Analyzing the Software Patch Discipline Across Different Industries and Countries

Robin Müller[1], Julius Ruppert[1], Katharina Will[1], Lukas Wüsteney[1], Tobias Heer [1]

Abstract:

In view of recent cyberattacks and new regulatory requirements, companies in different industries and countries are forced to implement additional IT security measures. Nevertheless, a large number of services with vulnerable or outdated software can be found on the Internet. In this work, we investigate whether industry-specific differences exist in the maintenance and use of outdated Internet-facing software. For this purpose, we combine results from Internet-wide port scans with product and version information as well as information of companies listed at stock markets in different countries. We show that different industries have more or less up-to-date software for different services like remote access tools, databases, webservers and file servers. With this approach, we discovered surprising amounts of outdated and even unsupported software in use across many industries and countries.

Keywords: Internet Scanning, Vulnerability Management, Patch Management

1 Introduction

The number and severity of cyberattacks increased dramatically within the last years. In many cases, attacks exploit new software vulnerabilities, or use already known software vulnerabilities in older software. Hence, companies should have an intrinsic interest in keeping their exposed and Internet-facing software up to date. Moreover, cybersecurity regulations in different industry sectors and in different countries have created an uneven playing field for companies. Some industry sectors in some countries are tightly regulated while only few regulations exist elsewhere. Examples of mandatory or voluntary regulations are NERC CIP in the US [No21], the NIS Directive in the EU [Eu21] and the PCI regulation in the financial sector [pc21].

Assuming that forced or voluntary cybersecurity standards raise the general cybersecurity maturity of a company, we evaluate whether or not differences in the application of patch management and the use of outdated or current software exist across industrial sectors and geographical areas. In order to achieve this goal, we conducted Internet-wide scans and correlated the results with product data as well as with company information and Internet network allocations. While the Internet-facing services of a company may not necessarily be part of the regulated IT infrastructure, we aim to evaluate whether a higher

[1] Hochschule Esslingen, Flandernstraße 101, 73732 Esslingen, Germany
{romuit04, juruit03, kawiit01, lukas.wuesteney, tobias.heer} @hs-esslingen.de

maturity and proficiency in cybersecurity might carry over to other less critical areas of an enterprise. An example of this would be critical infrastructure where the secure operation of an electrical plant may be tightly regulated while the Internet-facing customer portal may not be regulated in the same way. However, it still may be under the same IT security governance. In our evaluation, we focus on larger companies since these companies are more likely to fall under sectoral or local regulations regarding cybersecurity.

Within our research, we created a dataset containing a list of available services for stock listed companies. In addition to scanning, we collect available metadata (e.g., software-version, product name or sector of the company) as well. Our dataset reveals information about potential vulnerabilities, present on each host and a mapping between hosts and companies. To the best of our knowledge, we are the first presenting the approach to generate this structured dataset. With this dataset, we are analyzing our hypothesis and research questions. Specifically, we analyze (a) if different local regulations impact the occurrence of outdated software versions, and (b) if critical sectors, like finance and healthcare, are more up to date with their software.

Our contribution is threefold: (a) Develop and present a pipeline for Internet-wide version tracking [Mü22]. (b) Evaluate a large scan of stock market companies and their accessible services. (c) Visualize the software versions in use based on the location and the industrial sector of the companies for 14 different services.

This paper is structured as follows: First we summarize the concept of our scanning and evaluation approach in Section 2. In Section 3, we describe the path from our hypothesis towards a complete dataset for data analysis. Afterwards, we analyze our data to evaluate our hypothesis in Section 4. Next, we show the relation of our work to previous literature in Section 5. Finally, our conclusion in Section 6 is accompanied by our statement on ethical concerns of our work.

2 Analysis Pipeline Overview

In this section, we explain our approach on collecting and structuring data. We introduce all relevant data-sources and the path through our pipeline. Many services (e.g., SSH-Servers, HTTP-Servers, Mail-Servers, etc.) announce their identity via so-called banners upon connection establishment. Banners are service descriptions or information that can be visualized to users if they connect to a service. Often, these banners do not only contain the vendor of a product, but also contain product names and versions.

In order to acquire a list of active services on the Internet, we scan the IPv4 address space and collect the service banners. Service banners are retrieved during the connection phase with a service or protocol. They contain information which is either auto-generated or customized by the user (e.g., the administrator changes the name of the service). For each IP address, we only scan well-known ports for the protocols we intend to analyze. Our goal is to evaluate the landscape of software versions which are available on the Internet. For this

purpose, we create product identifiers in the form of *Common Platform Enumerations* (CPE) [Na11] for all collected banners. A CPE is used to identify a particular software or hardware product or operating system including the version number. The CPE is based on service information (e.g., the service announces the software and its version in an error response), or it can be derived from the banner (e.g., the service banner contains additional information about other software like the operating system). This leaves us with a rather unorganized set of IP addresses and identified products and versions.

In the next step, we identify whether a software version is up-to-date or not. To this end, we map the identified versions to vendor-specific lists in order to identify versions that are up-to-date, in extended support or have reached their end-of-life date (EOL). Without an adequate data source for such information, this matching involves manual compilation of version lists, support dates and end-of-life dates.

Making statements about the prevalence of older software in specific sectors requires a mapping from an IP address to a network to a company which in turn belongs to a certain sector or geographic region. In order to establish this mapping, we match and combine data from several sources. To map a scanned host to a network, we use IP address block information provided by ipinfo.io [ip21]. This enables us to group individual IP addresses so we can identify all services belonging to a company. Next, we match the names of the companies from ipinfo.io to sectors and countries by using publicly available stock market information from *Yahoo Finance*. In addition to identifying the industry sector, in which a company is active, we also retrieve geographic information about all companies based on their stock market registration. Using this legal information rather than IP-based geographic matching provides a more stable picture because load balancing services such as Content Delivery Networks (CDN) or geographic hosting can introduce a skew to IP geo-information based on the location of a scanner. So, depending on the scanner location, we will receive a response by different servers of an CDN. Yet, the fidelity of our IP information is still limited since companies may operate services at different branches in different countries. Hence, measurements that include geographic data must be considered with such limitations in mind.

3 Analysis Pipeline Implementation

In this section we discuss the pipeline, presented in Section 2, in more detail. First, we explain the process of creating the list of IP addresses to scan. Next, we describe the essential steps in the scanning process of our target hosts and the improvements we made to it. Finally, we conclude this section with our categorization and classification of the software discovered in the scanning process. The overview of the complete pipeline is depicted in Figure 1. Each step is explained and discussed in the following sections.

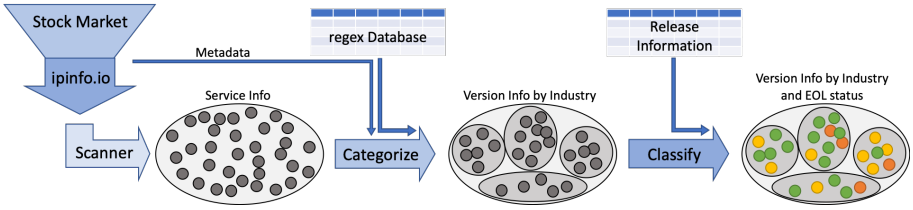


Fig. 1: Overview of Scanning, Categorization and Classification Pipeline

3.1 Gathering IP Addresses and Services

We identified stock market information as a useful source of company information. To be more efficient than scanning the complete Internet, we compiled a database from publicly available company data and map IP addresses to companies as discussed in Section 2. This list of IP ranges is used as basis for our IP and port scans. For each protocol, the port scanner *Zmap* can identify if the default port is open. In addition, *Zgrab2* can extract meta information, related to the service (e.g., banners).

Due to insufficient banner information or a low number of results for the protocol in general, we are not able to generate meaningful results for *BACnet*, *Niagara Fox*, *IPP*, *SMB* and *NTP*. Therefore, we stopped scanning for these protocols. The remaining scanned protocols are therefore: *HTTP*, *IMAP*, *POP3*, *SMTP*, *FTP*, *SSH*, *Telnet* and database services *MariaDB*, *Microsoft SQL Server*, *MongoDB*, *MySQL*, *Oracle Database* and *Redis*. Our evaluation is based on a single scan of the available IP addresses and protocols and therefore does not contain duplicate data for any host.

3.2 Scanning the Targets

In the context of scanning hosts, it is important to remember that companies might protect their infrastructure from malicious scanning and Denial-of-Service (DoS) attacks. Scanning from a single IP address leads to easily identifiable traffic. To avoid getting blocked based on the type and amount of traffic, there are two guidelines to follow: (a) As already presented by [Wa20], the scanning rate itself should be selected to be about 100,000 probes per second. (b) Additionally, IP addresses should not be scanned sequentially, but randomly. This spreads the scans across different address ranges and makes source-based identification over time more difficult. These steps help to achieve higher scan response rates, and also respect the service hosters by not posing a threat to the availability of their services. All of our scanning activities originated from a single source in the United States. Therefore, our data might not be complete as some services are not available from this location, or we might already be on blacklists, as our provider may be known for scanning activities.

Figure 2 visualizes the distribution of successful scans per sector. A successful scan means the port of the service is open and we receive additional information via a banner. We collect the information on industry sectors and companies from the data in the stock market

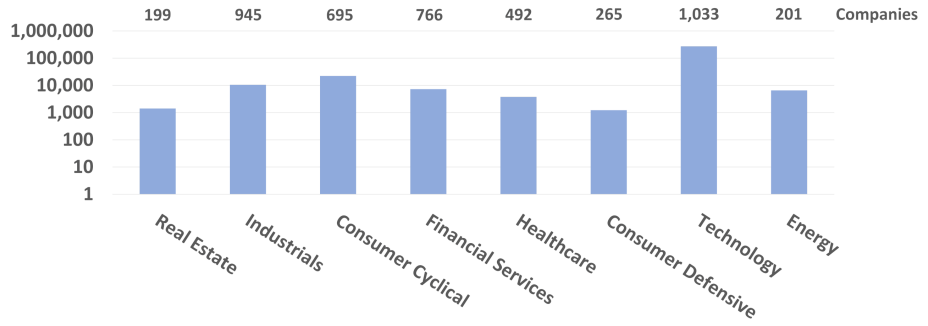


Fig. 2: Successful Scans per Sector

database. Figure 2 shows that the sector "Technology" clearly is the largest sector within our scans. However, the diagram shows a distribution of services across all sectors, which makes the comparisons between sectors possible. The sectors are as follows: Basic Materials, Communication Services, Consumer Cyclical, Consumer Defensive, Consumer Goods, Energy, Financial, Financial Services, Healthcare, Industrial Goods, Industrials, Real Estate, Services, Technology and Utilities. Within all of our diagrams we only present the sectors with the most CPEs.

3.3 Analyzing Banners

At this stage in our pipeline, we possess information about available hosts and services per company by scanning for service banners. In order to express statements other than "in sector ABC, the average company runs 100 services on 20 hosts", we need three more steps. Unfortunately, two of them require a lot of manual work (i.e., extracting data from the banners and identifying whether versions are still supported by their vendors).

As a first step, we extract information from all collected banners that are relevant to our research. This is a challenge since the banners vary in structure and content. While some banners might provide us with full information about the used product, software and even operating systems, others might only reveal the product itself. We use regular expressions (regex) to identify the parts of the banner that indicate product and version information. A regex defines a desired structure of a string and extracts substrings (e.g., a version in the format of aa.bb.cc with aa as major, bb as minor and cc as bugfix number). If a regex did not match a defined format, our tool continues with the next regex. The creation of the regex definitions requires manual analysis of the collected banners.

Depending on the banner structure, we can extract information about the product itself, its version, build dates and information about the operators or owners of the service. The combination of product name, vendor and version can be represented in a CPE. The quality of the gathered CPEs is highly dependent on our regular expressions. We visualize the distribution of the information we gain from the banners in Figure 3. Some protocols need a greater amount of scans to retrieve a meaningful amount of CPEs. For instance, from the

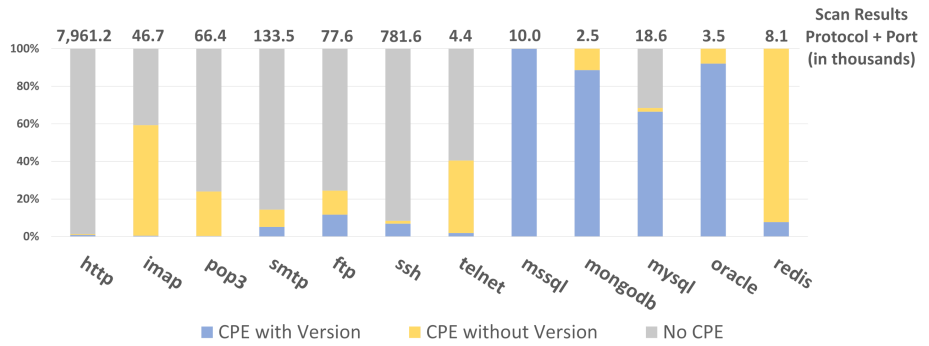


Fig. 3: Distribution of CPE information quality throughout our scans

HTTP protocol scans, only a very small percentage produce CPEs with product and version information, but the overall number of scans makes up for that fact. A CPE allows us to evaluate whether outdated and/or unsupported software is used.

In a second step, we compile a lookup table to decide which versions of the products that were revealed from the banners, are still maintained. This information is typically present on the vendor specific product page. After standard product support has ended, the vendor may still offer security fixes for an additional fee. This means we essentially have four cases per product: (1) still maintained, (2) maintained in certain cases, (3) end-of-life, and (4) no information. Case (4) represents all banners that revealed the product but not the version.

As a last step, we summarize all products of one category (e.g., *MySQL* and *MariaDB*, which are both database services, or *Apache HTTP Server* and *ngnix*, which are both webservers). In this way, we obtain an overview of the software landscape within a certain area on the Internet, which can be grouped by geographic location, industry sector or other metrics.

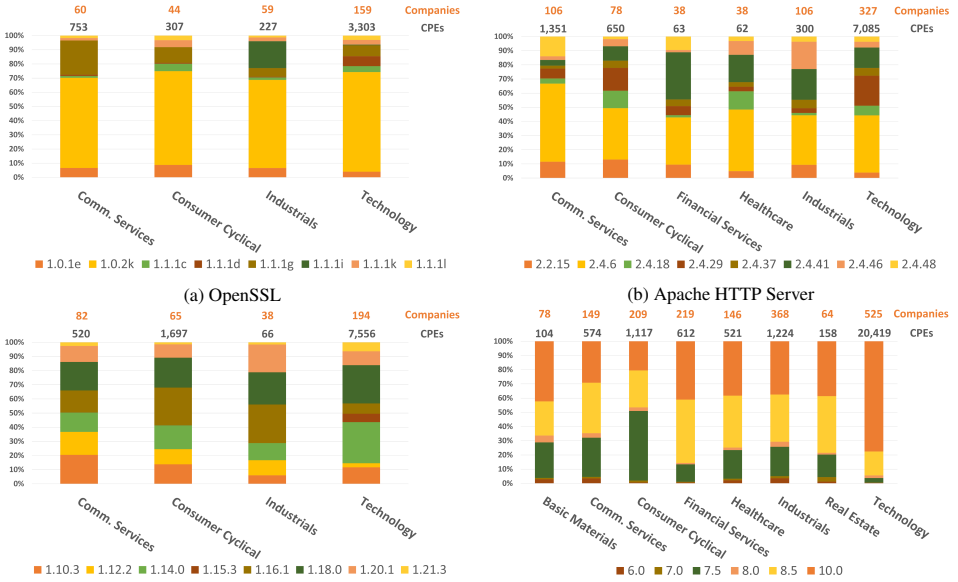
4 Evaluation

In this section we evaluate the collected dataset. We also present the findings about the software versions analyzed by region and sector.

4.1 Classification of Software Versions

In Figure 4, we compare services, commonly scanned, by the deployed software version. For a better overview, we present only sectors containing more than 30 companies and versions with more than 50 CPEs. We apply both filters for each diagram independently and limit the number of visualized sectors to eight. Hereby, each bar has a balanced distribution. In general, the figure shows that there are no obvious differences between the sectors since the distribution of versions is similar.

Unsupported *OpenSSL* versions such as 1.0.2 and below are still widely used across multiple sectors as we can see in Figure 4a. Especially version 1.0.2k is still very dominant in our



(a) OpenSSL (b) Apache HTTP Server

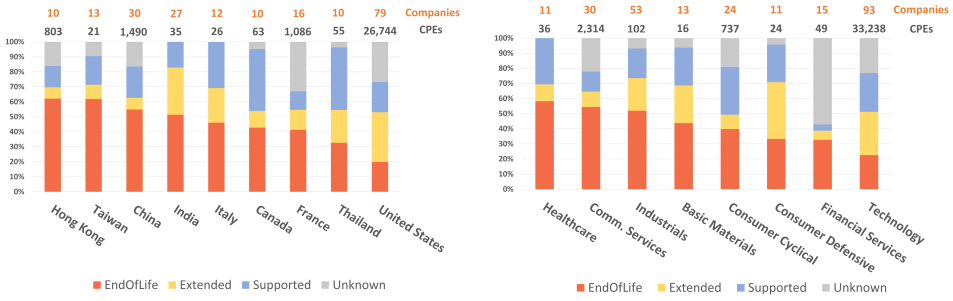
(c) Nginx (d) Microsoft IIS

Fig. 4: Distribution of software versions per sector

scans. However, it must be noted that the *OpenSSL Software Foundation* does offer extended support for this version. As shown in Figure 4b there are still *Apache HTTP Server* with version 2.2.x online. This version has reached its end-of-life in 2018 and was found about 550 times spread over 115 companies. The analysis of *nginx* webservers in Figure 4c shows that the majority of companies run software which is no longer supported (version 1.18 and below). All odd nginx versions represent the mainline branch, which receives updates most frequently. Figure 4c shows that the technology sector uses a mainline branch comparatively more often. For the other three sectors this is not the case. For *IIS*, the versions 7.5 and below are no longer maintained with security updates since January 2020. However, it is possible to get extended support for three years. *IIS* 6.0 is already end-of-life since 2015. The technology sector seems to keep this software relatively up to date, with versions 10 and 8.5 making up the bulk of Figure 4d.

4.2 Insights into Database Services

We compiled all our information related to databases into an aggregated view grouped by country and industry sector. For this evaluation, we excluded all sectors or countries that contained less than 10 companies to produce a more meaningful result. Our view on a per country basis (see Figure 5a) reveals that many old databases are still in use. This could be due to the fact that upgrading to a newer version is in many cases accompanied by a lot of work for migration. We might take the view that countries with rigid regulation bodies, like Thailand with a Cybersecurity Strategy [Of17] and the United States, are more



(a) Software support aggregated by country

(b) Software support aggregated by business sector

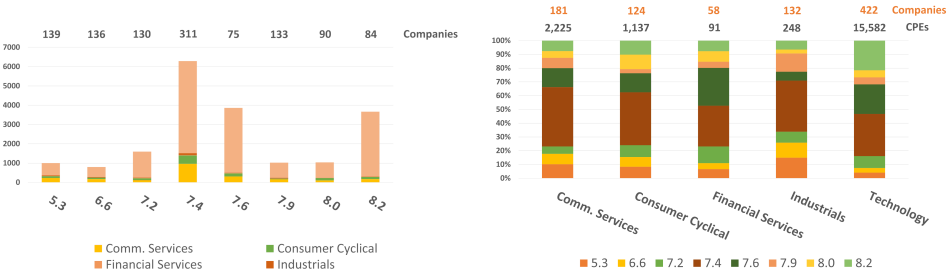
Fig. 5: The software support status of scanned database services (*MySQL*, *MariaDB*, *Oracle Database*, *Microsoft SQL Server* and *Redis*). Presented in a stacked diagram and visualizing the relation between identified end-of-life, possibly extended support, supported and unknown software versions.

likely to keep their software versions up to date. In the US, the FCC and NIST regulate on a federal level, while states have additional requirements like [Ca18] in California. For Thailand, personal data storages (often implemented with databases) security measures must be updated every three months [Of17]. Thailand also has a multitude of regulation bodies for cybersecurity, like the Organization of Critical Information Infrastructure (CIIO) and National Cybersecurity Committee (NCC). The data suggests that these enforcement measures may contribute to a better sense of security and therefore to a greater discipline when it comes to keeping versions up to date. Our literature research does not reveal similar strict regulations for the other countries, nevertheless such regulations may exist.

Figure 5b presents the view per sector. The technology sector seems to be faster when it comes to updating its services. This could be due to its proximity to the subject matter. In contrast, sectors with a completely different core business like Healthcare and Industrials operate a lot of older database versions. For some sectors, like financial services, our data contains too much noise in the form of unknown software configurations. This means, that we cannot make meaningful assumptions. Another consideration that needs to be made is that by focusing on the stock market data we only capture a relatively small number of the companies in a country or sector. Our assumptions can only be transferred to this limited amount of companies.

4.3 Insights into OpenSSH

We have taken a closer look at SSH, as it is a frequently used and widespread protocol. SSH is a security critical protocol, because it sets up remote shells with up to root privileges. In our scans, OpenSSH is the most common SSH server we discovered. Moreover, the banner analysis performed well for this protocol. We were able to assign 22,671 CPEs with version information and identify a total of 51 different versions. Figure 6a represents a filtered view, as all versions with less than 200 occurrences are filtered out. We see that the versions 7.4, 7.6 and 8.2 are widely used. We evaluated that here is a connection between the



(a) Distribution of matched OpenSSH CPEs per version. Only visualizing all versions with more than 800 hits

(b) Distribution of matched OpenSSH CPEs per company sector

Fig. 6: Deep Dive in OpenSSH

high numbers of these versions and LTS versions of widely used Linux operating systems. In our dataset 99% of OpenSSH versions 8.2 and 7.6 are represented by Ubuntu. These versions ship with Ubuntu LTS 20.04 and 18.04. For version 7.4 we cannot make a clear determination of the corresponding operating system. A major update of the OpenSSH stack often requires a full upgrade of the operating system, as many dependencies exist. This might be a reason why many outdated versions are used as well. However, old versions are not necessarily vulnerable as Operating System manufacturers implement security fixes into their provided OpenSSH fork.

Analysis of version distribution across the sectors (see Figure 6b) shows no clear trends. The same applies to the version distribution by country and therefore was not visualized. All sectors have roughly equal numbers of old and new software versions of OpenSSH in use. However, Technology and Financial Services seem to run the least old OpenSSH versions.

4.4 Scanning Performance and Matching Products and Versions to CPEs

Since we are scanning at a rate of 100,000 probes per second the regex matching needs to be able to keep up with the inflow of scans. The system we designed continually collects new scans from our database and matches them via the regular expressions. We can process 263,000 scans per second with a single core of an *AMD Ryzen™ 9 5900X*. Since there are no dependencies between scans for the CPE assignment process, this work can be parallelized. This process can further be optimized since scans that do not result in any banner data can be discarded without any applied processing time.

In our scan, we generated 198,205 CPEs containing vendor and product information from which 102,874 CPEs contained version information.

5 Related Work

In this section, we present related work in the area of scanning on the Internet, vulnerability identification and CPE matching. Several authors performed Internet scans with goals that

differ from ours. Dahlmanns et al. scanned the entire IPv4 address space to analyze OPC UA appliances [Da20]. OPC UA is a protocol for secure industrial communication, hence it is applicable for one industry sector. In our work, we try to compare services in different industries and match versions and CPEs to identify dated software. Internet-wide scans were used by Durumeric to identify new vulnerabilities [Du17]. Therefore, software versions which might already be outdated, are not within his focus. We use outdated product versions to identify bad patching discipline instead. Na et al. suggest identifying CPEs and their vulnerabilities automatically based on grabbed banners [NKK18]. We decided to match CPEs manually with regex, as this is more reliable and stable, especially since we need to identify versions as well.

Morishita et al. used Internet-wide scanning to identify honeypots [Mo19]. The authors identified these honeypots in research networks, in commercial hosting and access networks. We did not analyze, if responses to our scan can be matched to honeypots. However, the numbers presented in [Mo19] do not distort our large scan results.

Another point of view was taken by Wan et al. by researching the impact of the scanner's geographical location on the scan results and how inaccurate Internet-wide scans are [Wa20]. Their results also show that there is no optimal location to scan from. The authors show the different factors that can further impact results such as blocked scans based on source addresses. Compared to the large number of results collected, we argue that the effect of these blocked scans is marginal.

6 Conclusion

Within our IP scan of 9,838 stock-listed companies we have received a total of 368,953 successful protocol replies of operational services on the Internet. Across all 198,205 successfully identified CPEs, we can show a wide range of outdated software versions for all industry sectors.

We could not identify sectors that are more up-to-date in general (i.e., we have seen an even distribution for SSH). However, for specific protocols (i.e., all database CPEs combined) a difference between sectors and regions is noticeable. Based on local regulations for critical infrastructure, we assume certain software is updated more regularly. For database software, we have also presented a detailed analysis on the supported or outdated software versions in use. We found an astonishing amount of old services still operated by stock exchange listed companies. The technology sector showed the best patch discipline and the most up-to-date versions. This result is surprising because other sectors (e.g., healthcare, and financial services) are more tightly regulated in many countries.

Some sectors are not well represented in our dataset. Therefore, we are not able to draw deeper conclusions about their state. To improve statements made based on industry sectors and Internet-wide scanning, further investigation is needed so less ambiguous information can be extracted.

Appendix: Ethical Considerations

Starting with broad scanning processes requires a few things to remember and guidelines to follow. One always needs to respect certain parties not being happy about scans and should exclude them once notified. We implemented a webserver on our scan server with information on this project and added the according abuse fields in our *whois* entry. With this information, every target can contact us, even with auto-generated emails, and can get excluded from future scans.

References

[Ca18] California Legislative Information: Bill Information. In: Assembly Bill No. 1906. 2018, URL: https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill%5C_id=20172018OAB1906, visited on: 11/11/2021.

[Da20] Dahlmanns, M.; Lohmöller, J.; Fink, I. B.; Pennekamp, J.; Wehrle, K.; Henze, M.: Easing the Conscience with OPC UA: An Internet-Wide Study on Insecure Deployments. In: IMC '20: Proceedings of the ACM Internet Measurement Conference. Pp. 101–110, 2020.

[Du17] Durumeric, Z.: Fast Internet-Wide Scanning: A New Security Perspective, PhD thesis, University of Michigan, 2017.

[Eu21] European Union Agency for Cycbersecurity: enisa. In: NIS Directive. 2021, URL: https://www.enisa.europa.eu/topics/nis-directive, visited on: 11/11/2021.

[ip21] ipinfo.io: IP Ranges API, 2021, URL: https://ipinfo.io/developers/ranges, visited on: 11/11/2021.

[Mo19] Morishita, S.; Hoizumi, T.; Ueno, W.; Tanabe, R.; Gañán, C.; van Eeten, M. J.; Yoshioka, K.; Matsumoto, T.: Detect Me If You... Oh Wait. An Internet-Wide View of Self-Revealing Honeypots. In: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). 2019.

[Mü22] Müller, R.; Ruppert, J.; Will, K.; Wüsteney, L.; Heer, T.: HSES-Patchwatch Project Documentation, 2022, URL: https://hs-esslingen-it-security.github.io/hses-patchwatch/, visited on: 01/19/2022.

[Na11] National Institute of Standards and Technology (NIST) - U.S. Department of Commerce: Common Platform Enumeration: Naming Specification Version 2.3, 2011, URL: https://www.govinfo.gov/content/pkg/GOVPUB-C13-c213837a04c3bcc778ebfd420c6a3f2a/pdf/GOVPUB-C13-c213837a04c3bcc778ebfd420c6a3f2a.pdf, visited on: 11/11/2021.

[NKK18] Na, S.; Kim, T.; Kim, H.: Service Identification of Internet-Connected Devices Based on Common Platform Enumeration. In: Journal of Information Processing Systems. Vol. 14, pp. 740–750, 2018.

[No21] North American Electric Reliability Corporation: NERC. In: CIP Standards. 2021, URL: https://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx, visited on: 11/11/2021.

[Of17] Office of the National Security Council: Thailand. In: National Cybersecurity Strategy 2017-2021. 2017, URL: http://www.nsc.go.th/wp-content/uploads/2018/08/strategyit60-64-1.pdf, visited on: 11/11/2021.

[pc21] pci Security Standards Council: pci. In: Document Library. 2021, URL: https://www.pcisecuritystandards.org/document_library, visited on: 11/11/2021.

[Wa20] Wan, G.; Izhikevich, L.; Adrian, D.; Yoshioka, K.; Holz, R.; Rossow, C.; Durumeric, Z.: On the Origin of Scanning: The Impact of Location on Internet-Wide Scans. In: IMC '20: Proceedings of the ACM Internet Measurement Conference. Pp. 662–679, 2020.