Identification and Comparison of Public Scanning Tools for Network Attack Frequency Analysis

Alexander Brakowski
TU Darmstadt
Email: genix2006@gmail.com

Manjiri Birajdar
TU Darmstadt
Email: manjiri_birajdar@outlook.com

***Abstract*—Scanning through the internet for all connected devices is an interesting and big task. Especially since certain devices, like Internet of Things (IoT), are more vulnerable and prone to attacks. Therefore, network scanning tools were developed which are capable of scanning the full IPv4 address space. The purpose of this paper is to identify and compare these scanning tools (IP Device search engine). Additionally, our study includes details about the information which can be exploited by them and how this information could theoretically be used. We have identified few tools that completely and automatically perform the internet-wide scanning, periodically and later publish the results on a website, which is accessible by anyone (Shodan, Censys, Thingful, PunkSPIDER, IVRE, Zoomeye). These tools are generally managed by research teams. Some other useful tools we found, can be used for vulnerability and other scanning activities, but they do not publish the results on any public website (Nessus, skipfish, Acunetix, Vega). The anticipated outcome of this paper is an overview of the most popular and useful, public and user-interaction based tools for internet-wide scanning.**

Keywords*—*Internet of Things, IoT, Scanning Tools, Vulnerability Scanners, Google for Hackers, Internet, Network Security.

I. Introduction

The Internet is limitless and, in fact so big that even having a very tiny surface-level overview about it, is a major task. There are currently several automatic scanning tools available, which are capable of scanning the whole IPv4 address space and extract as much information as possible out of these findings. They typically operate in a distributed and random manner and publish their results on a publicly accessible website.

Since, it is such a big feat to be able to write and operate such tools, it is very useful for other research teams to be able to freely use this data without having to operate such a scanning tools themselves. The problem is that this data can also be used for malicious purposes. Potential attackers gain free reconnaissance data of the whole Internet without ever needing to come in contact which any of their targets. Additionally, they can use the data to identify large amount of potential victims for certain vulnerabilities, that the attackers know about.

While researching, we have found a few tools that are able to completely and automatically scan the internet, interpret the findings and publish results on a publicly accessible website (Shodan, Censys, Thingful, PunkSPIDER, IVRE, Zoomeye, scans.io). Some other useful tools we found, can be used for vulnerability and other scanning activities, but they do not publish the results on any public website (Nessus, skipfish, Acunetix, Vega).

In the first part of this paper, we are going to discuss the available *Public Scanning Tools* and *Personal/User Interaction Based Scanning Tools* including its description, general goals, in which context each automatic scanning tool is adopted, what kind of data/meta-data or other information are exploited by each scanning tool and does it implement a particular technique/algorithm to get these information. In the second part of the paper, we will discuss the pros and cons of each described scanning tools in the first part. We will end the paper by comparing the tools based on specific criteria such as mechanism of information gathering and detection and some more.

II. Public Scanning Tools

Public Scanning Tools are nothing but search engines which looks for the information. As per [1], search engines can be seen in different aspects such as :

- **General-purpose Search Engines:** Google, Bing or Yahoo. For example, Google search works based on query which has different input attributes such as text, image, audio, etc. It is mainly based on the content-type and indexing of a page.
- **Subject-specific Search Engines:** Subject-specific search engines does the internet-wide scanning for a specific defined subject area, such as hosted services, SSL/TLS vulnerabilities or concrete vulnerabilities in the Internet-enabled software, such as XSS or SQL injection [1]. In contrast to General-purpose Search Engines, subject-purpose Search Engines precisely processes the retrieved information about the system. It mainly focuses on vulnerabilities. For example, some tools take URL as and input attribute to find out internet-connected devices for vulnerability. Shodan uses query attribute as "Server: SQ-WEBCAM" and URL looks like https://www.shodan.io/search?query=Server%3A+SQ-WEBCAM

Here, we are going to discuss the tools based on Subject-specific Search Engines such as Shodan.

A. *Shodan*

1) Description: Shodan is a search engine for the Internet of Things (+ security, Buildings, the Web, the webcams, Refrigerators, Power Plants). Shodan allows users to find

devices that are publicly accessible on the Internet, and which may be vulnerable to hackers. Meaning, Internet-connected devices can be found by the search engine of Shodan. [2]

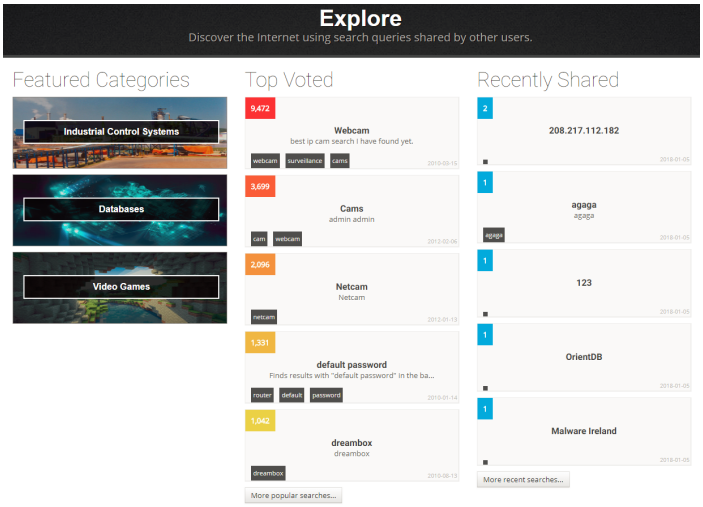

Fig. 1. Shodan Explore Results

2) Aim: Shodan can crawl the Internet 24/7 to provide the latest Internet intelligence. Following are some of the things for which Shodan is aiming for:

- Explore the Internet of Things : Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them [2].
- Monitor Network Security : Keep track of all the computers on your network that are directly accessible from the Internet. Shodan lets you understand your digital footprint [2].
- See the Big Picture : Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan! [2]
- Get a Competitive Advantage : Who is using your product? Where are they located? Use Shodan to perform empirical market intelligence [2].

Shodan could also find who buys Smart TVs, which countries are building the most wind farms and what companies are affected by Heartbleed [2].

Moreover, Shodan comes with Enterprise access too. Using Enterprise Shodan, one can download all the data crawled by Shodan, store it, analyze it and build products with it.

3) Adoption Context: As per the [3], Google Hacking Diggity Project is a research and development initiative dedicated to investigating Google Hacking, i.e. the latest techniques that leverage search engines, such as Google, Bing, and Shodan, to quickly identify vulnerable systems and sensitive data in corporate networks.

Shodan can also be used in different areas such as mentioned below as per [4]:

- Network Security: Monitor the internet-connected devices security in the organization
- Market Research: Search and observe the products used by people
- Cyber Risk: Expose vendors online as a risk metric
- Internet of Things: Keep track of devices connected to internet
- Tracking Ransomware: Calculate the risks of ransomware for devices connected to IoT.

4) Information Exploitation: Shodan is an Internet portal that has been scanning the Internet to find open ports on IP addresses and try to determine what services are running on found ports. Moreover, it also provides information about the services, ports, and headers. According to [5], user can also get the information like states of ports, services, and operating system (OS).

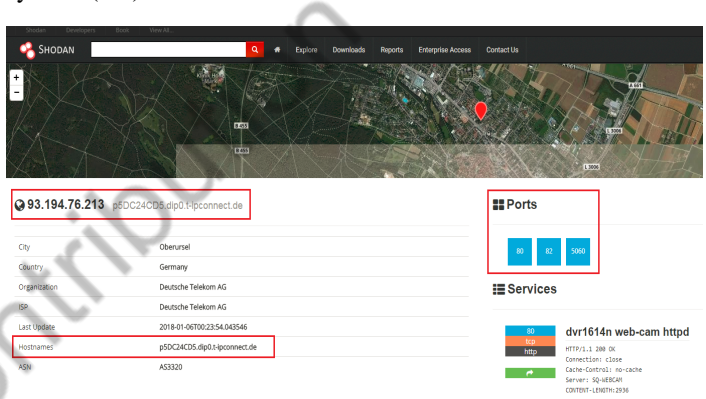

Fig. 2. Information Exploitation by Shodan

Shodan performs the internet-wide scanning using search queries. Following are some of the examples:

```
SERVER: SQ-WEBCAM
LINUX UPNP AVTECH
NETCAM
DEFAULT PASSWORD
DREAMBOX COUNTRY:ES
SERVER: SQ-WEBCAM COUNTRY:"US"
ADMIN+1234
CATEGORY:MALWARE
```

Moreover, if you do not want the text-based search then you can switch to Shodan Maps. You can see the search listing on a map. It also provides the zoom-in facility with which you can narrow down the search results [6].

Following figure 3 depicts the use of a query.

For statistics and breakdowns on various facets of a search query, Shodan can generate a report. This report is sent via email. Following Figure 4 shows the report sample generated for the search query for ”default password”.

5) Accumulated Information: As Shodan provides the information about services, ports, and headers, one can possibly use it to identify owner of the device, location of the device and some additional information such as if the device is vulnerable to attack [7].

Fig. 3. Shodan Query Search Example with Shodan Maps

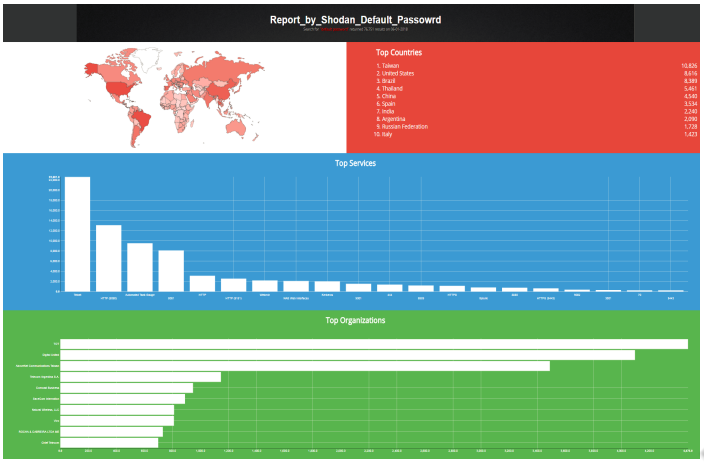

Fig. 4. Report generated by Shodan

6) Implementation Details: Shodan performs the internet-wide scanning using search queries. Shodan uses SYN scanning and banner grabbing method to collect the information of IP devices. After open ports of target device are discovered with SYN scan, banner grab is activated against those ports. The scan result can consist of open-port list of the devices, service and its version on that port [5]. With reference to the [8], following Figure 5 depicts the working flow.

Below given part from [5], explains details about the mechanism of information gathering and detection:

- TCP SYN Scan [5]
  TCP SYN Scan is a well-known scan and used as a default for major network scanning tools. It is also known as half open scan because it does not complete the three-way handshake. With sending SYN probe packet and receiving response packet, scanner can judge whether port is open or not. A port can be decided as open when ACK packet is received, while close when RST packet is received. To detect TCP SYN scan, misuse detection and anomaly detection are used. To find misuse for other TCP Scan, TCP Flag is added on the misuse database. However, it is not effective for SYN Scan because SYN packet is frequently happened in typical network communication. In Anomaly detection, SYN scan is defined as no ACK Packet response after Sending SYN/ACK. It is very easy to model unusual behavior.

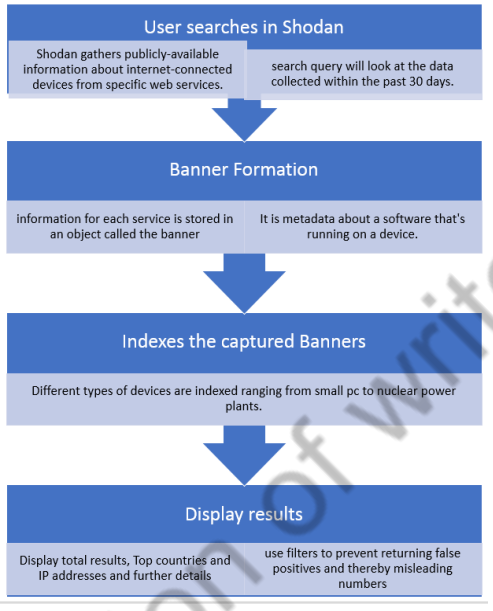

Fig. 5. Shodan search working

- Banner Grab [5]
  Banner Grab is an application layer scan technique and activated when scanner connects to the target host with TCP 3-way hand shaking. in a narrow sense, Banner is the simple text usually includes signatures of service and displayed when the connection is established in several protocols such as FTP, SMTP, POP3 and telnet. In a broad sense, it includes every behavior gathering information from the open port (i.e. GET method in HTTP). Process of banner grabbing is similar with TCP Connect Scan. Its detection is also based on misuse detection and anomaly detection.

- Horizontal Scan [5]
  This mechanism is focused on a range of the targets not a behavior. horizontal scan means that scan a single port on multiple hosts. Both search engine is According to the result of Shodan, each scanned port has different time that cannot consider to vertical scan. On Zmap which Censys uses for port scan, only horizontal scan is implemented.

- Distributed Scan [5]
  Those search engines cover a scale of whole IPv4 addresses. Thus, using multiple scanner is reasonable. Previously, it is mentioned that Shodan has distributed servers around the world and Censys uses scheduler and multiple scanners. Because each scanner has own IP Addresses unless they are behind NAT, it is one of challenges to detect and trace the distributed scan.

Shodan check horizontally whether the port of target is open or not using TCP SYN scan. After that, they grab the banners against the host with open port and upload it on the web site. [5].

Shodan also uses basic filters such as Shodan Maps and Shodan Exploits. Although, one can integrate Shodan with penetration testing tool such as Metasploit, Maltego and Nmap

[2].

B. *Censys*

1) Description: Censys is a search engine and data processing unit, that allows users and researchers to ask questions about devices and networks in the the whole Internet. It is driven by Internet-wide scanning tools called ZMap and ZGrab, an application layer scanner. ZMap itself is a complete data harvesting software, that *pings* more than four billion IP addresses every day. ZMap combined with ZGrab allows the identification of specific protocols and vulnerabilities of identified hosts. The collected data is then stored in a database by *Censys* and can be accessed through an open search engine, a programmatic REST API, Google BigQuery and Raw download of data by anyone. A query in the search engine throws up all the technical details of the search term including its certificate. The data that comes back can identify what kind of device responded, as well as details about its software, such as whether it uses encryption and how it is configured. [9]

2) Aim: The goal or aim of Censys is to maintain a complete database of every device exposed on the Internet and its provided services, or ”complete database of everything on the Internet”.

3) Information Exploitation: By scanning the whole IPv4 address range *Censys* identifies all occupied IP addresses. Further analysis of these through ZGrab allows it to exploit more detailed information about applications running on the identified addresses. Currently HTTP, HTTP Proxies, HTTPS, SMTP(S), IMAP(S), POP3(S), FTP, CWMP, SSSH, Modbus, StartTLS, Heartbleed and SSLv3 [9] can be identified and the handshakes of these protocols are stored in the database. Censys can also further perform some cipher suite checks.

4) Accumulated Information: Because of the open database access offered by *Censys* via the search engine, programmatic REST API and Google BigQuery, it allows all kind of people to use the data for any purpose. An very obvious usage would be identification of all servers, that still suffer from the Heartbleed bug, since that vulnerability is explicitly identified. Censys could help attackers find targets that have some weaknesses, e.g. using deprecated cipher suites, very efficiently with the offered search engine, and perhaps even automated by using the REST API. But not all usages need to be malicious. By having a ”complete database of everything on the Internet”, researchers could statistically analyse the internet and monitor developments or discover trends. Censys could also help prevent large scale attacks by informing vulnerable devices on the Internet about the potential dangers.

5) Implementation Details: Internally Censys uses three tools to perform its data collection: ”ZMap”, ”ZGrab” and ”ZTag”. These three tools are used within the three steps performed by Censys:

i **Internet-Wide Scanning**
In the first step ZMap [10] is used to perform scheduled internet-wide network surveys. With a 10gigE connection ZMap is able to scan the whole IPv4 address space in under 5 minutes. To achieve the huge volume of scan operations ZMap uses mostly UDP and TCP SYN scans. ZMap also supports ICMP, DNS queries, UPnP and BACNET for probing the network.

ii **Application Scanning**
In the next step Censys is using ZGrab, which performs a banner grab of the services that were identified in the last step. At the writing of this paper ZGrab supported HTTP, HTTP Proxy, HTTPS, SMTP(S), IMAP(S), POP3(S), FTP, CWMP, SSH, and Modbus, as well as StartTLS, Heartbleed, SSLv3, and specific cipher suite checks. [10]

iii **Validation, Extraction, and Annotation**
In the last step ZTag is being used in conjunction with ZMap and ZGrab to validate, extract and annotate [10] the raw scan data with additional meta-data and to transform the records.

C. *Thingful*

1) Description: Thingful.net is a search engine for the Internet of Things, providing a unique geographical index of real-time data from connected objects around the world, including energy, radiation, weather, and air quality devices as well as seismographs, iBeacons, ships, aircraft and even animal trackers [11]. It is a the worlds first search engine for the public Internet of Things [12]!

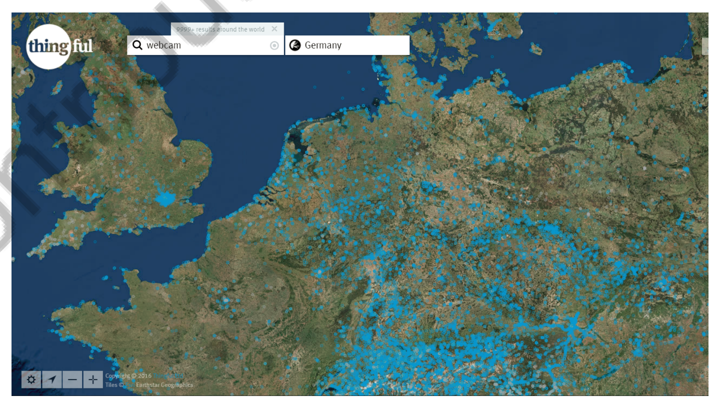

Fig. 6. Thingful basic search

Thingful.net has powerful search capabilities that enable people to find devices, datasets and real-time data sources by Geo-location across several famous Internet of Things networks. It collects the data using a proprietary IoT device data search ranking methodology. Millions of connected objects and sensors across the planet generate real-time open data. Thingfuls Datapipes make it quick and easy to find and use the IoT data thats most valuable as per user query [11].

For quick demonstration of the available sensor nodes all over the world, go to the main website https://www.thingful.net/ and just click enter. You will see the result as shown in figure 7. These sensor nodes publish their data publicly over the internet. Figure shows that some part of the world are poorly connected with the sensor nodes.

2) Aim: To enable people to find devices, datasets and real-time geo-located data sources across many popular IoT networks. The Thingful goal is to make data more valuable and meaningful [11].

On the other hand, as per [11], Thingful also aims to contribute in:

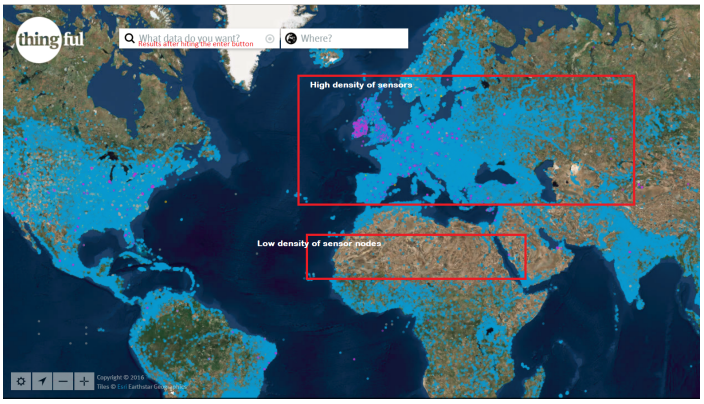

Fig. 7. Thingful Sensor Nodes all over the world

- Air quality Monitoring
- Measuring Radiation
- Flood Monitoring
- Thingful data for analysis, visualization, triggers, notifications or citizen-centric 'smart city' apps, among other things [13].

3) Adoption Context: Thingful data can possibly used for analysis, visualization, triggers, notifications or citizen-centric 'smart city' apps, among other things. As per information given here [13], it can be used by an open source mobile app which access the Thingful API to discover the air quality, weather and transport data dynamically so that it can help for finding better mode of transport. Thingful can be also used for finding and accessing cross-domain urban IoT data [14].

This article [15] states that connected vehicles such as cars produces and consumes huge amount of data and are connected to the network. Moreover, connected vehicles communicate with local and remote services in real-time. As per this article [15], following are the use cases for the connected vehicular data include telematics-based insurance, remote diagnostics for maintenance, real-time feedback on driving behavior and vehicle life-cycle management.

4) Information Exploitation: The visual interface of Thingful presents data using a proprietary patent-pending geospatial device data search ranking methodology [16]. By enabling the filters, as shown below in Figure 8, users can access the specific information.

Thingful visual interface also has the feature of zoom-in which helps user to narrow down to the specific region or area. Here, figure 9 shows the webcam in TU Darmstadt with its view count.

5) Implementation Details: Thingful uses proprietary search algorithms to identify exactly the data you need, and have it piped to you on demand or on a schedule [16]. As per [17], Thingful indexes all IoT data repositories across the world, such as environment, traffic, health and technology sensors. Some of them are shown in this Figure 8 as filters. These objects are interconnected and give an of geo-location and time-series data. These sensor nodes collect all required information and can help to identify the things happening around the world.

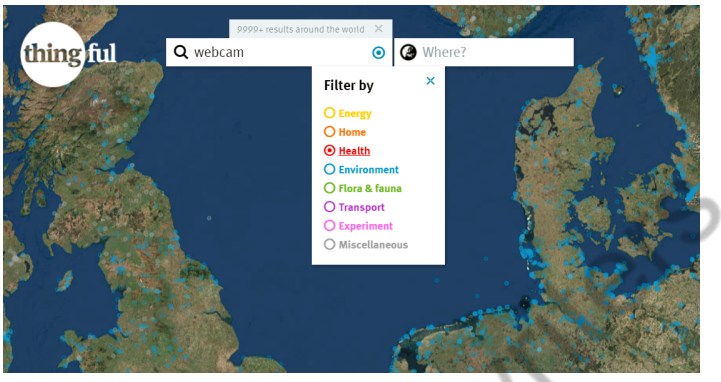

Fig. 8. Thingful Filters

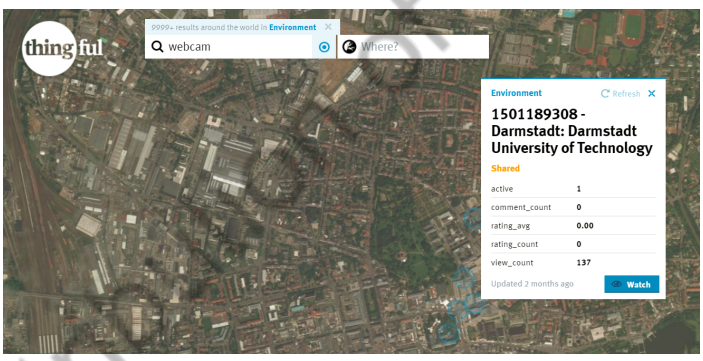

Fig. 9. Thingful Zoom-in

D. *PunkSPIDER*

1) Description: PunkSPIDER is a global web application vulnerability Search Engine [18].

2) Aim: The aim of PunkSPIDER is to allow the user to determine vulnerabilities in websites across the Internet quickly, easily, and intuitively [19]. Moreover, it also aims to find the number of vulnerabilities present on the given URL or site [19].

3) Information Exploitation: According to [19], following types of Vulnerabilities PunkSPIDER can Map:

- BSQLI = Blind SQL Injection
- SQLI = SQL Injection
- XSS = Cross Site Scripting
- TRAV = Path Traversal
- MXI = Mail Header Injection or Email Injection
- OSCI = Operating System Command Injection
- XPATHI = XPath Injection

following figure 10 shows the scan result for search query "webcam". Important things are highlighted in red color such as the site in not secure, search query and vulnerability results.

4) Accumulated Information: As per [20], after scan, you should receive a result back that looks like the following figure 11.

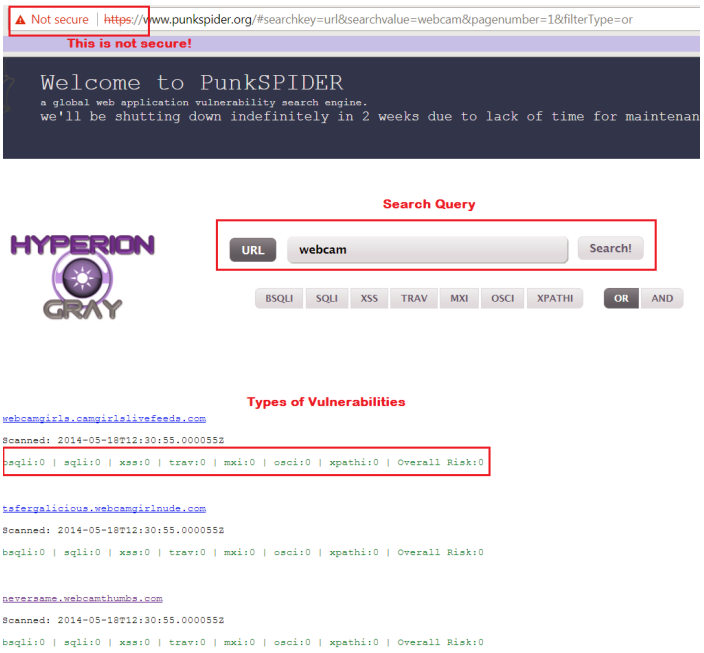

Fig. 10. PunkSPIDER Demo

```
1 www.nytimes.com
2 Scanned: 2014-05-18T12:30:55.000055Z
3 bsqli:0 | sqli:0 | xss:0 | trav:0 | mxi:0 | osci:0 | xpathi:0 | Overall Risk:0
```

Fig. 11. PunkSPIDER scan results

According to the description in [20], the first line shows the domain of the result. The timestamp field on line 2 is the time that the site was added to our system. Below that is the interesting part, the total number of vulnerabilities found on the website. If youre non-technical, you can ignore almost every part of that and just look at the Overall Risk field this will tell you the risk of visiting a website. As a rule of thumb anything with an Overall Risk of 1 should make you very wary, anything with an Overall Risk of greater than 1 you should stay away from entirely.

III. Personal/User-interaction based Scanning Tools

A. *IVRE*

1) Description: IVRE (Instrument de veille sur les rseaux extrieurs) or or DRUNK (Dynamic Recon of UNKnown networks) is an open-source framework for network recon [21]. It relies on open-source well-known tools (Nmap, Zmap, Masscan, Bro and p0f) to gather data (network intelligence), stores it in a database (MongoDB), and provides tools to analyze it [21].

As described features on [21], IVRE works as a Satellite imagery for cyberspace which shows the World map of Internet-exposed Modbus devices. It can be used for Data analysis, as IVRE includes tools to analyze data gathered from Network scans. Based on analysis, user can browse the scan results as IVRE comes with the Web interface. Additionally, it provides a dedicated Web interface which allows for flow analysis.

2) Aim: The goal of IVRE is to perform reconnaissance for network traffic [22]. Meaning, doing intrenet-wide scanning, gathering the information and investigating the gathered information to explore the potential vulnerable resources and their activities [21].

3) Adoption Context: IVRE is commonly used for digital forensics, information gathering, intrusion detection, or network analysis. Target users for this tool are pentesters, security professionals, and system administrators [22].

4) Information Exploitation: IVRE means "Instrument de veille sur les rseaux extrieurs". It is French for DRUNK, Dynamic Recon of Unknown NetworKs. The IVRE framework allows both active and passive data gathering. [22]

The main reference [21] describes about the basic steps that needs to be performed to gather the information:

- Scan and Sniff [21] IVRE has tools to run Nmap or Masscan against targets such as a network or an address range, a whole country, a specific AS, or the full IPv4 connected address space. It can use Zmap for a fast pre-scan, and collect info from network traffic (passively) using Bro, Argus, Nfdump and p0f.
- Browse [21] To browse the results, one can use the CLI tools, the Python API or the Web interface. To find the specific services or vulnerable versions, within a specific country or network, quickly access to previous results for a specific host, etc, filters can be used. Belwo figure 12 from [23] shows the use of filters.

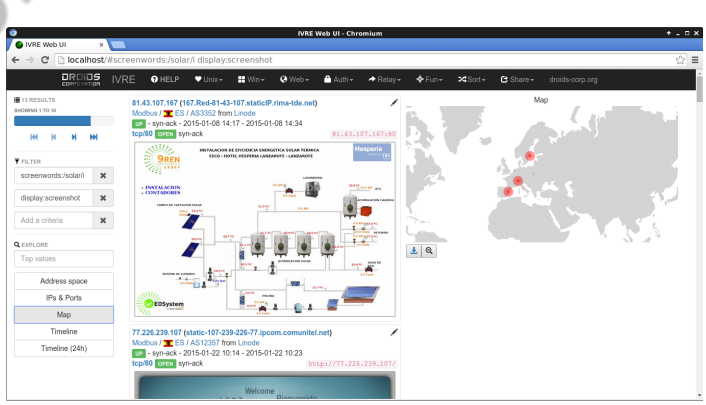

Fig. 12. IVRE Scan results details: using filters "solar" and map

- Analyze [21] Make the best of your scan results to identify similar hosts and corner-cases. Look for most (and least) common ports, services or products, and get a quick overview of the address space with the heatmap. Following figure 13 from [23] shows the Nmap scan results with overview of the address space with heatmap.
  Next figure 14 from [23] shows the scan result details, using the "heatmap" IP addresses to "zoom" in the address space.

Additionally, according to [21], IVRE provides another unique tool to visualize and analyze the complete network flow. It is called IVRE Flow tool which analyzes the network flow

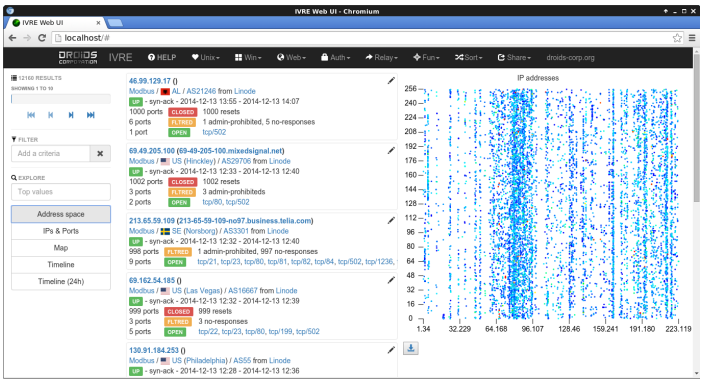

Fig. 13. IVRE Nmap results : Home page with "heatmap" IP addresses

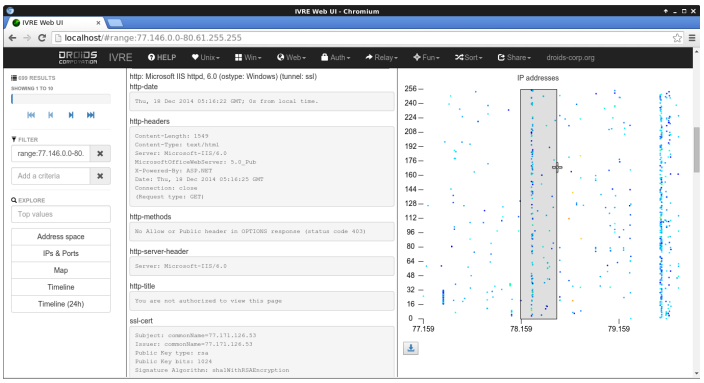

Fig. 14. IVRE Scan results details: using the "heatmap" IP addresses to "zoom" in the address space

among the hosts. This tool can be considered as a recon tool for the case of unknown network, can be a cartography tool to get a better understanding of a supposedly known network and can be a monitoring tool to spot unwanted flows in your network [21]. Below image 15 from [23] shows flow details for a specific host.

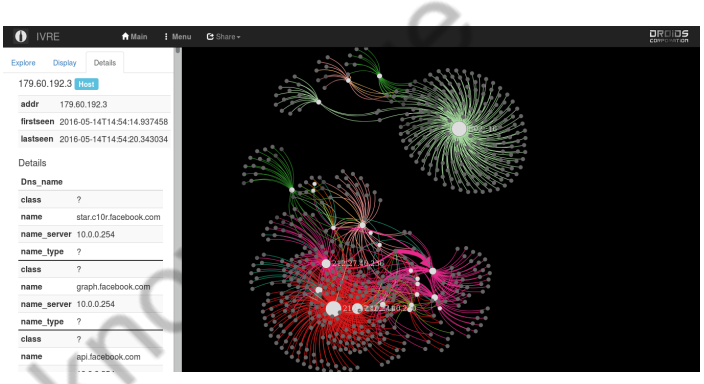

Fig. 15. IVRE Flows with details for a specific host

5) Implementation Details: IVRE framework is written in Python with a MongoDB backend. The description in [21] describes how it works:

IVRE makes use of the data from different sources such as Argus, Bro, Masscan, Nmap, zmap, and others. It extracts the data and saves it in MongoDB instance for later analysis. Neo4j database is also used in the network flows determination. Command-line, web interface, or the Python API is used to extract and display the data.

Based on the reading from [21], IVRE performs two kinds of recon : one is passive network recon and the other is active network recon which is Nmap-based. For example, following command run a standard scan against 1000 random hosts on the Internet by running 30 nmap processes in parallel [21].

$$\begin{aligned} &ivre \quad runscans \quad --routable \\ &--limit \quad 1000 \quad --output = XMLFork \end{aligned} \tag{1}$$

Another command is used to get all the hosts with the port 22 open :

$$ivre \quad scancli \quad --port \quad 22 \tag{2}$$

IVRE also uses Master-Slave model during the installation of IVRE Agent. Basically the computer running the IVRE is a Master and IVRE agents can run on the slaves.

B. *Vulners*

1) Description: Vulners is a very large database for information security content, that is being updated constantly. I provides a search engine to query for vulnerabilities, exploits, patches, and bug bounty programs.

2) Aim: Provides an extensive database for all known security related information by aggregating information from more than 70 sources. [24]

3) Information Exploitation: Vulners aggregates and displays security related information in six major types of data:

i Vulnerabilities, containing a general description and links
ii Vendor's security bulletins. These are bug-reports, which are published by software vendors about vulnerabilities in their own products.
iii Exploits, which are displayed in a convenient text editor
iv Nessus plugins for vulnerability detection. Which displays whether a particular vulnerability can be detected by Nessus.
v Bug disclousers for bug bounty programs
vi Publications from hacking resources

4) Implementation Details: The engine of Vulners is written in Python and Django and it uses MongoDB and Elasticsearch databases.

C. *Nessus*

1) Description: Nessus is a proprietary vulnerability scanner designed to automate the testing and discovery of known security problems.

2) Aim: The main aim of the scanning tool is to discover vulnerabilities in the network by scanning ports and services running or listening to that port. The hosts connected to that network can also be scanned (specifically scanning for the installed softwares that might be vulnerable) that are installed on those hosts to protect them from criminal hacking. At the time of writing this paper the version was Nessus 6.11. [25]

3) Information Exploitation: Nessus can perform a big range of vulnerability scans, the main types are: [26]

- Asset discovery scan: Figuring out which systems or hosts exist which ports are open and which services are listening on those ports and which OS are running on those systems. Scan a range of IP addresses
- Network Vulnerability scan: To figure out which hosts were vulnerable and at which ports.
- Patch Auditing: (using credentials and plugins that uses credentials) To check out all the missing vulnerability patches of the OS on every hosts as well as missing patches on some of the third party softwares.
- Configuration Auditing: (Also with credentials) Logs into system looks in the configuration and compares it to the setting with a .audit file
- Web Application Fuzz Testing: Finding previously-unknown web application vulnerabilities using fuzzing techniques.

D. *skipfish*

1) Description: skipfish is an active web application security reconnaissance tool. It recursively crawls the targeted website and performs a number of security checks in the process.

2) Aim: The goal is to generate a final report that can be serve as a foundation for professional web application security assessments.

3) Information Exploitation: skipfish can identify a high number of vulnerabilities, which mostly can be categorised in the following categories:

- High risk flaws (potentially leading to system compromise): [27]
  - Server-side SQL / PHP injection
  - Explicit SQL-like syntax in GET or POST parameters.
  - Server-side shell command injection (including blind vectors).
  - Server-side XML / XPath injection (including blind vectors).
  - Format string vulnerabilities.
  - Integer overflow vulnerabilities.
  - Locations accepting HTTP PUT.
- Medium risk flaws (potentially leading to data compromise):
  - Variety of XSS detections
  - Directory traversal / file inclusion (including constrained vectors).
  - Variety of MIME vulnerabilities
  - Many more...
- Low risk issues (limited impact or low specificity):
  - Directory listing bypass vectors.
  - SSL certificate problems
  - HTTP form submissions
  - Many more...
- Internal warnings
- Non-specific informational entries

E. *Acunetix*

1) Description: Acunetix is an automatic web vulnerability scanning tool, best suited for web application. This tool can scan any sort of websites or applications that are accessible over a web browser. This tool does not study the source code of the web application. That means it applies the Black and Gray Box scanning method. [28]

2) Aim: It is commonly used for penetration test, security assessment, or vulnerability scanning. Target users for this tool are pentesters, security professionals, and system administrators.

3) Information Exploitation: When using Acunetix to scan a website, it will provide a detailed list of security issues. These information are efficiently organized, offering details on the problem. It will also try to give advice to how solve the problems.

4) Implementation Details: Acunetix is using a three step process to generate it's final report: [29]

i Crawling Process
- A crawler (DeepScan) is used to analyze the whole website by following the links. It can also scan the dynamically generated links, such as Links that are constructed using JavaScript. It can also craw the sitemap.xml file if it is available.
- Along with the crawler if their AcuSensor Technology is activated then it can also access the files that are generally not accessible normally from the internet. Files, such as web.config, can be analyzed if the AcuSensor Technology is used. It also can run a back-end craw, that way it checks for files that are not linked through front end application but might have be placed by an attacker.

ii Vulnerability attacks / test
- Launches Vulnerability check (emulating a hacker behavior). General attacks, such as cross site scripting, SQL injection are launched and also input fields are also tested with different input combinations. If the AcuSensor is on then it checks for some more vulnerabilities such as Blind SQL injection, directory traversal.

iii Reporting process
- Scan results could be very detailed based on the craw option used. With AcuSensor on, details such as source code line number is

also reported. The report not only includes the vulnerabilities but it also recommends the fixes on those found vulnerabilities.
- Various types of reports such as Executive summary report, developer specific report could be generated.
- Two scan results for the same target can be compared [28]. It also provides an audit of the web server the web application is running on by running port scans. A report on network vulnerability includes information about the operating system and the services. The scanner also integrates OpenVas to check for network vulnerabilities.

F. *Vega*

1) Description: Another free and open source web security scanner and testing platform is Vega. It supports multiple scanning modes, from fully automated to manual [30] and is very extensible.

2) Aim: The goal of Vega is to find vulnerabilities of a website, fully automated or as part of an interaction with a certain website.

3) Information Exploitation: Vega is able to identify a number of typical web vulnerabilities like reflected cross-site scripting, stored cross-site scripting, blind SQL injection, remote file include, shell injection and TLS/SSL configurations. Specifically for TLS/SSL security settings, Vega tries to identify ways to improve a website's TLS/SSL security. The scanner can be used in three different modes [30]:

i **Automated Scanner** In that mode Vega is crawling a website fully automated. By providing certain user credentials, the scanner is also able to automatically perform logins.

ii **Intercepting Proxy** When Vega is run in proxy mode it can be used to observe the communication between clients and servers. It is not purely observing the communication, but can also interact with it.

iii **Proxy Scanner** Vega can also be used to perform user-driven, so semi-automated web security testing. In that mode it will scan the target site, which the user is browsing.

4) Implementation Details: Vega is written in Java and can therefore be used on Linux, OS X and Windows. The detection modules are written in JavaScript. Additional modules (extensions) can easily be created because of the extensive API exposed by Vega [30].

IV. Pros and Cons of Scanning Tools

A. *Shodan [31] [32]*

1) Pros:

- Create reports based on search query
- Simple search query using words
- Support text-based search and Shodan map
- Easy and free to access website. Fetch required data for free of cost
- Data after search results can be downloaded in different format such as JSON, CSV or XML.
- Knowledge sharing such as a shared search queries in community

2) Cons:

- It is not open source. The code and implementation details are unavailable.
- Due to publicly availability of data, this search engine can be misused by hackers to identify and hack into the devices connected to internet
- Shodan does not process sensor outputs
- Catching everyday objects (capturing live data) on this website is still difficult

B. *Censys*

1) Pros:

- Easy to use search engine (Similar google for internet service querying)
- Reports about security related protocols
- Provides an API to get the raw data and use the search engine from third-party applications
- Allows dedicated querying for hosts, websites or certificates
- Provides extensive filtering capabilities (by country, hosts, protocol, alexa rank, tags, etc...)
- Provides detailed information about the search target
- Passive scan

2) Cons:

- Requires account creation after a certain amount of interaction with the search engine
- Using the API is rate limited by token buckets
- No vulnerability analysis of targets by censys itself
- Data is available, but needs to be interpreted by the user
- Mostly relies on the older data (since the last censys scan)

3) Limitation:

- Does not support IPv6
- The devices behind a NAT can't be accessed

C. *Thingful [31] [32]*

1) Pros:

- Gathers large-scale data and support vast index of multi-domain data
- Sensors are used to collect the data
- Provides a unique geographical index of real-time data from connected objects around the world
- Easy and free to access website

2) Cons:

- It is not open source. The code and implementation details are unavailable.
- Public limitation on the availability of the collected data
- Thingful provides access to its data only via a dedicated UI
- Fast expiration of the data due to the highly dynamic nature of the IoT connected devices
- Unable to tap data where sensors are sparse
- To collect the temperature data at a particular location on the planet, chances of finding the existing sensor depending on the given access, are limited.

D. PunkSPIDER

1) Pros:

- It is global web application Vulnerability Search Engine
- It takes URL as a search query input
- Helps to determine the vulnerabilities in websites across the Internet

2) Cons:

- Unable to test the working due to unavailability of the website
- Manually need to input the website name
- It is currently not working (January 2018)
- does not really fit in IoT scanning tools category

E. IVRE [21]

1) Pros:

i Free software and open source framework
ii Works with Web UI and command line
iii It auto scans the network
iv IVRE Agent : meant to run in environment which is not really controlled
v IVRE Flow: for analyzing the network flows among the hosts
vi IVRE web UI : central view is a graph representing the network
vii Supports interaction with graphs
viii Provides data- flow filters for analysis

2) Cons:

i Requires manual setup. Installation and setup is time consuming
ii Strongly depends on the use, environment and purpose of the implementation
iii IVRE is enumerated and not a simple list, while the others are lists
iv Mostly has external dependencies. It relies on python, Nmap, Bro, MongoDB, etc. Therefore, difficult to maintain the version and compatibility among the behavior of external programs or dependencies.
v There are some inconsistencies among the IVRE Flow
vi Need to improve the customization possibilities in the framework

F. Vulners

1) Pros:

- As soon as updates arrive for a particular query, a user gets automatically notified by Vulners. (Subscriptions)
- Extensive filtering
- Very detailed description of vulnerabilities
- Provides auditing to check for vulnerabilities of a certain OS family, version and packages
- Provides an API for third-party tools
- Free to use search engine

2) Cons:

- On-site scanner is not free

G. Nessus

1) Pros:

- Can run on windows and linux (multi-platform)
- Uses a plugin based architecture to detect vulnerabities (extensible)
- Provides proxy support with authentication
- Targets can be queued up and scanned automatically
- Client/server architecture allows test automation
- Simple graphical front-end
- Supports multiple IDS evasion techniques
- High quality tests
- Current data

2) Cons:

- the free version (Nessus Home) is constrained to scans of only 16 IPs
- the professional edition is very expensive
- Real-time updates to the scan database require a subscription.
- Limited HTTP authentication support
- Active scanning

3) Limitation:

- Can't perform internet-wide scans

H. skipfish

1) Pros:

- Current data
- free to use, open source
- Generate extensive report

2) Cons:

- Active scanning
- Not easy to install, since it needs to be compiled from source
- Not very user-friendly, because it has no user-interface
- Only command-line

I. Acunetix

1) Pros:

- Easy to use
- In depth crawl and analysis
- Generates extensive vulnerability report
- Current data

2) Cons:

- Can only be used for web applications
- Generates a high number of garbage records in target backends [33]
- Free version is only valid for one year
- Active scanning

3) Limitation:

- Vulnerability analysis limited to web applications

J. Vega

1) Pros:

- Cross-Platform
- Ease to use graphical user-interface, based on the eclipse platform
- Automated Scanner
- Easy to extend by using the rich API exposed by Vega
- Detailed report of detected vulnerabilities

2) Cons:

- Active scanning
- No automatic target detection

3) Limitation:

- Limited to web applications

V. Comparison among Public Scanning Tools

One obvious comparison needs to be done for Censys and Shodan, because they seem to be very similar. Indeed, they both try to solve the same problem. In the end, it mostly comes down to a matter of taste between these two tools, because they both have powerful filtering, huge data sets and third-party tool support by providing a powerful API. A few things we preferred from one tool to another is the user interface of Shodan, which is more user-friendly and focuses more on presenting the results in the context of the location. But, we have found the banner grabs of Censys to be much more detailed than the ones in Shodan. If a user has never used any of these tools, then Censys also does a better job at explaining how to filter and search in general, by providing a very helpful dialog. Since Shodan provides an Enterprise edition of their platform, it could be preferable for organizations since it provides a lot more analytical functionalities to enterprise clients, which Censys does not provide at all. Following Table 16 shows comparison between shodan and censys:

Tools	mechanism of information gathering and detection	SYN Scan Tool	Required input	IP Device Search Engine	Scan Range	scan server	Target ports	Scan period
Shodan	SYN Scan / Banner grab	Unknown	specific query syntax	Yes	Horizontal Scan	Distributed	41 ports	Unknown
Censys	SYN Scan / Banner grab	Zmap and Zgrab	word or phrase based	Yes	Horizontal Scan	Distributed	35 ports	Automatically scheduled

Fig. 16. Comparison between Public Scanning Tools [5]

The main difference between Shodan/Censys and IoT search engines such as Thingful, is that Shodan/Censys is basically designed as a search engine for hackers [32] as the data is publicly available and can be misused by the intruders to gain unauthorized access to the machines. Moreover, It identifies and hacks into password protected devices which are connected to the Internet. Even, servers, routers as well as other Internet-connected devices have been archived with their IP addresses in its database. Unfortunately, the website itself does not process the sensor outputs so the access to live and updated data is troublesome. Due to its large and broad scope, catching everyday objects on this website is still difficult while servers and network devices constitute the majority of the things in its database.

According to Shemshadi, et. al [32], the only working example of the IoT search engine is Thingful and none of the IoT search engines in the literature have been deployed for real-world or large-scale data. Furthermore, the Thingful initiation itself is still limited, and a significant progress is needed to expand this area. One instance of such limitations is the public availability of the collected data. For example, Thingful provides an access to its data only via a specific User Interface. Another example of the limitations is the fast expiration of the data due to the highly dynamic nature of the IoT.

Below table 17 refers to a short summary about the scanning tools.

VI. Conclusion

As the Internet becomes more popular it is also expanding. The expansion happens by connecting more and more devices

Tools	Website for accessing the gathered data	Framework and need manual installation	Open source code	Proprietary	Use for Free	Add-on features with subscription	Search query input
Shodan	✔	x	x	✔	✔	✔	Search query syntax (words)
Censys	✔	x	(✔)	✔	✔	✔	Words/phrase
Thingful	✔	x	x	✔	✔	x	words/phrase
PunkSPIDER	✔	x	x	✔	✔	x	URL
IVRE	✔	✔	✔	x	✔	x	words/phrase
Vulners	✔	x	x	✔	✔	✔	words/phrase
Nessus	x	✔	x	✔	(x)	(✔)	-
Skipfish	x	✔	✔	x	✔	x	-
Acunetix	x	✔	x	✔	(x)	(✔)	-
Vega	x	✔	✔	x	✔	x	-

Legends: ✔ - Yes (✔) - Partly true x - No (x) - Partly false

Fig. 17. Summary about Scanning Tools

to the Internet. One of the latest trends of expansion is the Internet of Things. Often these devices are connected to the Internet by leaving security and privacy an after thought, resulting in a lot of devices being vulnerable. Scanning the Internet for these devices is of great interest for researchers, which is why multiple scanning tools were developed which are capable of finding these devices. This paper identified a few tools that perform Internet-wide scans, and publish these findings on a public website. Additionally, the paper discussed tools that perform vulnerability scans, but do not publish anything on public websites. We evaluated and compared these tools within their application context.

In conclusion, the Internet-wide scanning tools provide a lot of data about the devices and services, which is filterable by using their search engine or API, but they generally provide no or very limited analytical functionalities. These functionalities are provided by the vulnerability scanners that we also explored in the paper, but they require known targets or have only very limited automatic network scanners. Consequently these different tools complement each other, and form very effective tools in combination.

The existence of these tools has two sides, the positive being, that vulnerable devices could be warned about dangers preemptively, whereas the negative side would be that malicious users can use these tools to do damage to the vulnerable devices. We anticipate that just being listed in Censys or Shodan would increase the risk (frequency) of being attacked dramatically.

ACKNOWLEDGMENT

We would like to thank Prof. Dr. Max Mhlhuser, Rolf Egert, Dr.-Ing. Andrea Tundis and Nikolaos Alexopoulos for giving us this opportunity and support. Specially, Dr.-Ing. Andrea Tundis for providing insight into the topic and constructive advice during the seminar work. We are grateful for the assistance given by the department.

REFERENCES

[1] S. Kai, M. Cornelius, and K. Joerg, "Contactless vulnerability analysis using google and shodan," accessed January 16, 2018. [Online]. Available: http://jucs.org/jucs_23_4/contactless_vulnerability_analysis_using/jucs_23_04_0404_0430_simon.pdf

[2] J. Matherly, "Shodan official website," accessed November 4, 2017. [Online]. Available: https://www.shodan.io/,https://enterprise.shodan.io/

[3] "Google hacking diggity project," accessed November 4, 2017. [Online]. Available: http://www.bishopfox.com/resources/tools/google-hacking-diggity/

[4] S. H. Center, "Shodan help center." [Online]. Available: https://help.shodan.io/

[5] S. Lee, S. H. Shin, and B. h. Roh, "Abnormal behavior-based detection of shodan and censys-like scanning," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2017, pp. 1048–1052.

[6] achillean, "Introducing shodan maps," accessed January 16, 2018. [Online]. Available: https://shodanio.wordpress.com/2014/02/18/introducing-shodan-maps/

[7] V. J. Ercolani, M. W. Patton, and H. Chen, "Shodan visualized," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, Sept 2016, pp. 193–195.

[8] S. Verma, "Searching shodan for fun and profit," accessed January 16, 2018. [Online]. Available: https://www.exploit-db.com/docs/33859.pdf

[9] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by Internet-wide scanning," in *22nd ACM Conference on Computer and Communications Security*, Oct. 2015.

[10] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications."

[11] Umbrellium, "Thingful official website," accessed November 4, 2017. [Online]. Available: http://umbrellium.co.uk/initiatives/thingful/,https://www.thingful.net/

[12] thingful uh, "Thingful blog," accessed January 16, 2018. [Online]. Available: http://blog.thingful.net/post/85807476836/welcome-to-thingful

[13] Umbrellium, "Thingful github," accessed November 4, 2017. [Online]. Available: https://thingful.github.io/

[14] thingful uh, "Thingful blog londoncambridge cycling finding," accessed January 16, 2018. [Online]. Available: http://blog.thingful.net/post/149362243876/showcase-londoncambridge-cycling-finding

[15] thingful uh and K. Usamah, "Thingful blog connected vehicles leveraging iot data," accessed January 16, 2018. [Online]. Available: http://blog.thingful.net/post/149455464806/showcase-connected-vehicles-leveraging-iot-data

[16] V. M. Aceves, E.; Larios, "White paper: Data visualization for georeferenced iot open data flows for a gdl smart city pilot," 2015. [Online]. Available: https://smartcities.ieee.org/images/files/pdf/davgdl_iotvisualinterface.pdf

[17] thingful uh, "Thingful blog virtual sensor project," accessed January 16, 2018. [Online]. Available: http://blog.thingful.net/post/149696551076/virtual-sensors-using-thingful-and-data-science

[18] H. Gray, "Punkspider official website," accessed January 16, 2018. [Online]. Available: https://www.punkspider.org/

[19] Darknet, "Darknet blog," accessed December 5, 2017. [Online]. Available: https://www.darknet.org.uk/2016/09/punkspider-web-vulnerability-search-engine/

[20] A. Caceres, "Punkspider search help," accessed January 16, 2018. [Online]. Available: https://hyperiongray.atlassian.net/wiki/spaces/PUB/pages/10190871/PunkSPIDER+Search+Help

[21] P. Lalet, F. Monjalet, and C. Mougey, "IVRE, a network recon framework," https://github.com/cea-sec/ivre/, accessed January 16, 2018. [Online]. Available: https://ivre.rocks/

[22] L. S. Expert, "Ivre tool review," accessed January 16, 2018. [Online]. Available: https://linuxsecurity.expert/tools/ivre/

[23] P. Lalet, F. Monjalet, and C. Mougey, "IVRE, screenshots," https://github.com/cea-sec/ivre/, 2011–2017, accessed January 16, 2018. [Online]. Available: https://github.com/cea-sec/ivre/blob/master/doc/SCREENSHOTS.md

[24] Vulners.com, "Vulnerability data base - 'google' for hackers," accessed January 15, 2018. [Online]. Available: http://vulners.com/landing

[25] H. Anderson, "Securityfocus printable infocus 1741 - introduction to nessus," accessed January 16, 2018. [Online]. Available: http://cryptomex.org/SlidesSeguRedes/TutNessus.pdf

[26] tenable, "Vulnerability reports generated by nessus," accessed November 26, 2017. [Online]. Available: https://www.tenable.com/products/nessus/sample-reports

[27] Z. Michal, H. Niels, and R. Sebastian, "skipfish - web application security scanner," accessed January 16, 2018. [Online]. Available: https://code.google.com/archive/p/skipfish/wikis/SkipfishDoc.wiki

[28] E. Erturk and A. Rajan, "Web vulnerability scanners: A case study," *CoRR*, vol. abs/1706.08017, 2017. [Online]. Available: http://arxiv.org/abs/1706.08017

[29] acunetix, "Introduction to acunetix - why you need to secure your web applications," accessed January 16, 2018. [Online]. Available: https://www.acunetix.com/support/docs/introduction/

[30] subgraph, "Vega vulnerability scanner and web security testing platform," accessed December 3, 2017.

[31] M. Sheng, Y. Qin, L. Yao, and B. Benatallah, *Managing the Web of Things: Linking the Real World to the Web*. Elsevier Science, 2017. [Online]. Available: https://books.google.de/books?id=q0PQDAAAQBAJ

[32] A. Shemshadi, Q. Z. Sheng, W. E. Zhang, A. Sun, Y. Qin, and L. Yao, "Searching for the internet of things on the web: Where it is and what it looks like," *CoRR*, vol. abs/1607.06884, 2016. [Online]. Available: http://arxiv.org/abs/1607.06884

[33] N. Suteva, D. Anastasov, and A. Mileva, "One unwanted feature of many web vulnerability scanners," 2015.