Patchwatch - Analyzing the patching behavior of internet-connected devices

Julius Ruppert, Robin Müller, Yakub Cifci
University of Esslingen

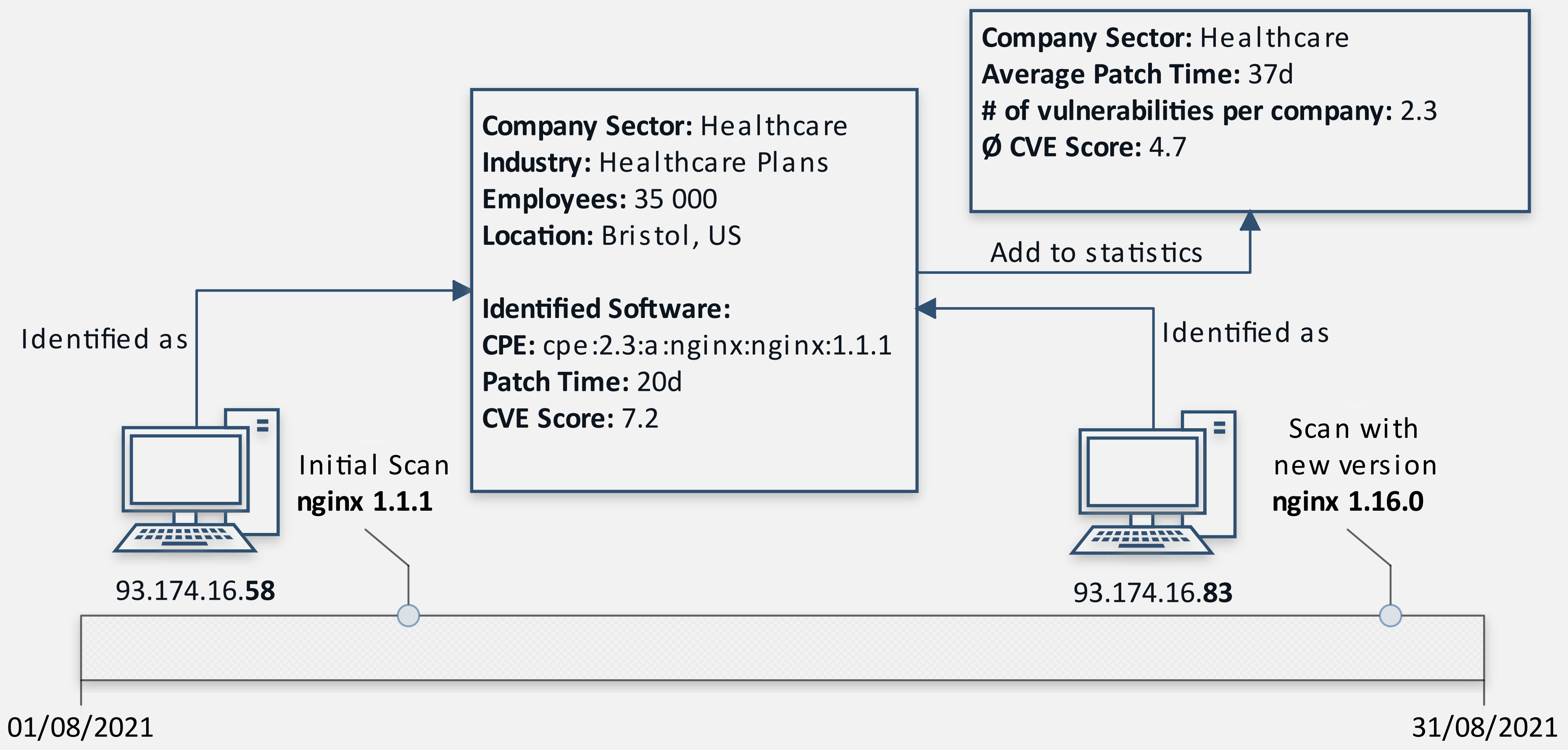
ESSLINGEN UNIVERSITY

1. Introduction

Many publicly available devices on the internet are not sufficiently updated to keep them free from vulnerabilities, which can be exploited by attackers. The goal of our project is to identify how different groups of users handle the patching process of their devices. Important questions are:

- How long does it take to implement necessary patches
- How serious are the vulnerabilities which are exposed
- Are there discernible patterns between different user groups

With this information we want to analyse current problems with the way patching is handled and give guidance to improve this behaviour in the future.



Company Sector: Healthcare
Average Patch Time: 37d
# of vulnerabilities per company: 2.3
Ø CVE Score: 4.7

Company Sector: Healthcare
Industry: Healthcare Plans
Employees: 35 000
Location: Bristol, US

Identified Software:
CPE: cpe:2.3:a:nginx:nginx:1.1.1
Patch Time: 20d
CVE Score: 7.2

Add to statistics

Identified as

Identified as

Initial Scan
nginx 1.1.1

Scan with
new version
nginx 1.16.0

93.174.16.**58**

93.174.16.**83**

01/08/2021

31/08/2021

2. Methods

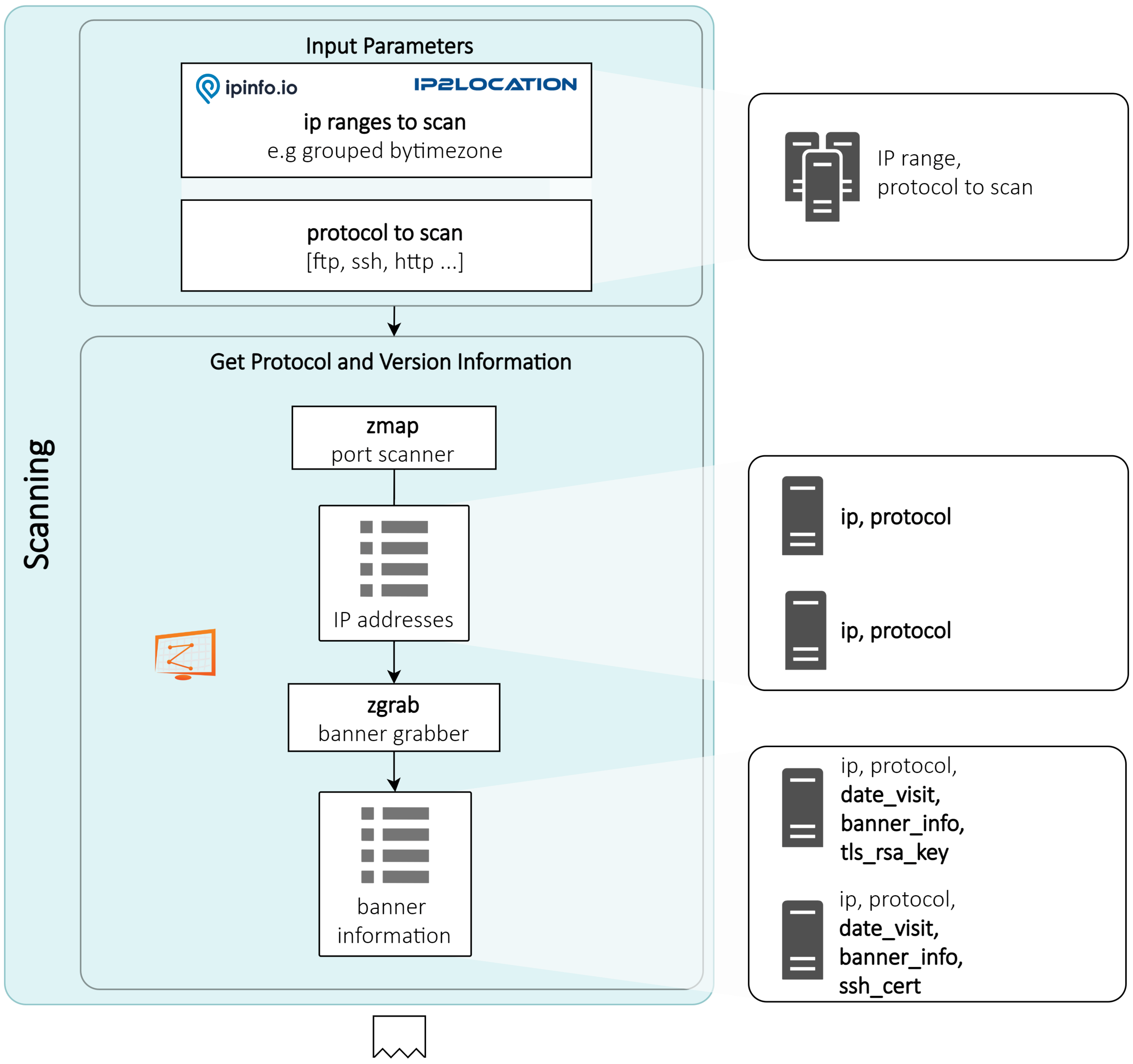
1. Scan internet devices and gather information
   - We scan the internet-connected devices either by using a service (e.g., Censys.io) or by using a port scanner
   - IP Information services can help us aggregate further information for a single device e.g., geo location, registry information
2. Uniquely identify the device and software version
   - Identify devices by fingerprinting certificate information
   - Using unique identifiers supplied by the protocols running on a device
3. Categorize and analyze the information
   - Categorize by branch/country/company size
   - Analyze update frequency/vulnerability percentage

3. Ethics and rules of engagement

In order not to act maliciously nor to disturb any activities, which would be highly illegal, we laid down a set of rules we carefully abide by:

- We set up a public website, on which we inform affected parties about our activities and offer an opt-out for everyone
- We will disclose our contact information and use technical methods (e.g., reverse DNS entry) to link the scanning activities back to us
- We inform about the IP range of our scanning server, so that the whole subnet of our provider won't be blocked
- We disclose the scanning activities to our provider and to our university
- Whilst scanning, we only use valid data. Using invalid data could potentially be interpreted as an intent to exploit foreign systems, which is illegal

4. Workflow



Scanning

Input Parameters

ipinfo.io IP2LOCATION

ip ranges to scan
e.g grouped bytimezone

protocol to scan
[ftp, ssh, http ...]

Get Protocol and Version Information

zmap
port scanner

IP addresses

zgrab
banner grabber

banner
information

IP range,
protocol to scan

ip, protocol

ip, protocol

ip, protocol,
date_visit,
banner_info,
tls_rsa_key

ip, protocol,
date_visit,
banner_info,
ssh_cert

Input parameters

In this initial step we use a subset of all IPv4 addresses to:

- scan the address range of one company only
- scan hosts which are in a specific timezone only (mitigate the risk of host obtaining a new ip address during scan)

We get this information of the databases of ip2location, ipinfo.io and by stock exchange platforms.
One scan covers one protocol for multiple addreses. Each step of one scan adds new data (see bold information in next steps).

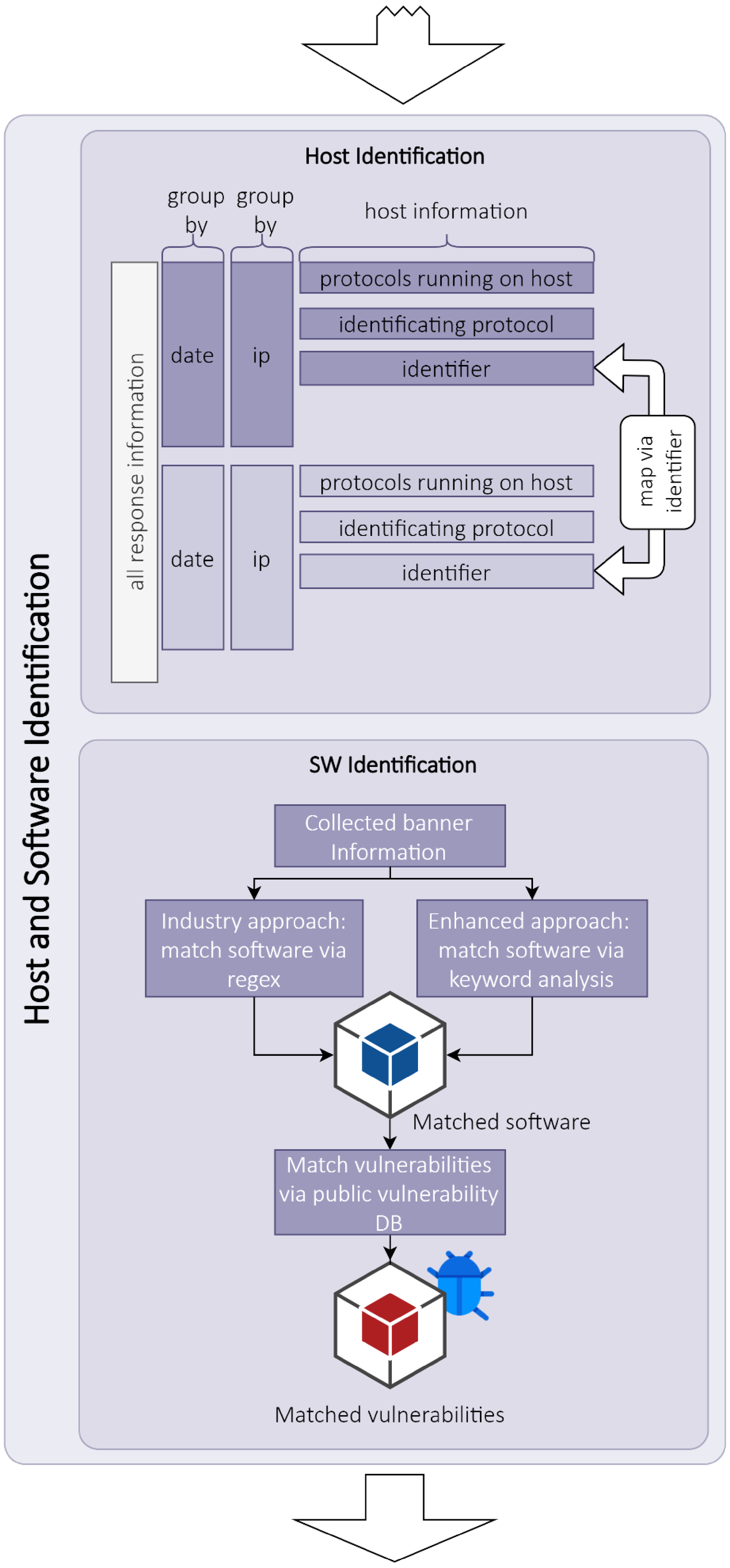
Port scanning with zmap

This stage determines which host is listening on which port by doing TCP and UDP handshakes with zmap. One challenge is to maximize the scan speed as much as possible while reducing the data loss e.g lost packets.

Banner grabbing with zgrab

With zgrab, we get protocol information of the port scanned hosts. This information is used to identify a host uniquely (e.g. with a public key of TLS) and its running software version of the service. Some protocols are good for identification while others provide precise software version information. Both are valuable four our project.

Project Patchwatch – Workflow and an outlook into the future



Host and Software Identification

Host Identification

all response information

group by

group by

host information

date

ip

protocols running on host

identificating protocol

identifier

date

ip

protocols running on host

identificating protocol

identifier

map via identifier

SW Identification

Collected banner Information

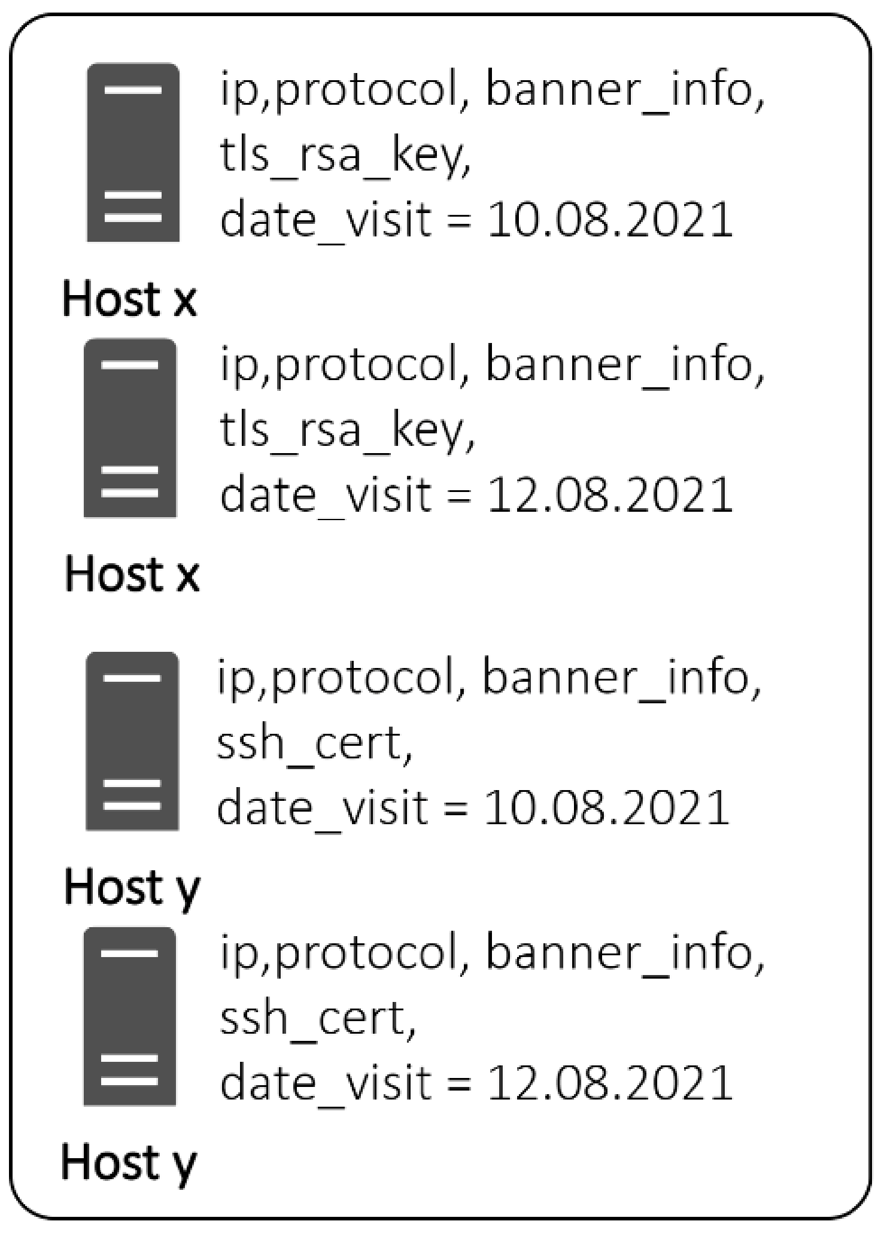
Industry approach: match software via regex

Enhanced approach: match software via keyword analysis

Matched software

Match vulnerabilities via public vulnerability DB

Matched vulnerabilities



ip,protocol, banner_info, tls_rsa_key, date_visit = 10.08.2021

Host x

ip,protocol, banner_info, tls_rsa_key, date_visit = 12.08.2021

Host x

ip,protocol, banner_info, ssh_cert, date_visit = 10.08.2021

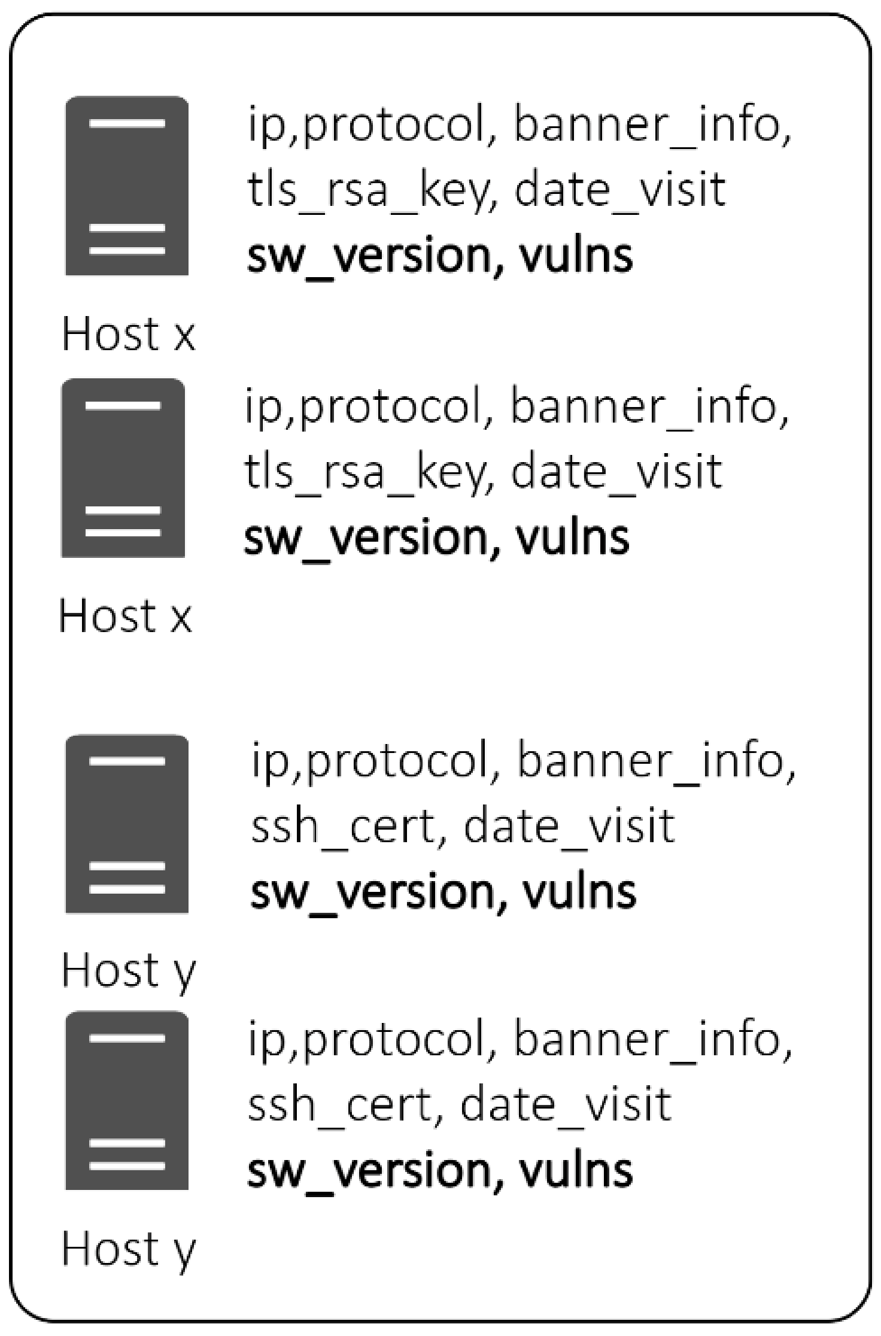
Host y

ip,protocol, banner_info, ssh_cert, date_visit = 12.08.2021

Host y

Get history of hosts

To check if a host has patched some software, we have to be able to identify the host uniquely over time. IP addresses may change over time, but combinations of protocol information or the public key of a certificate may be more static. So we try to map host by public keys.
With this strategy we get a history of identified hosts.



ip,protocol, banner_info, tls_rsa_key, date_visit **sw_version, vulns**

Host x

ip,protocol, banner_info, tls_rsa_key, date_visit **sw_version, vulns**

Host x

ip,protocol, banner_info, ssh_cert, date_visit **sw_version, vulns**

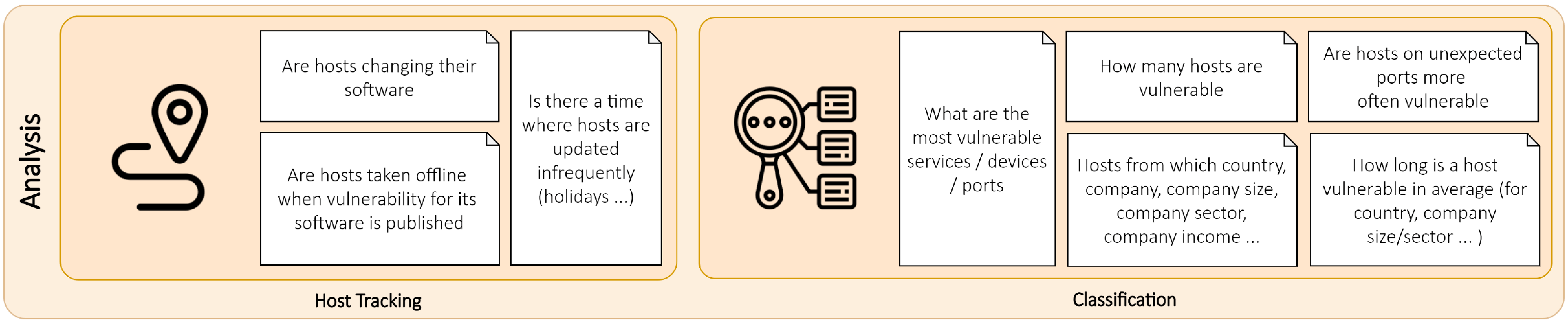
Host y

ip,protocol, banner_info, ssh_cert, date_visit **sw_version, vulns**

Host y

Identifying software

To determine possible vulnerabilities we need to identify the software running on the hosts we scanned. We are currently using a pattern matching approach that is already employed by internet scanning services like Censys. In parallel we are working on an improved technique based upon keyword analysis which can identify more software with less manual work. To identify possible vulnerabilities for the found software we use public vulnerability databases like the National Vulnerability Database (NVD) and others.



Analysis

Are hosts changing their software

Are hosts taken offline when vulnerability for its software is published

Is there a time where hosts are updated infrequently (holidays ...)

Host Tracking

What are the most vulnerable services / devices / ports

How many hosts are vulnerable

Are hosts on unexpected ports more often vulnerable

Hosts from which country, company, company size, company sector, company income ...

How long is a host vulnerable in average (for country, company size/sector ...)

Classification

5. Results

This far we were able to:

- Identify suitable scanning tools and methods for an internet wide port scanning operation
- Design a workflow that enables us to derive the desired information about the patching behavior of scanned hosts
- Implement a first version of this workflow which we can now improve upon
- Collect some first scan data which we can use to test our workflow with
- Create an enhanced approach to identifying software on a host which we can use to further improve the analysis

6. Outlook

For our future work, we plan to expand our current capabilities. This includes renting an external server from a provider which allows port scanning activities. With an external “operation base” we have more performance and speed to conduct our internet wide scans.

We also plan to improve the analysis capabilities we have developed this far. This includes improving the accuracy of the software identification. Going further we plan to aggregate and enrich the collected data to be able to identify the patching behavior of different user groups and company sectors.

Handling a lot of data which needs to be stored and processed in a timely manner is one of the challenges with this project. Therefore, some of our future work will also be improving the performance of our data pipeline.