "New Rendezvous Range" which creates a rendezvous given a time range and how much each rendezvous lasts. When the rendezvous are added, an item in the list will appear for each one of them.

After the rendezvous is created, the administrator can click the blue string mentioning who is booked to the rendezvous (next to the rendezvous icon) to edit it. Through the pop-up panel he can edit the time, the duration, the assigned period and the booked team. Also the rendezvous can be removed using the red "Delete Rendezvous" button. Using the search bar the admin can search for a rendezvous using the team number or the assigned students.

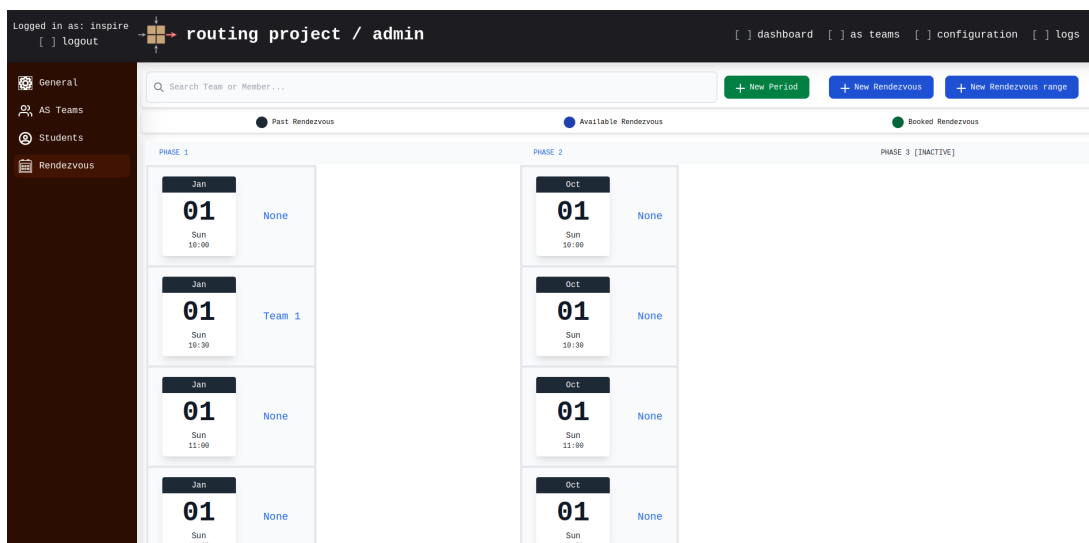
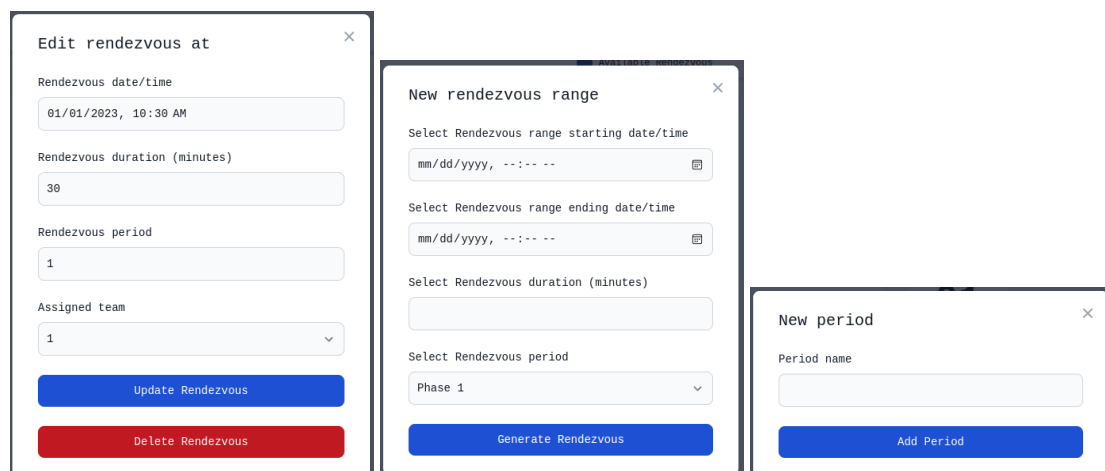


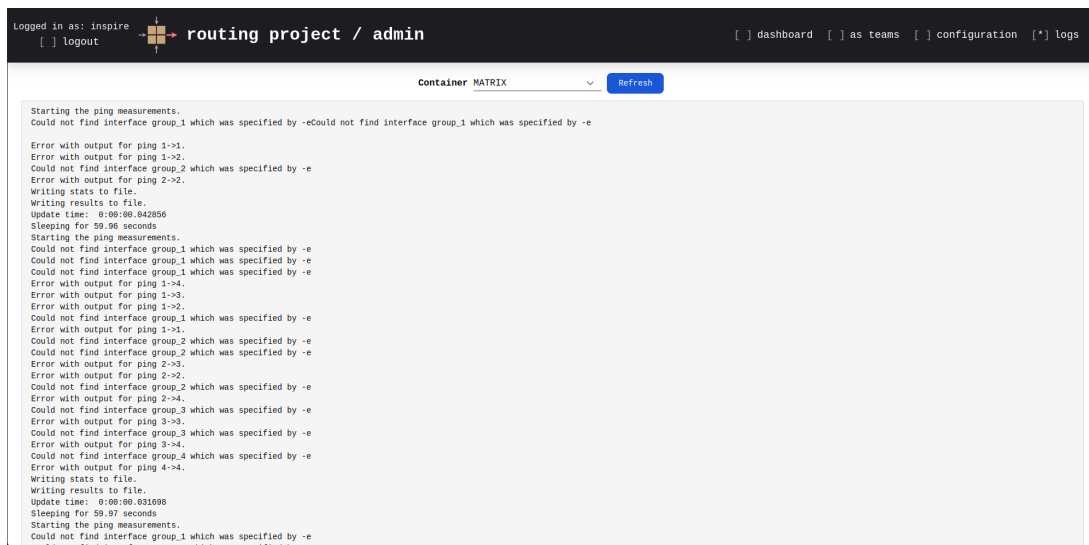
Figure 2.16: Rendezvous Configuration



2.4.4 Docker Container Logs

In the logs index of the admins web-server, the admin can monitor the docker logs of every running container in the server by selecting the desired container from the dropdown and clicking the "Refresh" button.

The web-server makes a request to the server through the Docker pipe and the files containing the most recent logs are updated for the selected container.



```
Logged in as: inspire
[ ] logout

routing project / admin
[ ] dashboard [ ] as teams [ ] configuration [*] logs

Container MATRIX
Refresh

Starting the ping measurements.
Could not find interface group_1 which was specified by -e
Error with output for ping 1->1.
Error with output for ping 1->2.
Could not find interface group_2 which was specified by -e
Error with output for ping 2->2.
Writing stats to file.
Writing results to file.
Update time: 0:00:00.042856
Sleeping for 59.96 seconds
Starting the ping measurements.
Could not find interface group_1 which was specified by -e
Could not find interface group_1 which was specified by -e
Could not find interface group_1 which was specified by -e
Error with output for ping 1->4.
Error with output for ping 1->3.
Error with output for ping 1->2.
Could not find interface group_1 which was specified by -e
Error with output for ping 1->1.
Could not find interface group_2 which was specified by -e
Could not find interface group_2 which was specified by -e
Error with output for ping 2->3.
Error with output for ping 2->2.
Could not find interface group_2 which was specified by -e
Error with output for ping 2->4.
Could not find interface group_3 which was specified by -e
Error with output for ping 3->3.
Could not find interface group_3 which was specified by -e
Error with output for ping 3->4.
Could not find interface group_4 which was specified by -e
Error with output for ping 4->4.
Writing stats to file.
Writing results to file.
Update time: 0:00:00.031098
Sleeping for 59.97 seconds
Starting the ping measurements.
Could not find interface group_1 which was specified by -e
```

Figure 2.17: Docker Logs Page

3 BGP Hijacks

3.1 Introduction

BGP hijacks are a devious trick on the Internet's routing system. They exploit the Border Gateway Protocol (BGP) to divert traffic intended for one AS to a different one controlled by the attacker. This can be like rerouting a delivery truck to a fake address.

These hijackings are a serious threat for a few reasons. First, BGP relies on trust between networks, making it vulnerable to someone forging routing information and appearing legitimate. A successful hijack can disrupt access to websites, online services, or even entire regions of the internet, causing widespread problems. Attackers have various motives for these detours, from stealing financial information to launching denial-of-service attacks or even censoring content. The difficulty in detecting these short-lived and targeted attacks makes them even more concerning. Despite mitigation strategies, BGP's inherent trust-based system makes it susceptible to manipulation, highlighting why BGP hijacks remain a significant threat to the global Internet's stability and security.

With BGP Hijacks being such an important vulnerability and threat for global routing, the INSPIRE Group (Led by Mr. Xenofontas Dimitropoulos [7]) invented an open-source software called ARTEMIS [8] which monitors BGP hijacks in real time, and if one is detected, ARTEMIS mitigates the hijack so that the AS recovers as fast as possible.

During a video conference about the mini-internet platform with MR Thomas Holterbach (the lead creator of the mini-internet platform), he recommended that a good addition to the mini-internet platform would be the ARTEMIS system. That way, the platform could be used to teach students how BGP prefix announcement works and the dangers of BGP hijacks as well as be used for experimental runs where one could perceive how the network and the ARTEMIS tool reacts to a BGP hijack.

3.2 Integrating ARTEMIS and EXA-BGP in the mini-internet platform

For the bgp monitoring, hijack detection and real-time mitigation functionality of the platform, EXA-BGP [9] and ARTEMIS detector were implemented, operating alongside the platform:

3.2.1 EXA-BGP Monitor

For the integration of ARTEMIS in the mini-internet platform, the BGP updates of each router needed to be monitored so that they could later be passed to ARTEMIS for processing. This functionality is served by using EXA-BGP [9] monitors. When an AS is set to be "Monitored" on the AS configuration of the platform (explained in section 1.1.4) a docker container running EXA-BGP monitor is generated which is connected to all of the AS's routers. With the configuration given to the monitor containers, they create a iBGP session with the connected routers, which means that they will be updated for every eBGP update the routers receive.

When the monitor receives these updates a parser, written in python, processes the BGP updates and appends them to a log file in CSV format. An example of a monitor log file can be seen in Figure 3.1.

All monitor files can be located in the following directory: `/platform/groups/exabgp_monitor/output`.

```
chris@ChrisPC-Ubuntu:~/Classes/mininet_extensions/platform/groups/exabgp_monitor/output$ tail -n 100 -f 1_output.csv
4.0.0.0/8|4|1|4 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.4.1': [{'nlri': '4.0.0.0/8'}]}}}1713556348.0184116
4.0.0.0/8|4|1|4 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.4.1': [{'nlri': '4.0.0.0/8'}]}}}1713556348.6266608
2.0.0.0/8|2|1|2 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.1.2': [{'nlri': '2.0.0.0/8'}]}}}1713556358.1481228
3.0.0.0/8|3|1|3 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.5.2': [{'nlri': '3.0.0.0/8'}]}}}1713556364.5559742
2.0.0.0/8|2|1|2 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.151.0.1', 'cluster-list': ['1.152.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlri': '2.0.0.0/8'}]}}}1713556465.5294173
2.0.0.0/8|2|1|2 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.151.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlri': '2.0.0.0/8'}]}}}1713556465.5297096
3.0.0.0/8|3|1|3 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.151.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlri': '3.0.0.0/8'}]}}}1713556465.5302677
4.0.0.0/8|4|1|4 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.153.0.1', 'cluster-list': ['1.152.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlri': '4.0.0.0/8'}]}}}1713556465.5306559
3.0.0.0/8|3|1|3 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlri': '3.0.0.0/8'}]}}}1713556465.5310354
4.0.0.0/8|4|1|4 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.153.0.1', 'cluster-list': ['1.151.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlri': '4.0.0.0/8'}]}}}1713556465.5313423
2.0.0.0/8|2|1|2 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.151.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlri': '2.0.0.0/8'}]}}}1713556465.575619
3.0.0.0/8|3|1|3 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlri': '3.0.0.0/8'}]}}}1713556465.5759866
4.0.0.0/8|4|1|4 |exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.153.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlri': '4.0.0.0/8'}]}}}1713556465.57633
```

Figure 3.1: Output of the `tail` command

The CSV file uses the `"]"` character as a delimiter so that the output can be easily readable. The monitor logs follow the following format:

PREFIX | ORIGIN AS | PEER AS | PATH | COLLECTOR | PEER | MESSAGE TYPE | MESSAGE | TIMESTAMP

Integration drawbacks

Due to the nature of the BGP protocol, a router advertising the AS's prefix needs to have that exact prefix in their routing table. For that reason, and because the big /8 network that is used for each AS does not exist as a whole in any routers routing table, a static route for the prefix routed to a Null interface needs to be present. The route to a Null interface could become an issue but due to Longest Prefix Match, if a more specific route exists for a packets destination IP address, it will be routed to the correct destination.

The drawback of the static route to the AS's prefix is that the border routers (routers connected to neighbor ASes) know already how to reach the AS's prefix. For that reason, when an advertisement for the AS's prefix arrives at the border routers (a.k.a. a hijack) the router discards the advertisement since it is not needed. The result of this discard is that the monitor will never be informed about the hijack.

There are two workarounds for this problem:

- The first way to detect hijacks is to remove this static route when the detection is happening. This workaround has the drawback that while the AS is in hijack detection mode (where the static route is removed) it will not advertise its prefix.
- The second way to detect a hijack is to read advertisements from all the topology's AS which will receive the advertisement but will not discard it due to the static route being for their prefix and not

for the one which is currently monitored. The drawback of this workaround is that the administrator who wants to detect the hijacks needs to have access to all other AS monitor logs.

For the rest of this thesis the second workaround will be followed due to the simplicity of accessing every other ASes monitor log files. In a real world scenario, where ASes may be competing corporations, this could be difficult to be arranged and for this reason the first approach should be considered.

3.2.2 ARTEMIS Detector

For the integration of ARTEMIS in the mini-internet platform, a lightweight implementation of ARTEMIS needed to be implemented in order for a server to be able to handle mini-internet and ARTEMIS at the same time (ARTEMIS is a heavyweight multi-docker application, same as mini-internet). For this purpose a much simpler implementation of ARTEMIS was created in GoLang [10] (also known as GO). The GO implementation of ARTEMIS takes as input the BGP updates of a number of routers of the same AS and detects if a hijack has happened.

The detector takes also requires as an input the prefixes that each AS is allowed to advertise. This input should come in a CSV file (with ”|” as a delimiter) with the following format:

```
PREFIX | ORIGIN AS
```

The prefixes file is generated by the platform and exists at the following directory: */platform/groups/ex-abgp_monitor/as_prefixes.csv*.

ARTEMIS Detector Functionality

The detector can be found in the */artemis_go* directory.

The ARTEMIS detector can be ran using the following syntax:

```
./artemis_detector <detector_mode> {detector_flags}
```

The detector modes and their respective arguments are the following:

detect_all

The *detect_all* mode detects hijacks for any AS in the given prefix file. The arguments for this mode are the following:

- *--updates <File(.csv)/Dir>: The BGP updates file/directory, typically the monitor's log or the directory containing all of them.*
- *--input_type <"file"/"directory">: The type of input you are providing the tool with (default: "file").*
- *--prefixes <prefixes_file.txt>: The prefixes file containing the legal prefixes of each AS.*
- *--output <output_file(.csv)>: The file where the detector will write the final detected hijacks (will be created if not exists).*

detect_as

The *detect_as* mode detects hijacks targeted for the prefix of the AS specified with the *-asn* flag. The arguments for this mode are the following:

- *--updates <File(.csv)/Dir>*: The BGP updates file/directory, typically the monitor's log or the directory containing all of them.
- *--input_type <"file"/"directory">*: The type of input you are providing the tool with (default: "file").
- *--prefixes <prefixes_file.txt>*: The prefixes file containing the legal prefixes of each AS.
- *--output <output_file(.csv)>*: The file where the detector will write the final detected hijacks (will be created if not exists).
- *--asn <AS#>*: The AS number of the AS that the detector will execute the detection.

active

The *active* mode repeatedly detects hijacks targeted for the prefix of the AS specified with the *-asn* flag in an interval (in minutes) specified with the *-interval* flag. If a mitigation script is specified using the *-mitigation_script_path* when a hijack is detected for a prefix belonging to the specified AS, the detector will call the mitigation script for the hijacked prefix. The mitigation script must be a bash script accepting an AS number with the *-a* flag and the prefix to mitigate with the *-p* flag. The platform's mitigation script can be found in the */platform/utls/bgp_hijack/mitigate.sh*.

The arguments for the *active* mode are the following:

- *--updates <File(.csv)/Dir>*: The BGP updates file/directory, typically the monitor's log or the directory containing all of them.
- *--input_type <"file"/"directory">*: The type of input you are providing the tool with (default: "file").
- *--prefixes <prefixes_file.txt>*: The prefixes file containing the legal prefixes of each AS.
- *--output <output_file(.csv)>*: The file where the detector will write the final detected hijacks (will be created if not exists).
- *--asn <AS#>*: The AS number of the AS that the detector will execute the detection.
- *--interval <minutes>*: The waiting time between each run of the detector.
- *--mitigation_script_path <mitigation.sh>*: The path to the mitigation script.

3.3 The platform in action

For the presentation of the platform i will use a simple topology of 4 ASes, each one containing 4 routers. The internal topology of each AS can be seen in Figure 3.2. The EXA-BGP monitor container is connected to all routers. The AS orientation is present to help distinguish which routers are connected to each other

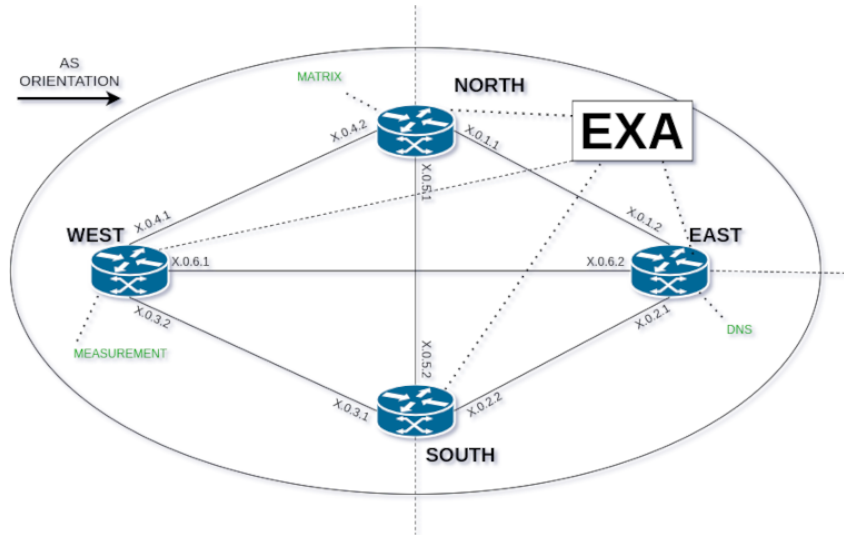


Figure 3.2: The internal topology of each AS

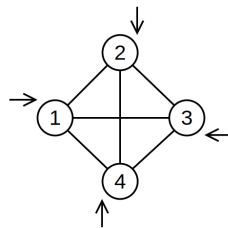


Figure 3.3: The AS topology of the platform

The four ASes are connected to each other in a full mesh topology according to Figure 3.3. The orientation of each AS is indicated with the small arrow next to every AS.

3.3.1 Hijacking a prefix

Lets now take a look in the monitor's log at AS 1. The monitor's files can be found in the platform's files since they are regular files mounted in the docker as a volume.

The monitor files can be located at: `/platform/groups/exabgp_monitor/output/<AS#>_output.csv`.

```
chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/platform/groups/exabgp_monitor/output$ tail -n 100 -f 1_output.csv
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.4.1': [{'nlri': '4.0.0.0/8'}]}}]|1713556348.0184116
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.4.1': [{'nlri': '4.0.0.0/8'}]}}]|1713556348.6266608
2.0.0.0/8|2|1|2 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.1.2': [{'nlri': '2.0.0.0/8'}]}}]|1713556358.1481228
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.5.2': [{'nlri': '3.0.0.0/8'}]}}]|1713556364.5559742
2.0.0.0/8|2|1|2 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.151.0.1', 'cluster-list': ['1.152.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlri': '2.0.0.0/8'}]}}]|1713556465.5294173
2.0.0.0/8|2|1|2 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.151.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlri': '2.0.0.0/8'}]}}]|1713556465.5297096
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.152.0.1', 'cluster-list': ['1.151.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlri': '3.0.0.0/8'}]}}]|1713556465.5302677
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.153.0.1', 'cluster-list': ['1.152.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlri': '4.0.0.0/8'}]}}]|1713556465.5306559
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.152.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlri': '3.0.0.0/8'}]}}]|1713556465.5310354
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.153.0.1', 'cluster-list': ['1.151.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlri': '4.0.0.0/8'}]}}]|1713556465.5313423
2.0.0.0/8|2|1|2 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.151.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlri': '2.0.0.0/8'}]}}]|1713556465.575619
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.152.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlri': '3.0.0.0/8'}]}}]|1713556465.5759866
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'originator-id': '1.153.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlri': '4.0.0.0/8'}]}}]|1713556465.57633
```

Figure 3.4: Output of the `tail` command

Initially, the monitors of the files contain the first advertisements sent by the routers after the platform started up (Figure 3.4). Using the `tail` command followed by `-f` for follow mode we can keep an eye on the logs in a separate terminal window.

We will now execute a hijack from AS 2 for the prefix of AS 3 `3.0.0.0/8`. This hijack will mainly affect AS 1 and AS 4 since the hijack will not be accepted by AS 3 (due to the static route, as explained in section 3.2.1).

In order to execute a hijack we can use the hijack with the following arguments:

- `-a <AS#>` The as number that will launch the attack.
- `-p <IP_ADDR>` The prefix that will be hijacked.
- `-t <SEC>` The duration of the attack in seconds.

The script can be located at: `/platform/utls/bgp_hijack/hijack.sh`.

```
chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/platform/utls/bgp_hijack$ ./hijack.sh -a 2 -p 3.0.0.0/8 -t 4
(AS2)Hijacking 3.0.0.0/8 for 4 minute(s) on 2_WESTrouter
(AS2)Hijacking 3.0.0.0/8 for 4 minute(s) on 2_SOUTHrouter
(AS2)Hijacking 3.0.0.0/8 for 4 minute(s) on 2_EASTrouter
(AS2)Hijacking 3.0.0.0/8 for 4 minute(s) on 2_NORTHrouter
(AS2)4 minutes Remaining
```

Figure 3.5: Executing the hijack script

Looking back to the monitor (Figure 3.6) files we can see the 3 new (latest) entries. These advertisements can be traced back to AS 2 which advertised the prefix `3.0.0.0/8` that belongs to AS 3 (as explained in section 1.3.3):

In order to see what this hijack caused to the network, we can execute a ping measurement from one of

```

chris@chrisPC-Ubuntu:~/Classes/mint_internet_extention/platform/groups/exabgp_monitor/output$ tail -n 100 -f 1_output.csv
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'lo
cal-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.4.1': [{'nlri': '4.0.0.0/8'}]}}]|1713556348.0184116
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'lo
cal-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.4.1': [{'nlri': '4.0.0.0/8'}]}}]|1713556348.0266608
2.0.0.0/8|2|1|2 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'lo
cal-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.1.2': [{'nlri': '2.0.0.0/8'}]}}]|1713556358.1481228
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'lo
cal-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.5.2': [{'nlri': '3.0.0.0/8'}]}}]|1713556364.5559742
2.0.0.0/8|2|1|2 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.151.0.1', 'cluster-list': ['1.152.0.1'], 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlr
i': '2.0.0.0/8'}]}}]|1713556465.5294173
2.0.0.0/8|2|1|2 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.151.0.1', 'cluster-list': ['1.153.0.1'], 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlr
i': '2.0.0.0/8'}]}}]|1713556465.5297096
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.152.0.1', 'cluster-list': ['1.151.0.1'], 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlr
i': '3.0.0.0/8'}]}}]|1713556465.5302677
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.153.0.1', 'cluster-list': ['1.152.0.1'], 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlr
i': '4.0.0.0/8'}]}}]|1713556465.5306559
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.152.0.1', 'cluster-list': ['1.153.0.1'], 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlr
i': '3.0.0.0/8'}]}}]|1713556465.5310354
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.153.0.1', 'cluster-list': ['1.151.0.1'], 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlr
i': '4.0.0.0/8'}]}}]|1713556465.5313423
2.0.0.0/8|2|1|2 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.151.0.1', 'cluster-list': ['1.154.0.1'], 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlr
i': '2.0.0.0/8'}]}}]|1713556465.575619
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.152.0.1', 'cluster-list': ['1.154.0.1'], 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlr
i': '3.0.0.0/8'}]}}]|1713556465.5759866
4.0.0.0/8|4|1|4 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.153.0.1', 'cluster-list': ['1.154.0.1'], 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlr
i': '4.0.0.0/8'}]}}]|1713556465.57633
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'local-prefer
ence': 100}, 'announce': {'ipv4 unicast': {'179.0.1.2': [{'nlri': '3.0.0.0/8'}]}}]|1713556545.8907251
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.151.0.1', 'cluster-list': ['1.153.0.1'], 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlr
i': '3.0.0.0/8'}]}}]|1713556545.9444582
3.0.0.0/8|3|1|3 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'lo
cal-preference': 100}, 'originator-id': '1.151.0.1', 'cluster-list': ['1.154.0.1'], 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlr
i': '3.0.0.0/8'}]}}]|1713556545.945165

```

Figure 3.6: Output of the `tail` command

the hosts of AS 1 who sits in between the two conflicting AS advertisements.

Connecting to the host (as mentioned in section 1.3.2) and executing a ping to 3.101.0.1 (the IP of the AS 3's host, connected to NORTH router) it can be noticed that the AS 1 host does not have connectivity to the AS 3 host.

```

root@WEST_host /> ping 3.151.0.1
PING 3.151.0.1 (3.151.0.1) 56(84) bytes of data.
^C
--- 3.151.0.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3068ms
root@WEST_host /> 

```

Figure 3.7: Pinging AS3 from AS1

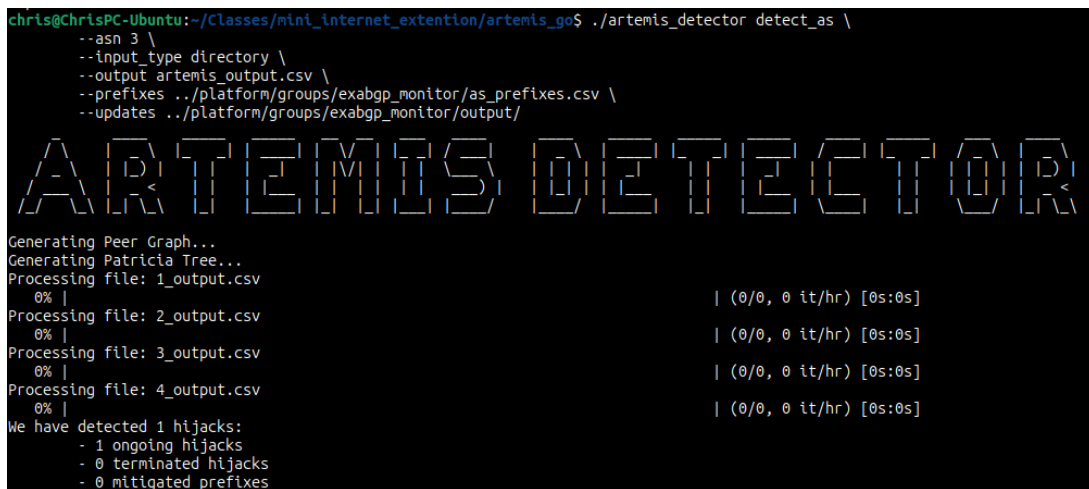
Lets now execute ARTEMIS detector on the monitor logs to figure out if any hijacks were detected:

We want to detect hijacks for a specific AS' prefix (AS3) and we are going to provide the detector with a directory of monitor logs that contains logs from all 4 ASes. This directory is: `../platform/groups/exabgp_monitor/output/`. The AS prefixes file is also required so that the detector knows what prefix each AS owns.

The command we are going to use is the following:

```
./artemis_detector detect_as
--asn 3
--input_type directory
--output artemis_output.csv
--prefixes ../platform/groups/exabgp_monitor/as_prefixes.csv
--updates ../platform/groups/exabgp_monitor/output/
```

The ARTEMIS detector's modes and arguments are explained in section 3.2.1.



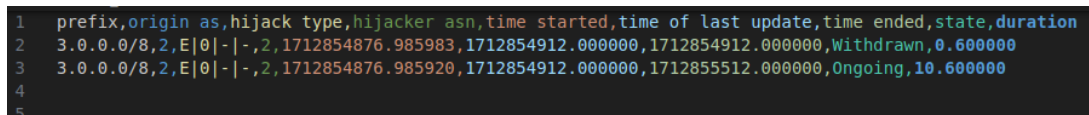
```
chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/artemis_go$ ./artemis_detector detect_as \
--asn 3 \
--input_type directory \
--output artemis_output.csv \
--prefixes ../platform/groups/exabgp_monitor/as_prefixes.csv \
--updates ../platform/groups/exabgp_monitor/output/

ARTEMIS DETECTOR

Generating Peer Graph...
Generating Patricia Tree...
Processing file: 1_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 2_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 3_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 4_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
We have detected 1 hijacks:
- 1 ongoing hijacks
- 0 terminated hijacks
- 0 mitigated prefixes
```

Figure 3.8: ARTEMIS detector in detect_as mode

As we can see in Figure 3.8 a hijack was detected! Lets check the *artemis_output.csv* file for more information:



```
1 prefix,origin as,hijack type,hijacker asn,time started,time of last update,time ended,state,duration
2 3.0.0.0/8,2,E|0|-|-,2,1712854876.985983,1712854912.000000,1712854912.000000,Withdrawn,0.600000
3 3.0.0.0/8,2,E|0|-|-,2,1712854876.985920,1712854912.000000,1712855512.000000,Ongoing,10.600000
4
5
```

Figure 3.9: The output file of the ARTEMIS detector

In the detector output (Figure 3.9) we can confirm that the monitors detected an Exact prefix hijack on the prefix 3.0.0.0/8, advertised from AS2.

3.3.2 Mitigating the hijack

When the attacked AS detects the hijack, they AS should take action to recover from this attack. One of the simplest ways that the AS can mitigate the attack is to re-advertise its prefix by advertising from every router the two subnets that combine to make this prefix. These subnets in the case of 3.0.0.0/8 are 3.0.0.0/9 and 3.128.0.0/9.

In order to mitigate a prefix we can use the mitigate script with the following arguments:

- *-a <AS#>* The as number that will mitigate the prefix.
- *-p <IP_ADDR>* The prefix that the AS wants to mitigate.

- `-t <SEC>` The duration of the mitigation (unused for indefinite mitigation).

The script can be located at: `/platform/utls/bgp_hijack/mitigate.sh`.

```
chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/platform/utls/bgp_hijack$ ./mitigate.sh -a 3 -p 3.0.0.0/8
(AS3)Advertising prefixes 3.0.0.0/9 and 3.128.0.0/9 from router
(AS3)Advertising prefixes 3.0.0.0/9 and 3.128.0.0/9 from router
(AS3)Advertising prefixes 3.0.0.0/9 and 3.128.0.0/9 from router
(AS3)Advertising prefixes 3.0.0.0/9 and 3.128.0.0/9 from router
(AS3)Mitigating indefinitely
```

Figure 3.10: The mitigation script

Taking a look to AS 1's monitor output (Figure 3.11) we see the 8 newest advertisements that are part of the mitigation:

```
chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/platform/groups/exabgp_monitor/output$ tail -n 100 -f 1 output.csv
4.0.0.0/8|4|14 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.4.1': [{'nlrt': '4.0.0.0/8'}]}]}|1713556348.0184116
4.0.0.0/8|4|14 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.4.1': [{'nlrt': '4.0.0.0/8'}]}]}|1713556348.6266608
2.0.0.0/8|2|12 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.1.2': [{'nlrt': '2.0.0.0/8'}]}]}|1713556358.1481228
3.0.0.0/8|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.5.2': [{'nlrt': '3.0.0.0/8'}]}]}|1713556364.5559742
2.0.0.0/8|2|12 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.151.0.1', 'cluster-list': ['1.152.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlrt': '2.0.0.0/8'}]}]}|1713556465.5294173
2.0.0.0/8|2|12 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.151.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlrt': '2.0.0.0/8'}]}]}|1713556465.5297096
3.0.0.0/8|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.151.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/8'}]}]}|1713556465.5302677
4.0.0.0/8|4|14 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.153.0.1', 'cluster-list': ['1.152.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlrt': '4.0.0.0/8'}]}]}|1713556465.5306559
3.0.0.0/8|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/8'}]}]}|1713556465.5310354
4.0.0.0/8|4|14 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.153.0.1', 'cluster-list': ['1.151.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlrt': '4.0.0.0/8'}]}]}|1713556465.5313423
2.0.0.0/8|2|12 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.151.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlrt': '2.0.0.0/8'}]}]}|1713556465.575619
3.0.0.0/8|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/8'}]}]}|1713556465.5759866
4.0.0.0/8|4|14 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [4]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.153.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.153.0.1': [{'nlrt': '4.0.0.0/8'}]}]}|1713556465.57633
3.0.0.0/8|2|12 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.1.2': [{'nlrt': '3.0.0.0/8'}]}]}|1713556545.8907251
3.0.0.0/8|2|12 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.151.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlrt': '3.0.0.0/8'}]}]}|1713556545.9444582
3.0.0.0/8|2|12 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [2]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.151.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.151.0.1': [{'nlrt': '3.0.0.0/8'}]}]}|1713556545.945165
3.0.0.0/9|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.5.2': [{'nlrt': '3.0.0.0/9'}], {'nlrt': '3.128.0.0/9'}]}]}|1713556673.2077806
3.128.0.0/9|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'local-preference': 100}, 'announce': {'ipv4 unicast': {'179.0.5.2': [{'nlrt': '3.0.0.0/9'}], {'nlrt': '3.128.0.0/9'}]}]}|1713556673.2077806
3.0.0.0/9|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/9'}], {'nlrt': '3.128.0.0/9'}]}]}|1713556673.2586133
3.128.0.0/9|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.154.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/9'}], {'nlrt': '3.128.0.0/9'}]}]}|1713556673.2586133
3.0.0.0/9|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/9'}], {'nlrt': '3.128.0.0/9'}]}]}|1713556673.2601078
3.128.0.0/9|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.153.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/9'}], {'nlrt': '3.128.0.0/9'}]}]}|1713556673.2601078
3.0.0.0/9|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.151.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/9'}], {'nlrt': '3.128.0.0/9'}]}]}|1713556673.2611601
3.128.0.0/9|3|13 [exabgp|1|A|{'attribute': {'origin': 'igp', 'as-path': {'0': {'element': 'as-sequence', 'value': [3]}}, 'med': 0, 'local-preference': 100, 'originator-id': '1.152.0.1', 'cluster-list': ['1.151.0.1']}, 'announce': {'ipv4 unicast': {'1.152.0.1': [{'nlrt': '3.0.0.0/9'}], {'nlrt': '3.128.0.0/9'}]}]}|1713556673.2611601
```

Figure 3.11: The output of the `tail` command

To ensure connectivity is restored, a ping can be executed at the hosts that previously were unreachable. This time the ping is successful since the destination host is part of the prefix that was mitigated.


```

root@WEST_host /> ping 3.151.0.1
PING 3.151.0.1 (3.151.0.1) 56(84) bytes of data.
64 bytes from 3.151.0.1: icmp_seq=1 ttl=61 time=3.54 ms
64 bytes from 3.151.0.1: icmp_seq=2 ttl=61 time=2.54 ms
64 bytes from 3.151.0.1: icmp_seq=3 ttl=61 time=2.56 ms
64 bytes from 3.151.0.1: icmp_seq=4 ttl=61 time=2.55 ms
64 bytes from 3.151.0.1: icmp_seq=5 ttl=61 time=2.55 ms
^C
--- 3.151.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 2.536/2.747/3.541/0.396 ms
root@WEST_host /> 

```

Figure 3.12: AS3 is unreachable from AS1

3.3.3 Using ARTEMIS detector to mitigate the hijack

The *active* mode of the ARTEMIS detector actively detects for hijacks of a specific AS prefix in a given interval. Lets observe how the tool detects and reacts to a hijack:

First up we need to launch ARTEMIS detector with the proper mode and its arguments. The command used for starting the detector in active mode is the following.

```

./artemis_detector active \
  --mitigation_script_path ../platform/utils/bgp_hijack/mitigate.sh \
  --updates ../platform/groups/exabgp_monitor/output \
  --input_type directory \
  --prefixes ../platform/groups/exabgp_monitor/as_prefixes.csv \
  --output mini-internet_hijacks.csv \
  --interval 1 \
  --asn 3

```

Using the above values we specify where the detector can find the mitigation script (the same script manually executed previously), the BGP update logs, the AS prefixes file and the desired output file. The input type is also specified as a directory, the interval every 1 minute and the AS to watch for hijacks as AS 3.

In Figure 3.14 we can see the output of the command.

```

chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/artemis_go$ ./artemis_detector active \
--mitigation_script_path ../platform/utils/bgp_hijack/mitigate.sh \
--updates ../platform/groups/exabgp_monitor/output \
--input_type directory \
--prefixes ../platform/groups/exabgp_monitor/as_prefixes.csv \
--output mini-internet_hijacks.csv \
--interval 1 \
--asn 3

ARTEMIS DETECTOR

Generating Peer Graph...
Generating Patricia Tree...
Real-time hijack detection...
Processing file: 1_output.csv
0% | | (0/0, 0 it/hr) [0s:0s]
Processing file: 2_output.csv
0% | | (0/0, 0 it/hr) [0s:0s]
Processing file: 3_output.csv
0% | | (0/0, 0 it/hr) [0s:0s]
Processing file: 4_output.csv
0% | | (0/0, 0 it/hr) [0s:0s]
Sleeping for 1 minutes...

```

Figure 3.13: ARTEMIS detector in active mode

Lets now execute a hijack while the detector is still running in a separate terminal using the hijack script that was presented earlier.

```

chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/platform/utils/bgp_hijack$ ./hijack.sh -a 2 -p 3.0.0.0/8 -t 20
(AS2)Hijacking 3.0.0.0/8 for 20 minute(s) on 2_WESTrouter
(AS2)Hijacking 3.0.0.0/8 for 20 minute(s) on 2_SOUTHrouter
(AS2)Hijacking 3.0.0.0/8 for 20 minute(s) on 2_EASTrouter
(AS2)Hijacking 3.0.0.0/8 for 20 minute(s) on 2_NORTHrouter
(AS2)20 minutes Remaining

```

Figure 3.14: Executing the hijack script

By executing a ping measurement to an IP address of the prefix from a host in AS1 we can observe that there is no connectivity. That means that the hijacker AS has successfully hijacked the prefix of AS 3.

```

root@WEST_host /> ping 3.151.0.1
PING 3.151.0.1 (3.151.0.1) 56(84) bytes of data.
^C
--- 3.151.0.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3058ms
root@WEST_host /> 

```

Figure 3.15: AS3 is unreachable from AS1

Looking back in the ARTEMIS detector (Figure 3.16) after the time interval has passed we can see that a hijack was detected and the mitigation script was executed successfully. The script's output is also printed in the terminal.

```

chris@ChrisPC-Ubuntu:~/Classes/mini_internet_extention/artemis_go$ ./artemis_detector active \
--mitigation_script_path ../platform/utills/bgp_hijack/mitigate.sh \
--updates ../platform/groups/exabgp_monitor/output \
--input_type directory \
--prefixes ../platform/groups/exabgp_monitor/as_prefixes.csv \
--output mini-internet_hijacks.csv \
--interval 1 \
--asn 3

ARTEMIS DETECTOR

Generating Peer Graph...
Generating Patricia Tree...
Real-time hijack detection...
Processing file: 1_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 2_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 3_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 4_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Sleeping for 1 minutes...
Processing file: 1_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 2_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 3_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 4_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Sleeping for 1 minutes...
Detected hijack for prefix: 3.0.0.0/8
Mitigating the hijack...
Mitigation script output:
(AS3)Advertising prefixes 3.0.0.0/9 and 3.128.0.0/9 from router
(AS3)Advertising prefixes 3.0.0.0/9 and 3.128.0.0/9 from router
(AS3)Advertising prefixes 3.0.0.0/9 and 3.128.0.0/9 from router
(AS3)Advertising prefixes 3.0.0.0/9 and 3.128.0.0/9 from router
(AS3)Mitigating indefinitely
End of mitigation script output.
Processing file: 1_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 2_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 3_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Processing file: 4_output.csv
0% | (0/0, 0 it/hr) [0s:0s]
Sleeping for 1 minutes...

```

Figure 3.16: ARTEMIS detected and mitigated the hijack

By executing the same ping measurement from the same host to the same IP address, we can confirm that the prefix mitigation was successful since the receiver responds to the ping request.

```

root@WEST_host /> ping 3.151.0.1
PING 3.151.0.1 (3.151.0.1) 56(84) bytes of data.
64 bytes from 3.151.0.1: icmp_seq=1 ttl=61 time=3.48 ms
64 bytes from 3.151.0.1: icmp_seq=2 ttl=61 time=2.56 ms
64 bytes from 3.151.0.1: icmp_seq=3 ttl=61 time=2.51 ms
64 bytes from 3.151.0.1: icmp_seq=4 ttl=61 time=2.52 ms
64 bytes from 3.151.0.1: icmp_seq=5 ttl=61 time=2.54 ms
^C
--- 3.151.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 2.513/2.722/3.479/0.378 ms
root@WEST_host />

```

Figure 3.17: AS3 is reachable from AS1

To stop the detector when it is in active mode, the operator needs to terminate the task by using Ctrl + C. When the detector is terminated the output file is generated containing the hijacks at the point of the termination.

3.4 Use Case: BGP Hijacks Assignment

As part of the 4th assignment of the HY436 course (Software Defined Networks) the platform was used for the students to experience how BGP hijacks work as well as how a network administrator monitors and protects their AS.

A mini-internet platform was created, accessible to the students through ssh, containing the modifications that were presented above:

A total of 15 students participated in the assignment. Each student was assigned to an AS. Since we didn't know how many students would participate in the assignment, we created a topology containing 32 ASes. The AS topology can be seen in Figure 3.18

3.4.1 Assignment's Platform

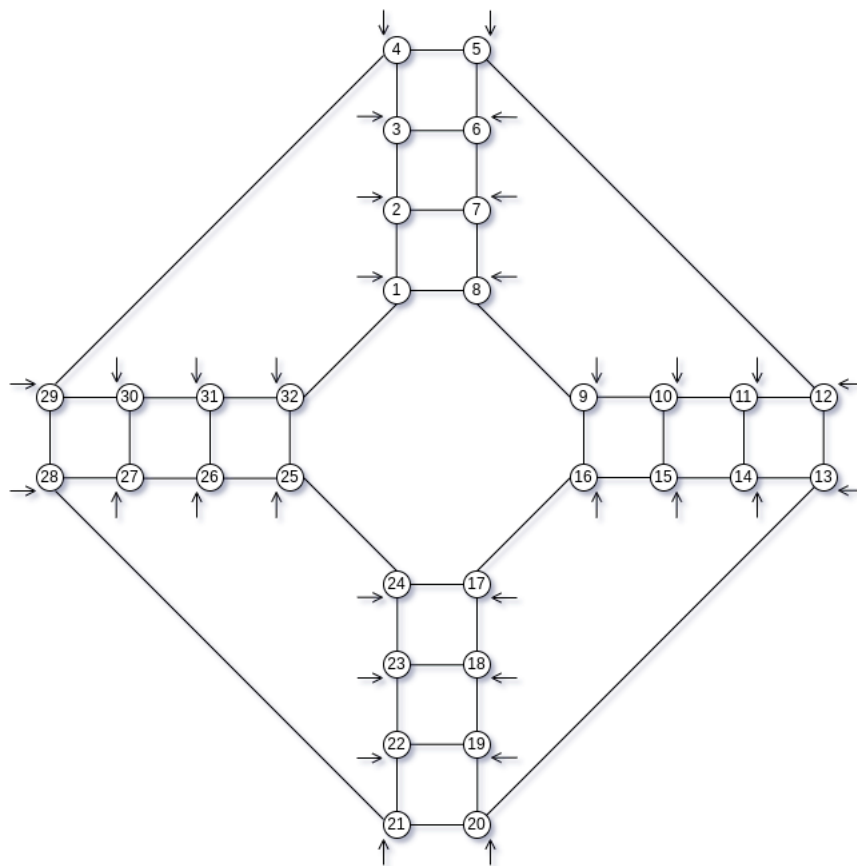


Figure 3.18: Mini-Internet topology

The platform's internal AS topology (Figure 3.19) follows the topology we saw earlier: 4 routers connected in a full-mesh topology and a monitor container connected to each one of them.

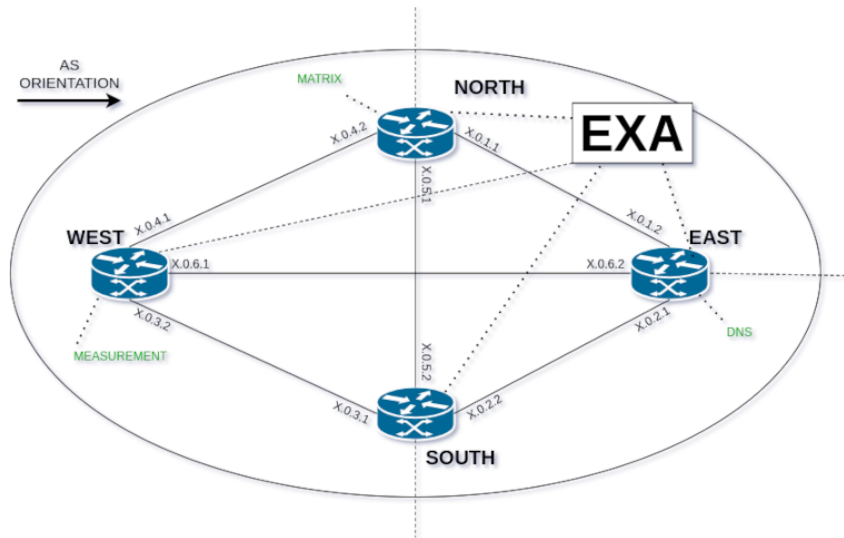


Figure 3.19: The internal topology of the assignments ASes

An extra connection to a container can also be seen on each router signifying the ability of each router:

- NORTH router: MATRIX Container *The matrix container is the source of the pinging service that creates the connectivity matrix we saw in section 2.1.*
- EAST router: DNS Container *The DNS container served as a Domain Name System server for the routers. It translating the links IP addresses to their respective names for more readable traceroute measurements.*
- SOUTH router: MATRIX Target *The matrix target container serves as a container for the matrix service to target the ping measurements.*
- WEST router: MEASUREMENT Container *The matrix container is connected to every WEST router in the topology and is equipped with a script that can execute traceroute measurements from a selected AS. The container is accessible to all students through ssh..*

3.4.2 Assignment's Execution

Teaching the students

Even though many students who participated in the project of the course HY335b (Computer Networks) were already familiar with the mini-internet platform, a lab session was planned. In this session the students were presented with the platform through a slideshows explaining how to use the platform and how ARETMIS Detector works.

The slideshows can be found in the following directory of my personal GitHub repository:

- Platform's Slideshow: [/HY436-Assignment4/session1/mini-internet_intro.pdf](#)
- ARETMIS Slideshow: [/HY436-Assignment4/session1/artemis_detector.pdf](#)

Following up to the first lab session, a second lab session explaining how BGP Hijacks work was also executed:

The BGP Hijacks slideshow can be found in the following directory:

HY436-Assignment4/session2/bgp_hijacks.pdf.

Assignment's Tasks

The students were grouped in groups of two, where each student would attack the group's prefix and when attacked by their group they should defend their prefix by mitigating it.

The assignment's tasks were split in two parts:

Attack Part In the attack part the students should configure their AS properly to execute a BGP Hijack to their group's prefix. They should report the inaccessibility of the grouped AS with a screenshot of the connectivity matrix after the attack has taken place. In Figure 3.20 we can see an example screenshot where ASes 22,24,26 and 28 are hijacked.

The red cells indicate inaccessibility in pinging the AS in the top row from the left column. Since for two ASes to communicate connectivity should be established both ways, the matrix is symmetrical on the top-left to bottom-right diagonal. When a line appears, it signifies that the AS's prefix is not reachable by multiple ASes. The cause of these unreachable areas is due to a BGP prefix hijack.

Defense Part In the defense part of the assignment the students were requested to monitor the connectivity matrix for incoming hijacks and when their group initiated a hijack they should use ARTEMIS detector (as shown in section 3.3.1).

After the students detect the hijack they should apply the correct configuration to mitigate the hijack of their group. When the mitigation was complete, the students were requested to attach a screenshot of the connectivity matrix proving that connectivity to their AS is restored for all other ASes of the topology.

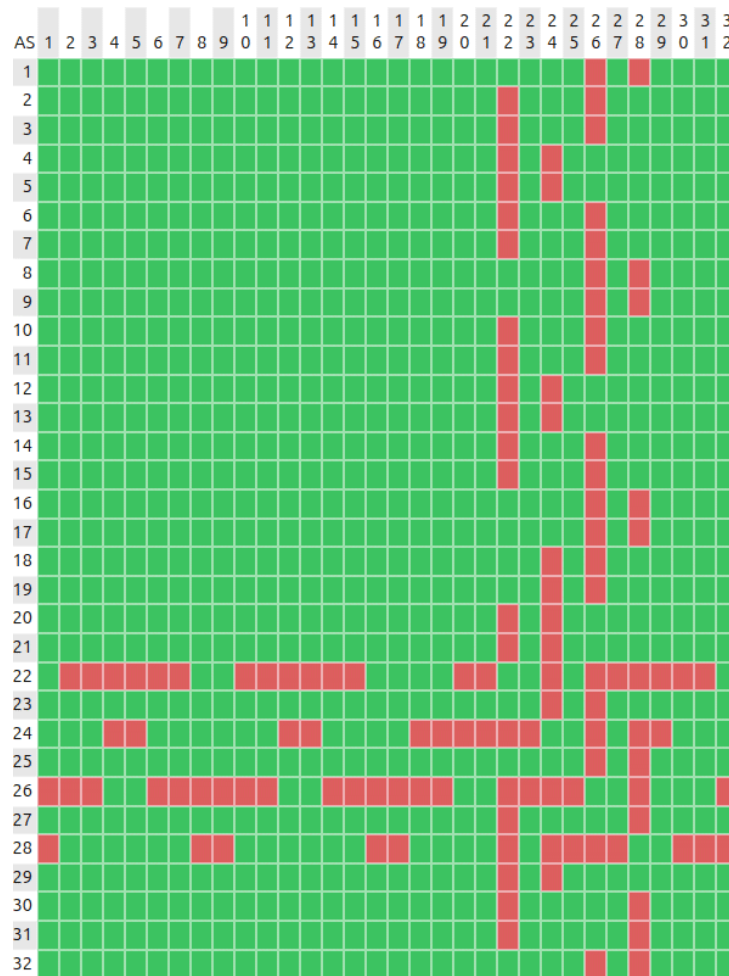


Figure 3.20: The connectivity matrix where ASes 22 24 26 28 are hijacked

3.4.3 Assignment's Results

From the 15 participating students, 14 students managed to come through with the assignment's tasks. All 14 students managed to hijack their target correctly as well as mitigate the hijack of their own prefix.

Based on students feedback, after the assignment was over, the assignment's tasks encouraged them learn how to execute and mitigate BGP hijacks since they had to step in the role of the attacker as well as the defender.

3.4.4 Acknowledgements

The assignment was executed and maintained with the help of Emmanouil Lakiotakis [11].

4 Summary

The platform is an astonishing tool when it comes to teaching students how to configure network devices while trying to give life to a network topology. Using the web-server and the tools provided by default, the debugging of the network configuration becomes easier as well as interactive.

The additions that were implemented as part of this thesis mostly improve the platform administrator's experience since with the addition of the admin web-server managing the course as well as the students attending becomes easier.

Since BGP Hijacks is such a serious form of attack in global networking, teaching future network engineers how to prevent such attacks becomes essential. Using the platform, the trainees gain hands on experience on how to detect and mitigate the attacks, configuring devices much alike devices in the real world.

BIBLIOGRAPHY

- [1] Ethz’s official mini-internet platform repository. <https://mini-inter.net>.
- [2] Vtysh documentation. <https://docs.frrouting.org/projects/dev-guide/en/latest/vtysh.html>.
- [3] Miguel Grinberg. Flask framework. <https://flask.palletsprojects.com/en/3.0.x/>, 2018.
- [4] Armin Ronacher. Jinja web template engine. <https://jinja.palletsprojects.com/en/3.1.x/>.
- [5] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. *IEEE/ACM Transactions on networking*, 9(6):681–692, 2001.
- [6] Michael Bayer. Sqlalchemy. <http://aosabook.org/en/sqlalchemy.html>, 2012.
- [7] Xenofontas dimitropoulos. <https://scholar.google.com/citations?user=pWyNwcMAAAAJ&hl=en>.
- [8] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. Artemis: Neutralizing bgp hijacking within a minute. *IEEE/ACM transactions on networking*, 26(6):2471–2486, 2018.
- [9] Exa-networks/exabgp: The bgp swiss army knife of networking. <https://github.com/Exa-Networks/exabgp>.
- [10] Go language official webpage. <https://go.dev/>.
- [11] Emmanouil Iakiotakis. <https://ieeexplore.ieee.org/author/37086057273>.

LIST OF ABBREVIATIONS

AS	Autonomous System
CLI	Command Line Interface
LAN	Local Area Network
VLAN	Virtual LAN
OVS	Open vSwitch
FRR	Free Range Routing
GO	GoLang
CSV	Comma-Separated Values
DNS	Domain Name System

Protocols

OSPF	Open Shortest Path First
BGP	Border Gateway Protocol
iBGP	internal Border Gateway Protocol
eBGP	external Border Gateway Protocol