AUTOENUM: Documentation for beta test April 2021 v1.0

Høgli, Sander
Lygre, Jarl Tengesdal
Małecki, Wojciech
Marjara, Avleen Singh

April 2021

Background

We are almost done with the system, and in that regard we arranged a meeting with NTNU SOC, represented by Christoffer Vargtass Hallstensen. The meeting took place on Teams March 26th 2021, and we gave Christoffer a live demo of the system (Autoenum). After the presentation Christoffer suggested that he wanted to test the system in NTNU SOC's own environment. This will benefit both parties: the group gets valuable feedback on how the system performs in a larger, real-world environment, and the employer get the opportunity to check if Autoenum fulfills their requirements. After the demo, Christoffer proposed some changes. The chagnes made since the demo, are listed in section 2.5. Before conducting the test, Christoffer wanted concise documentation on how to deploy and run Autoenum, along with an Ansible Playbook and the code. This document contains all information needed to deploy and run Autoenum.

1. *Installation*

1. Clone the Git repo `https://github.com/Monastyr/AutoEnumV3`
2. Create a "roles" directory in ansible directory(/etc/ansible)
3. Copy "autoenum" directory to the newly created "roles"
4. Create your own or use the example-playbook.yml from the repo and run it.
5. After the playbook finishes running, there should be docker conatiners running in the background with all the services needed to run the scan.
6. Run scanner/scanner.py using python3. Make sure target.txt is populated with IPs or CIDR blocks and target.txt is in the same directory as scanner.py

Code listing 1.1: Installation and setup example

```bash
#!/bin/bash

git clone https://github.com/Monastyr/AutoEnumV3
mkdir -p /etc/ansible/roles
cp -r AutoEnumV3/autoenum /etc/ansible/roles

# Sample Playbook:
ansible-playbook AutoEnumV3/example-playbook.yml

# Check if containers are running:
docker ps
# Should be 7 containers running

# Add targets to target.txt
cd AutoEnumV3/scanner
echo "192.168.1.0/24" > target.txt
python3 scanner.py
```

Code listing 1.2: Example Playbook

```yaml
 - name: Installing autoenum locally
  hosts: localhost
  connection: local
  become: true
  become_method: sudo

  roles:
    - autoenum
```

1.1 Potential issues during installation

Docker sometimes returns EOF error when building. See output from error 1.3. This occurs both when running the playbook and when running docker compose from bash. **Solution/workaround:** Open AutoEnumV3/autoenum/files/api/Dockerfile and make a change. Save the change. Open it up again and revert the changes and save. Try running playbook again.

Code listing 1.3: Partial output when running the included playbook. Same output when running docker compose from bash

```
fatal: [localhost]: FAILED! => {
    "changed": true,
    "cmd": "docker-compose␣-f␣/etc/ansible/roles/autoenum/files/docker-compose.yml␣
        up␣-d",
    "delta": "0:00:08.943178",
    "end": "2021-04-03␣12:02:21.767541",
    "invocation": {
        "module_args": {
            "_raw_params": "docker-compose␣-f␣/etc/ansible/roles/autoenum/files/
                docker-compose.yml␣up␣-d",
            "_uses_shell": true,
            "argv": null,
            "chdir": null,
            "creates": null,
            "executable": null,
            "removes": null,
            "stdin": null,
            "stdin_add_newline": true,
            "strip_empty_ends": true,
            "warn": true
        }
    },
    "msg": "non-zero␣return␣code",
    "rc": 1,
    "start": "2021-04-03␣12:02:12.824363",
    "stderr": "Building␣autoenum-api\nService␣'autoenum-api'␣failed␣to␣build:␣error
        ␣pulling␣image␣configuration:␣Get␣https://registry-1.docker.io/v2/library/
        python/blobs/sha256:
        bbec2a56954d351fcff1067348762a52d09c566709d0e04518cbed419aab979b:␣EOF",
    "stderr_lines": [
        "Building␣autoenum-api",
        "Service␣'autoenum-api'␣failed␣to␣build:␣error␣pulling␣image␣configuration:
            ␣Get␣https://registry-1.docker.io/v2/library/python/blobs/sha256:
            bbec2a56954d351fcff1067348762a52d09c566709d0e04518cbed419aab979b:␣EOF"
    ],
    "stdout": "Step␣1/7␣:␣FROM␣python:3.8-slim-buster\n3.8-slim-buster:␣Pulling␣
        from␣library/python",
    "stdout_lines": [
        "Step␣1/7␣:␣FROM␣python:3.8-slim-buster",
        "3.8-slim-buster:␣Pulling␣from␣library/python"
    ]
}
```

2. *Documentation*

2.1 Docker

Autoenum consists of several components. Some of the components are running locally on the host whilst the majority of the components have been Dockerized to support microservice architecture. All containers with the Autoenum prefix in figure 2.1 are made by the group. The containers with the CVE prefix are not made or modified by the group. They are part of the CVE-Search-Docker project on GitHub, and the project was recommended to us by Christoffer. The GitHub repository for CVE-Search-Docker can be found here: `https://github.com/cve-search/CVE-Search-Docker`.

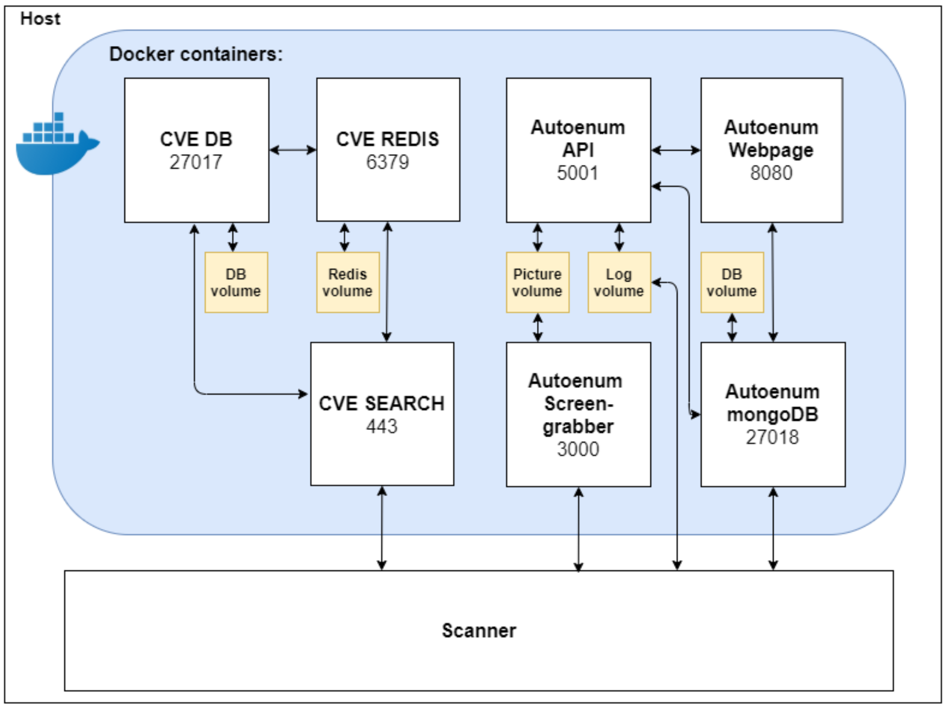


Figure 2.1: Architecture detailed

2.2 Scanner

The scanner is written in Python, and is heavily based on Nmap. The scanner runs directly on the host.

Usage

cd to directory where **scanner.py** is located. Make sure **cve_lookup.py** and **target.txt** are in the same directory as **scanner.py**. Make sure **target.txt** is populated with individual IPv4 addresses or CIDR blocks, each separated by a new line.

Run the scanner with:

```
python3 scanner.py [options]
```

Options for scanner:

- **-v / –verbose** Enables some output to screen. Tells the user which function it's currently running.
- **-t / –test** Enables test mode. Doesn't perform any scans. Instead reads data from 'udp.json' and 'tcp.json', and tries screengrab, CVE lookup and writes to database. This option is used when the user wants to check if everything works, but doesn't want to wait for a full scan to complete. Although the testing environment is different to our we have included 'udp.json' and 'tcp.json'.
- **-w / –write** Creates new 'udp.json' and 'tcp.json'. **DO NOT** use in conjunction with with **-t/–test**

The scanner also outputs a log, which by default can be found at:

```
/var/lib/docker/volumes/files_log-volume/_data/Scanner.log
```

The log can also be accessed through the API endpoint '**/log**'

Known issues

1. Doesn't remove **\n** from some Nmap outputs such as SSL certificates.
2. Generates errors when trying to screengrab ports that don't run a web server.

2.3 API

The API is written in Python/Flask. The API is the preferred way of interacting with the system. It returns all data from the database on the requested host object. The API is available on port **5001**.

Endpoints

The API has 8 different endpoints. All API-responses are in JSON, unless specified otherwise :

1. **/** Returns message to show that connection to API is working and a list of which endpoints are available
2. **/uuid/<uuid>** Returns the object with the given UUID
3. **/<cpe>** Returns all host objects with the given CPE. Works with cpe:/a, cpe:/h and cpe:/o.
4. **/ip/<ip address>** Returns all host objects with given IP address
5. **/mac/<mac address>** Returns all host object with the given MAC address
6. **/date/<yyyymmdd>** Returns all host objects scanned on given date
7. **/all** Returns the whole database
8. **/log** Returns scanner log for debug purposes **(Not JSON)**

Known issues

1. Issue 1 in 2.2 affects the JSON response. SSL certificates have a **\n**

2.4 Web interface

The web interface is made using Node JS, Express and EJS. The web interface provides a visual way of interacting with the data, and displays the data in a more human-readable format compared to the JSON responses from the API. The web interface should be treated as a proof of concept, where only some of the data is displayed. Screenshot samples can be found in appendix A.

Usage

Web interface runs in a Docker container on port **8080**. No authentication is needed to access the interface. We have implemented the following core functionalities:

- Front page where basic information such as IP, MAC and open ports about all host objects is displayed as cards
- Basic information display in list format
- A detailed view of each host object. Displays all results from OS-detection, ports, hostname, state, CVEs, CPEs, screengrabs and a button to get the host object from the database in JSON
- Host object search based on IP, MAC, CPE, UUID and date

Known issues

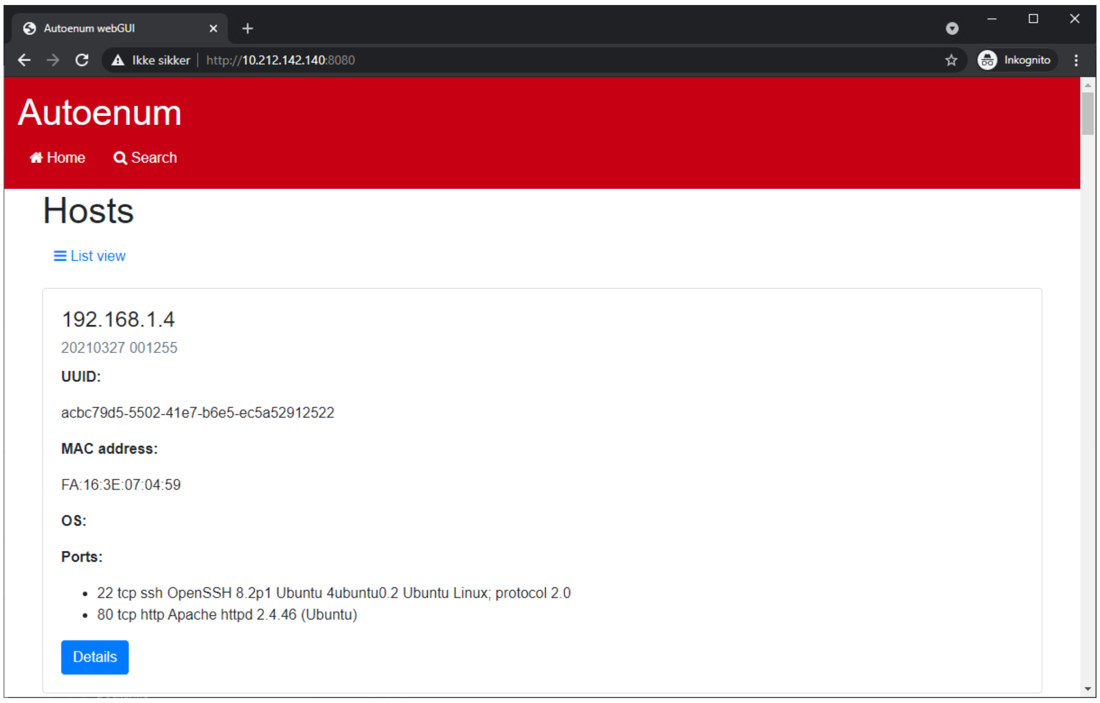
1. Not able to display image in /hosts/details or redirect to CVE database if accessing host through search or list view. Redirects to wrong URL, because the redirect is based on the requester's URL.

2.5 Changes

These are the changes implemented since the meeting with Christoffer March 26th:

- Added UUIDs to host objects. API and web interface updated to support the change.
- All references to mongoDB ObjectIds removed from API responses and web interface.
- Changed the search functionality. One search bar instead of four. User can search for host objects by UUID.
- Added new API endpoint for UUID.
- CPEs and CVEs are displayed on web interface
- CVEs listed on the web interface are clickable links which takes the user to the local instance of 'CIRCL CVE SEARCH'.
- The detailed view of the host

A. *Web interface screenshots*



Hosts

192.168.1.4
20210327 001255

UUID:

acbc79d5-5502-41e7-b6e5-ec5a52912522

MAC address:

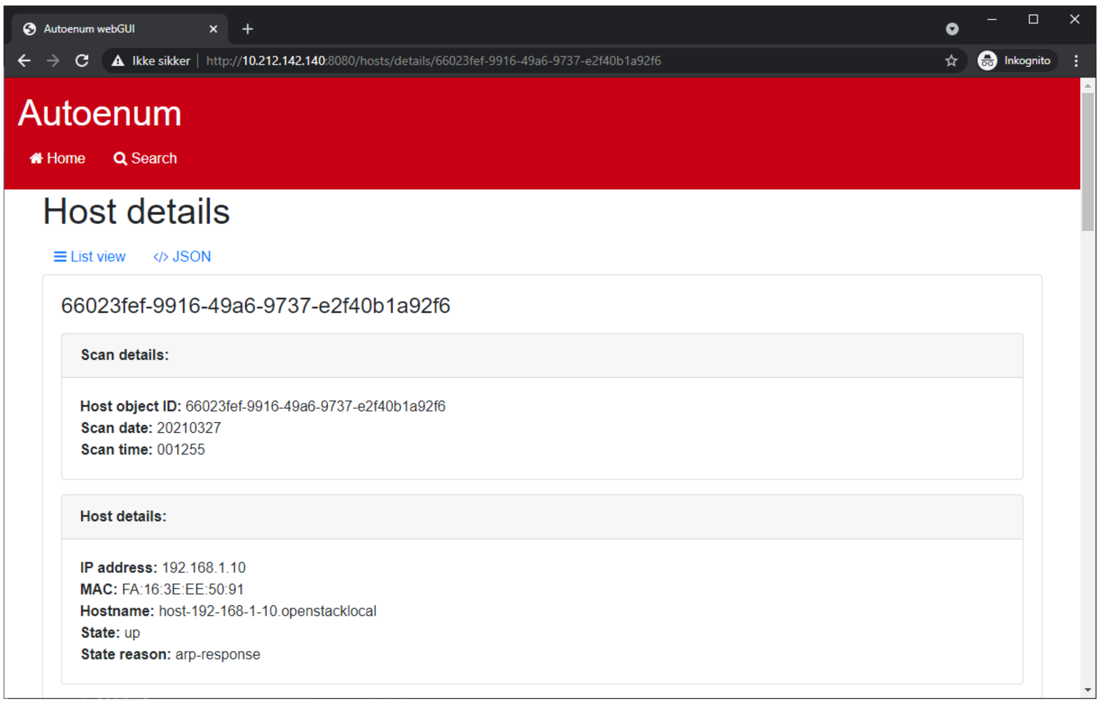
FA:16:3E:07:04:59

OS:

Ports:

- 22 tcp ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 Ubuntu Linux; protocol 2.0
- 80 tcp http Apache httpd 2.4.46 (Ubuntu)

Figure A.1: Home/index


Autoenum webGUI
Ikke sikker | http://10.212.142.140:8080/hosts/details/66023fef-9916-49a6-9737-e2f40b1a92f6
Inkognito
Autoenum
Home
Search
List view
JSON

Host details

66023fef-9916-49a6-9737-e2f40b1a92f6

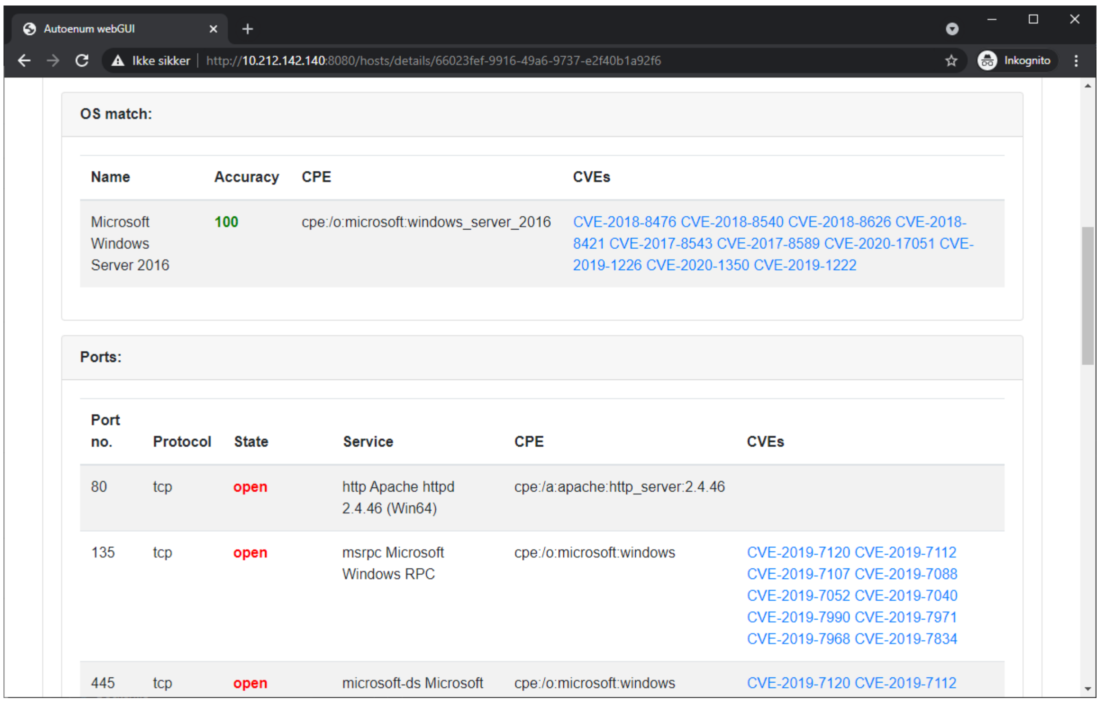
Scan details:

Host object ID: 66023fef-9916-49a6-9737-e2f40b1a92f6
Scan date: 20210327
Scan time: 001255

Host details:

IP address: 192.168.1.10
MAC: FA:16:3E:EE:50:91
Hostname: host-192-168-1-10.openstacklocal
State: up
State reason: arp-response

Figure A.2: Detailed view 1/3


Autoenum webGUI
Ikke sikker | http://10.212.142.140:8080/hosts/details/66023fef-9916-49a6-9737-e2f40b1a92f6
Inkognito

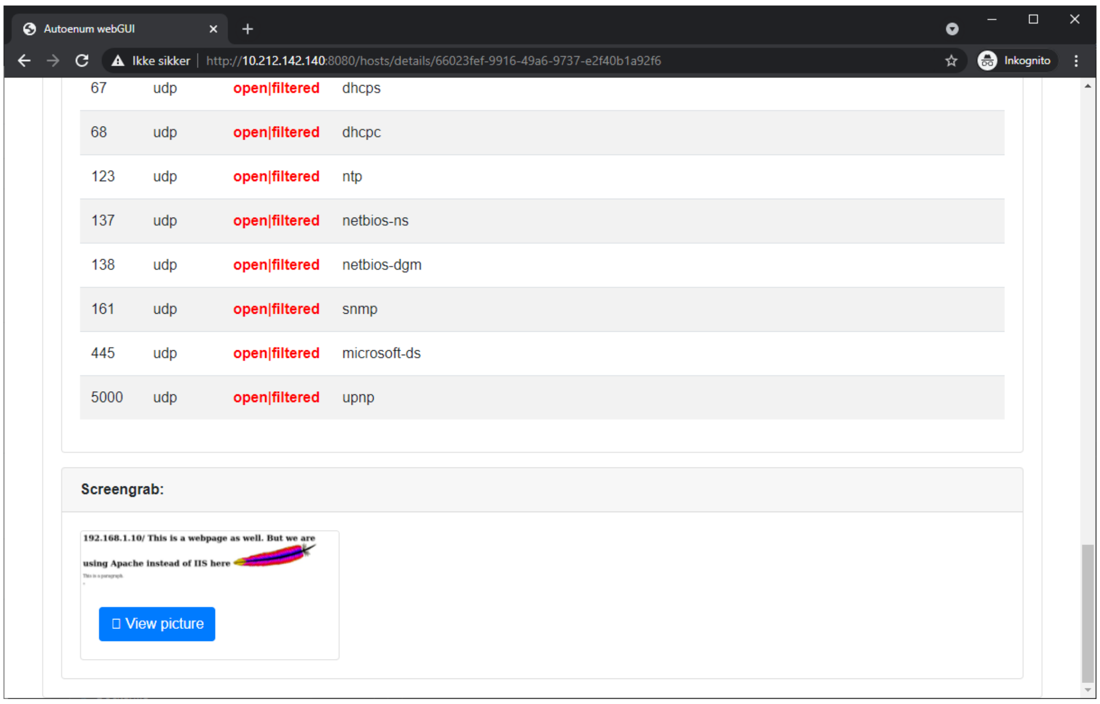
OS match:

Name	Accuracy	CPE	CVEs
Microsoft Windows Server 2016	100	cpe:/o:microsoft:windows_server_2016	CVE-2018-8476 CVE-2018-8540 CVE-2018-8626 CVE-2018-8421 CVE-2017-8543 CVE-2017-8589 CVE-2020-17051 CVE-2019-1226 CVE-2020-1350 CVE-2019-1222

Ports:

Port no.	Protocol	State	Service	CPE	CVEs
80	tcp	open	http Apache httpd 2.4.46 (Win64)	cpe:/a:apache:http_server:2.4.46	
135	tcp	open	msrpc Microsoft Windows RPC	cpe:/o:microsoft:windows	CVE-2019-7120 CVE-2019-7112 CVE-2019-7107 CVE-2019-7088 CVE-2019-7052 CVE-2019-7040 CVE-2019-7990 CVE-2019-7971 CVE-2019-7968 CVE-2019-7834
445	tcp	open	microsoft-ds Microsoft	cpe:/o:microsoft:windows	CVE-2019-7120 CVE-2019-7112

Figure A.3: Detailed view 2/3


Autoenum webGUI
Ikke sikker | http://10.212.142.140:8080/hosts/details/66023fef-9916-49a6-9737-e2f40b1a92f6
Inkognito

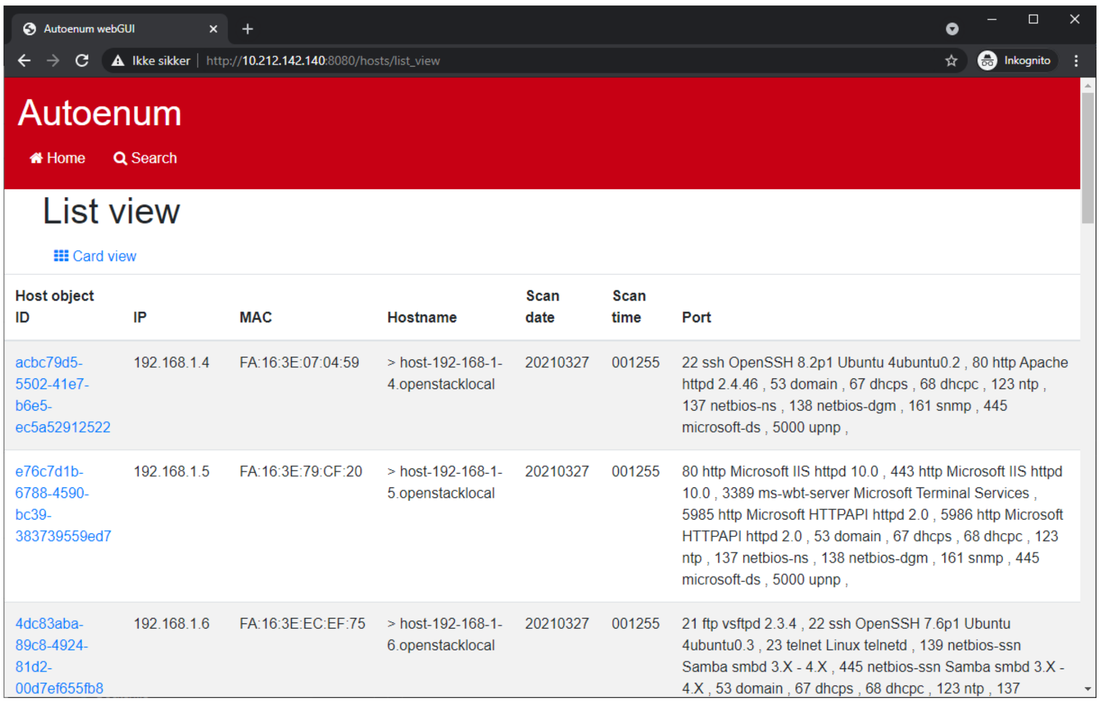
67	udp	open|filtered	dhcps
68	udp	open|filtered	dhcpc
123	udp	open|filtered	ntp
137	udp	open|filtered	netbios-ns
138	udp	open|filtered	netbios-dgm
161	udp	open|filtered	snmp
445	udp	open|filtered	microsoft-ds
5000	udp	open|filtered	upnp

Screengrab:

192.168.1.10/ This is a webpage as well. But we are using Apache instead of IIS here

This is a paragraph.

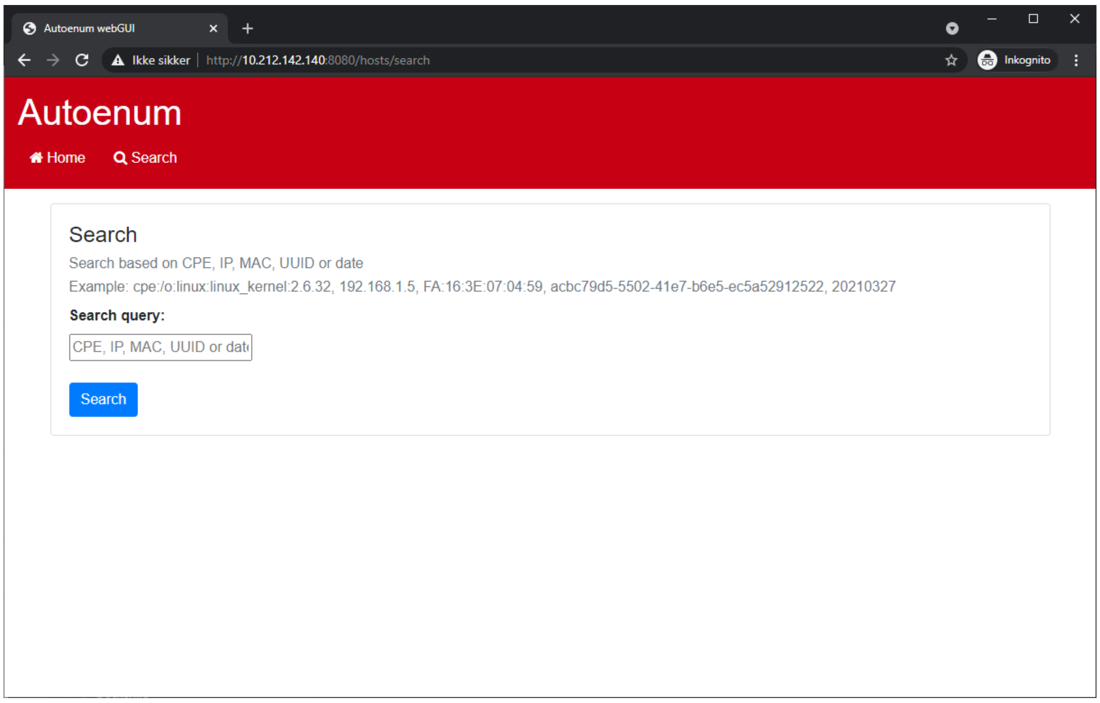
Figure A.4: Detailed view 3/3



List view

Host object ID	IP	MAC	Hostname	Scan date	Scan time	Port
acbc79d5-5502-41e7-b6e5-ec5a52912522	192.168.1.4	FA:16:3E:07:04:59	> host-192-168-1-4.openstacklocal	20210327	001255	22 ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 , 80 http Apache httpd 2.4.46 , 53 domain , 67 dhcps , 68 dhcpc , 123 ntp , 137 netbios-ns , 138 netbios-dgm , 161 snmp , 445 microsoft-ds , 5000 upnp ,
e76c7d1b-6788-4590-bc39-383739559ed7	192.168.1.5	FA:16:3E:79:CF:20	> host-192-168-1-5.openstacklocal	20210327	001255	80 http Microsoft IIS httpd 10.0 , 443 http Microsoft IIS httpd 10.0 , 3389 ms-wbt-server Microsoft Terminal Services , 5985 http Microsoft HTTPAPI httpd 2.0 , 5986 http Microsoft HTTPAPI httpd 2.0 , 53 domain , 67 dhcps , 68 dhcpc , 123 ntp , 137 netbios-ns , 138 netbios-dgm , 161 snmp , 445 microsoft-ds , 5000 upnp ,
4dc83aba-89c8-4924-81d2-00d7ef655fb8	192.168.1.6	FA:16:3E:EC:EF:75	> host-192-168-1-6.openstacklocal	20210327	001255	21 ftp vsftpd 2.3.4 , 22 ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 , 23 telnet Linux telnetd , 139 netbios-ssn Samba smbd 3.X - 4.X , 445 netbios-ssn Samba smbd 3.X - 4.X , 53 domain , 67 dhcps , 68 dhcpc , 123 ntp , 137

Figure A.5: List view


Autoenum webGUI
Ikke sikker | http://10.212.142.140:8080/hosts/search
Inkognito
Autoenum
Home
Search

Search

Search based on CPE, IP, MAC, UUID or date

Example: cpe:/o:linux:linux_kernel:2.6.32, 192.168.1.5, FA:16:3E:07:04:59, acbc79d5-5502-41e7-b6e5-ec5a52912522, 20210327

Search query:

CPE, IP, MAC, UUID or date
Search

Figure A.6: Search