Article

Detecting IoT Devices and How They Put Large Heterogeneous Networks at Security Risk

Sharad Agarwal [1,2†*], Pascal Oser [1,3] and Stefan Lueders [1]

1 European Organization for Nuclear Research (CERN), Geneva, Switzerland
2 University of Wisconsin Madison, Madison, WI 53706, USA
3 Ulm University, Helmholtzstraße 16, 89081 Ulm, Germany
* Correspondence: sharad.agarwal@cern.ch; Tel.: +33-769-465-489 (S.A.)
† Current address: European Organization for Nuclear Research (CERN), 1211 Geneva 23, Switzerland

Received: date; Accepted: date; Published: date

Abstract: The introduction of the Internet of Things (IoT), i.e. the interconnection of embedded devices over the Internet, has changed the world we live in from the way we measure, make calls, print information and even the way we get energy in our offices or homes. The convenience of IoT products, like CCTV cameras, IP phones, and oscilloscopes, is overwhelming for end-users. In parallel, however, security issues have emerged and it is essential for infrastructure providers to assess the associated security risks. In this paper, we propose a novel method to detect IoT devices and identify the manufacturer, device model, and the firmware version currently running on the device using the page source from the web user interface. We performed automatic scans of the large-scale network at the European Organization for Nuclear Research (CERN) to evaluate our approach. Our tools identified 233 IoT devices that fell into eleven distinct device categories and included 49 device models manufactured by 26 vendors from across the world.

Keywords: Internet of Things; Security; Vulnerabilities and protective measures; Control network security; Operation in multi-user environments; Risk assessment

---

1. Introduction

The Internet of Things has become the latest trend in today's world. For 2020, the installed base of Internet of Things devices is forecast to grow to almost 31 billion worldwide [1]. These days all devices like printers, switches, routers, phones and any other electrical devices are interconnected to increase the ease of access and maintenance, but at the same time it increases the security risk of being compromised.

IoT devices do not have the traditional host-centric security solutions like antiviruses, firewalls, or any safety feature to detect malware. Instead, they all run on certain firmware which is hardware specific, and each type of device has a different protocol on whose principles it runs. As the IoT devices collect a lot of data, these firmwares should be developed by the manufacturers in a secured style, but is rarely the case. Access to the data collected and stored by these devices can aid criminals to gain a lot of sensitive information, like patients' healthcare data or video footage of the cameras.

The European Organization for Nuclear Research (CERN), the world's largest High Energy Physics Laboratory and home to the Large Hadron Collider (LHC), is running a plethora of embedded IoT devices. The users in CERN are from a wide variety of fields, ranging from physicists to all kinds of engineers. CERN hosts four major physics experiments: Compact Muon Solenoid (CMS), Atlas, Alice, LHCb and many other small experiments, which employ a wide range of IoT devices. To name a few, there are programmable logic controllers (PLCs), arduinos, oscilloscopes, thermometers, cameras, and others that are extensively used to run the experiments and the LHC successfully. Like all other international organizations, CERN also operates a large technical infrastructure consisting of other general purpose IoT devices such as CCTV cameras, printers, IP phones. It is important to know the security footprint of that IoT hardware that gets integrated into its network complex: unknown devices can run on firmware versions which are not updated and use old legacy code which introduces vulnerabilities. However, to secure a device, we need to first learn all about the device. With CERN as our primary resource, we argue that insecure IoT devices can escalate the security risk inherent in large heterogeneous networks.

CERN provides easy network access and allows users to set up and register devices which might help them in their work. Users set up their devices on the CERN network by only registering the MAC address of their devices. As the network database does not provide sufficient information to identify and differentiate different IoT devices running on the network, the first step we took was to identify the devices installed at CERN and then did a manual security assessment. As shown in Figure 1. b, we classify the identified IoT devices into four vulnerability classes and adapt this paradigm to the CERN network. Since IoT devices are networked, they are attractive targets and may become the weakest link for breaking into a secure infrastructure [2] or, instead, leaking sensitive information [3] about users and their behaviours [4]. Integrating these unsecured IoT devices in mission-critical networks with industrial control systems, may put directly controlled assets at risk and possibly endanger the whole connected facility. Earlier this year, the European Union Agency for Network and Information Security (ENISA) published guidelines for the development or repositioning of standards, facilitating the adoption of standards and governance of EU standardisation in the area of Network Information Security (NIS) [5], but the manufacturers, consumers and the EU Authorities have not yet fully implemented it.

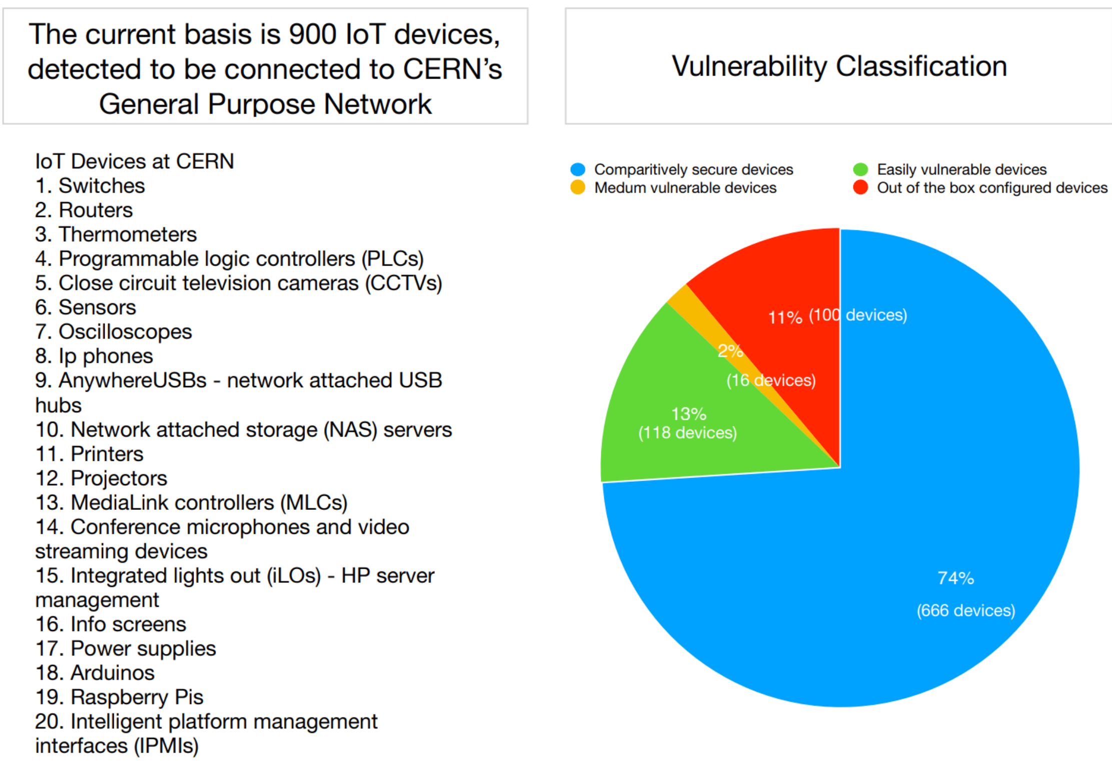

Figure 1. a Overview of IoT Devices at CERN. **b** Vulnerability Classification

In this paper, we identify and present a security assessment of 20 categories of various devices connected to the CERN network, as shown in Figure 1. These 20 categories mentioned in Figure 1 a., were identified by us using our "NetScanIoT" tool. The vulnerability classification in Figure 1 b., was also done by us as per our Vulnerability Assessment results, discussed later in the paper. We not only detected unprotected ports that allow changing the device's configuration but also the devices that are prone to remote code execution. Remote code execution can be used as a gateway for an attacker to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 September 2019 doi:10.20944/preprints201908.0295.v2

Peer-reviewed version available at Sensors 2019, 19, 4107; doi:10.3390/s19194107

gain access to the internal network from the outside and dig further while operating on a trustworthy device.

1.1. Contributions

In this paper, we present an approach to scan the large heterogeneous network without causing faults on remote devices and identify IoT device models based on the web interface. Installing or modifying anything on the device under test (DUT) is not needed. We list our contributions as follows:

- We developed two tools: “NetScanIoT" tool and “Web-IoT Detection (WID)" tool;

  “NetSanIoT" tool detects IoT devices on a large heterogeneous network and is able to detect 20 categories of IoT devices;

  “Web-IoT Detection (WID)" tool identifies the manufacturer name, model and firmware versions of the respective IoT device. It is able to identify 92.45% of IoT device models and 100% of IoT device that have a web user interface;

- We implemented a manual security assessment of 20 categories of devices that were identified by our “NetScanIoT" tool on the highly heterogeneous, large-scale network at CERN.

None of the devices installed at CERN are CERN specific or manufactured only for CERN. The devices installed are manufactured by various vendors from across the world, readily available and also used by other organizations and individuals. The next subsection introduces related work and Section 2 explains the methodologies used for detection of IoT devices and the vulnerability assessment. Section 3 tells about the evaluation of the WID approach following the results of the vulnerability assessment before we discuss our findings.

1.2. Related Work

Most of the related work has identified very few unique categories of IoT devices by scanning a network. Scanning can be done either actively or passively. Active scanning is one-to-one probing communication and passive is where the client listens to every channel’s transmission, which is monitored periodically. Some tools also employ web interface fingerprinting but have assumptions and constraints like working on only single-page applications or analysing the HTTP response messages only [6]. Other tools depend on Nmap [7] port scanning and downloading the landing page using Curl [8] to find the firmware version for the IoT devices [9] which doesn’t work for all IoT devices. An IP-based IoT Device Detection approach requires the knowledge of servers run by the manufacturers and are able to evaluate using only ten device models by seven vendors [10]. Another solution, "IoTScanner", detects by passive measurement identifying devices using packet’s MAC address[11]. We cannot use passive scanning in our tool as we have 1000s of star points at CERN and depending only on MAC address is not sufficiently reliable.

2. Materials and Methods:

This section introduces the different approaches we developed to detect and identify IoT devices and the vulnerability assessment performed by us. The first subsection explains the tools we developed and the next subsection tells about the vulnerability assessment we tried manually on these identified IoT devices.

2.1. Identification

While all devices connected to CERN’s networks need to be registered, CERN does not have a specific database for IoT devices in particular. There are hundreds of devices running on various networks and new devices being installed every day. CERN provides a way for all users and visitors to add their devices to the network by just registering the MAC address of the device. The central

Peer-reviewed version available at Sensors 2019, 19, 4107; doi:10.3390/s19194107

network database of CERN does not have any other detailed information. Therefore, there is no way to distinguish an IoT device from a computer or a cellphone. As we cannot depend only on MAC addresses to identify an IoT device, we developed tools to detect and identify these IoT devices. These tools provide more information about the device, which can help the administrators in maintaining the security of these devices. This subsection explains the tools we developed to solve this problem as follows:

2.1.1. NetScanIoT Tool

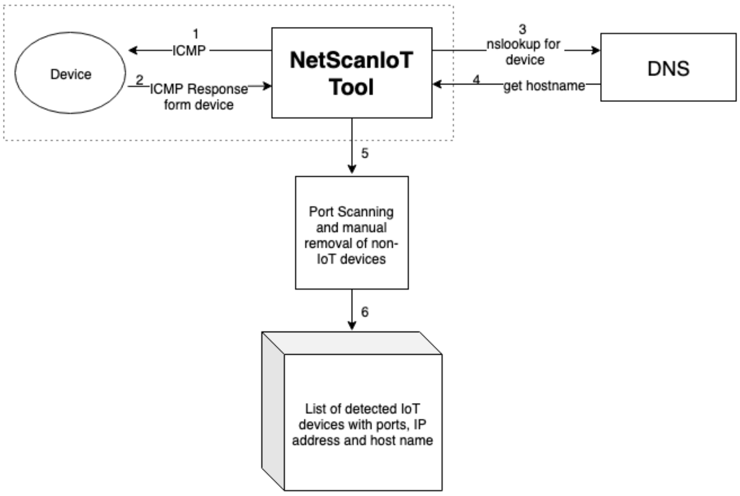

Figure 2. Overview of NetScanIoT Tool

We wrote a Python [12] tool called NetScanIoT, which pings the devices within the network and checks the ICMP [13] message if the target is reachable. If the device response is positive, we go for a nslookup [14] to find the hostname and save the list of the IP addresses along with their hostnames. We pre-filtered the output devices by port scanning and then manually also removed the non-IoT devices from the list connected to the network. We were able to identify 20 categories of IoT devices, as shown in Figure 1. a. Figure 2 shows a graphical working of the NetScanIot tool. With the help of this tool, we were able to detect 900 physical IoT devices.

2.1.2. Web-IoT Detection (WID) Tool

Scraping a web page can be done with many available tools these days, but with so many different manufacturers, the challenge becomes tough. We initially tried to use Wget [15], Curl, Scrapy [16] and other tools, but there are multiple web pages that require to render JavaScript code, which these tools can not. The reason for doing so is that 20% of the device models' web pages render JavaScript first in order to show the complete page source. Therefore, we wrote the WID tool in Python using Selenium [17] in headless mode, which renders the web page with a web driver (Chrome/Firefox) to get the page source. We first analysed the page source manually and identified six classifiers. With the help of Beautiful Soup [18], we used these classifiers and automated the process to find the category of the IoT device. WID not only scrapes the index page of the device but also scrapes the sub-URLs recursively in order to identify the devices' information such as the model and the firmware version running on the device.

Figure 3 shows an overview of the Web-IoT Detection tool and explains the working of the tool to identify the IoT device. With the help of this tool, we were able to identify 233 physical IoT devices, which have a web interface. Figure 4 shows a sample working of the WID tool that takes an IP address as input and identifies the device, model and firmware version. We also present the classifiers in the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 September 2019 doi:10.20944/preprints201908.0295.v2

Peer-reviewed version available at Sensors 2019, 19, 4107; doi:10.3390/s19194107

example, analysed from different sub-URLs that the tool scraped.

The WID tool produces the following output:

- IP-address and the host name of the device
- Web page availability of the device
- Category of device identified
- Manufacturer/Vendor name
- Model name
- Firmware version

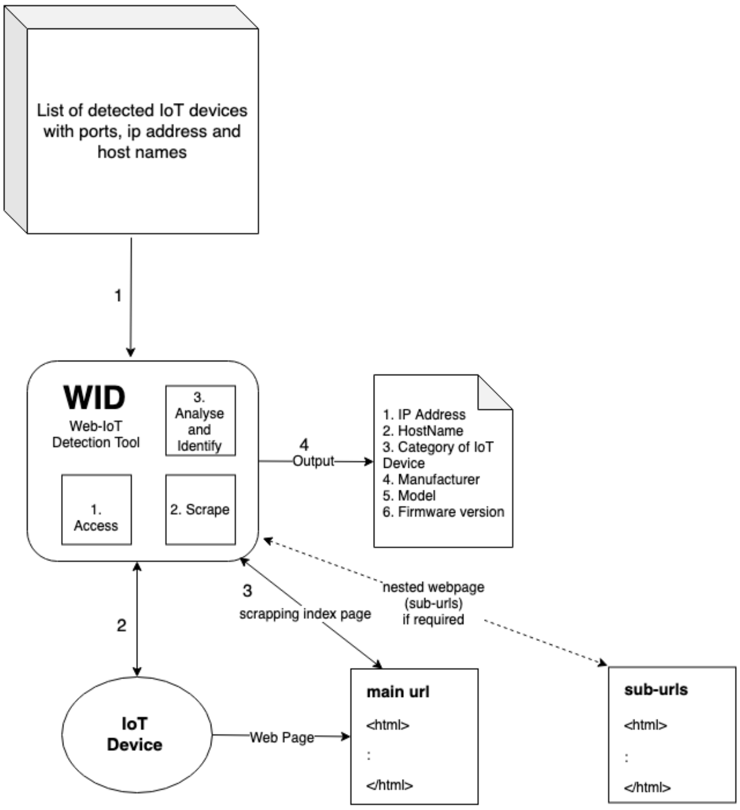

Figure 3. Overview of Web-IoT Detection (WID) Tool

```
sharad:iot_html_analysis SharadAggrawal$ python device_recog.py --ip <ip address>
Matrox Device Found
Firmware: 2.2.0.0008
Model: Monarch HD

classifiers:

<title>
        <device name>
</title>
<span id="ctl00_MainContent_DeviceNameLabel"> <device name> </span>
<span class="MatroxHD">
<img alt="Matrox MonarchHD" src="images/MonarchHDLogoSmall.png" style="float: right"/></span>
http:// <ip address> /Monarch/About.aspx
<span id="ctl00_MainContent_FirmwareRevisionLabel">2.2.0.0008</span>
sharad:iot_html_analysis SharadAggrawal$
```

Figure 4. Sample Working of Web-IoT Detection (WID) Tool

2.2. *Vulnerability Assessment*

The Internet of Things being a very new technology; there is no specially designed vulnerability assessment tool that is known to us. There are some general tools like Nessus [19], OpenVAS [20] and others but do not deliver good results as they do for regular clients or servers. So, we started with the network and web interface of all the IoT devices since it is the primary interface through which the users can connect to the devices. We started looking for web interface injections and attacks to check for the devices. The first step was to use the top OWASP IoT Vulnerabilities [21] for investigating the IoT device's web interface and tried to get administrative access. Next aim was to find the network side vulnerabilities by scanning the devices and checking the open and filtered ports vulnerable to attack - some of them being secure shell (SSH) [22], Telnet [23], SIP [24], RTSP [25], JetDirect [26] - and get administrative access into the configurations of the devices. We used software like PRET [27] and Routerscan [28]. We also modified three available exploits from Google Hacking Database [29] to find vulnerable IoT devices on the network.

3. Results

3.1. *Evaluation of tools*

In this section, we elaborate on the results of the Web-IoT Detection (WID) tool. As mentioned in Table 1, we show the models and manufacturers names that we are successfully able to identify with our tool. It shows eleven out of 20 categories of devices that have a web interface and all of them were identified. We achieve an accuracy of 92.45% as only 49 out of 53 models were identified by the software using the six classifiers. The "NetScanIoT" and the "Web-IoT Detection" tools can be used in any organization or even by an individual to detect, identify and use the output information to keep their IoT devices secure.

3.2. *Vulnerability Assessment Results*

After doing the manual security assessment, we found out that 100 out of 900 devices have the default configuration and we classify this as "Out of the box configured" category. The devices of this category had no authentication setup on the web user interface or the command-line interface. The "Easily vulnerable" category consisted of 118 devices that had easily guessable or standard manufacturer configured credentials, which made them easily accessible to users within the network. Apart from this, there were certain devices like thermometers which had a hard-coded super admin password that cannot be changed. We also found 16 devices vulnerable to known exploits, which included Real Time Streaming Protocol (RTSP) Bypass authentication. This exploit affects two manufacturers' Close Circuit Television Cameras (CCTV) with various running firmware versions. There is more than one model prone to this exploit. Using the PRET software and the JetDirect port, we were able to access the configuration of printers installed at CERN. Apart from this, we also discovered that we are able to change the standard welcome message on most of the printers.

Once the vulnerability assessment was completed, we wanted to mitigate the vulnerabilities by reporting them to the administrators and users of the devices. Sending emails to each and every affected device administrator and user at CERN was a tedious task, so we used a platform called Fast Incidence Response (FIR) [30] modified at CERN according to our needs. FIR is a centralized platform and is used to report devices to the owners and responsible users. We used this to report about the affected devices to the responsible owners and provided them with more information on how to mitigate the issues. By doing this assessment, we raise security awareness at CERN. Adding all devices from the categories, as mentioned in Figure 1. b, there were 234 vulnerable devices, which were reported and suggested solutions to mitigate them as that could have caused security issues.

Table 1. Devices Detected by WID

Category	Model	Manufacturer	Quantity
Matrox	Matrox Monarch HD	Matrox	26
Matrox	Matrox LCS	Matrox	2
Telepresence	SX20	Cisco	23
Telepresence	C20/40	Cisco	10
Oscilloscope	Tektronix	Tektronix	3
Oscilloscope	Lecroy	Teledyne Lecroy	3
Oscilloscope	Keysight53230A	Keysight Technologies	3
IP Phone	Polycom	Polycom	2
IP Phone	Cisco	Cisco	2
IP Phone	FLX	Revolabs	1
IP Phone	Yealink	Yealink	1
NAS	Diskstation	Synology	24
Printer	Color Laserjet M553	Hewlett Packard	5
Printer	Laserjet 500 color	Hewlett Packard	4
Printer	Color Laserjet m750	Hewlett Packard	2
Printer	Laserjet 2430	Hewlett Packard	3
Printer	3130cn	Dell	2
Printer	DCP-L	Brothers	2
Printer	HL-5470	Brothers	1
Printer	HL-3070CW	Brothers	1
Printer	mfc-8370dn	Brothers	1
Printer	Color Laserjet mfp m277	Hewlett Packard	3
Printer	Laserjet cp1525N	Hewlett Packard	1
Printer	Color Laserjet cm1312nfi mfp	Hewlett Packard	1
Printer	Laserjet 400 m401	Hewlett Packard	1
Printer	Star Asura	Star POS Printing Soln.	3
Printer	HP envy	Hewlett Packard	3
Printer	Photosmart plus printer	Hewlett Packard	2
Printer	Designjet T120	Hewlett Packard	2
Printer	Epson wf-3720 series	Epson	1
Printer	zebra zbr3878142	Zebra	1
Printer	sws / syncthru	Samsung	2
Printer	Officejet pro l7700	Hewlett Packard	1
Infoscreens	GM F420SEA F470S/GM F420S	JVC	9
CCTV Camera	cc8370	Vivotek	11
CCTV Camera	ip8365eh	Vivotek	9
CCTV Camera	Flexidome ip corner 9000mp	Bosch	6
CCTV Camera	M1114	Axis	2
CCTV Camera	q6000-e	Axis	3
CCTV Camera	P5635-E MKII	Axis	4
CCTV Camera	Q24	Mobotix AG	3
CCTV Camera	M24	Mobotix AG	2
CCTV Camera	M25	Mobotix AG	6
CCTV Camera	DCS-910	D-link	2
CCTV Camera	AW-HE60H	Panasonic	2
CCTV Camera	SNC-RZ50	Sony	1
PLC	Saia	SBC	10
Arduino	Arduino Yun/Uno	Arduino	9
IPMI	ILO	Hewlett Packard	12

4. Discussion

We present our results to identify and assess IoT devices on a large-scale and heterogeneous network. The tools developed by us can be used by any organization or an individual, to detect and identify IoT devices. The information provided by the tools can be used to secure their IoT devices. With our NetScanIoT software, a total of 20 categories of IoT devices were identified successfully. After identifying these devices, we performed a manual vulnerability assessment on them. This assessment showed that IoT manufacturers did not secure their devices and, moreover, certain devices like the thermometers [31] did not even allow the user to change the credentials at all. The Web-IoT Detection (WID) tool was able to identify eleven out of 20 categories of IoT devices consisting of 49 various models, manufactured by 26 different vendors from across the world. We also identified the corresponding manufacturer and firmware version for these 49 device models of IoT devices which can be used for risk identification, associated with these firmware versions. None of these identified devices are CERN-specific nor specially manufactured for CERN. They are the same devices that the manufacturers sell in the global market.

One of the significant findings was that 118 devices administered by 90 users were using default passwords and old firmware versions. The administrators did not consider to change them at all as they were not made aware by any kind of prompt that they should change the default password or update to the latest firmware version of the device. Therefore, we propose periodic scans on all networks to detect devices that might be vulnerable.

We showed that the approach is effective on a large-scale network with a larger dataset compared to similar studies. Moreover, no other work was able to classify this amount of heterogeneous IoT device models by using the web interface. For future work, we plan to identify new types of IoT devices that come up together with industrial IoT devices on our accelerator complex testbed.

Author Contributions: conceptualization, S.A. and P.O.; methodology, S.A., P.O.; software, S.A.; validation, S.A., P.O. and S.L.; formal analysis, S.A.; investigation, S.A.; resources, S.A.; data curation, S.A.; writing–original draft preparation, S.A.; writing–review and editing, S.A., P.O, S.L.; visualization, S.A.; supervision, P.O., S.L.; project administration, S.L.

Funding: This research received no external funding.

Acknowledgments: We thank Hannah Short (CERN) for proofreading that greatly improved the manuscript. This research was supported by European Organization for Nuclear Research (CERN) and CERN Openlab. We thank our colleagues from CERN who provided insights and expertise that greatly assisted the research.

Conflicts of Interest: The authors declare no conflict of interest.The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
WID	Web-IoT Detection
CERN	European Organization for Nuclear Research
LHC	Large Hadron Collider
DUT	Device Under Test
PLC	Programmable Logic Controller
ENISA	European Union Agency for Network and Information Security
NIS	Network Information Security
UI	User Interface
NAS	Network Attached Storage
MLC	Media Layer Controller
IP	Internet Protocol
MAC	Media Access Control
HTTP	Hypertext Transfer Protocol
SSH	Secure Shell
CCTV	Close Circuit Television
OWASP	Open Web Application Security Project
SIP	Session Initiation Protocol
RTSP	Real Time Streaming Protocol
URL	Uniform Resource Locator

References

1. Statista.Com. 2016. Stats and Analysis. Available at: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide (accessed on 20 June 2018)
2. Bruce Schneier. 2014. The Internet of Things Is Wildly Insecure - And Often Unpatchable. Available at: https://www.wired.com/2014/01/theres-no-good-way-to-patch-the-internet-of-things-and-thats-a-huge-problem (accessed on 18 Aug 2018)
3. Tianlong Yu, Vyas Sekar, Srinivasan Seshan, Yuvraj Agarwal, and Chenren Xu. 2015. Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In Proceedings of the 14th ACM Workshop on Hot Topics in Networks. ACM, 5.
4. Bruce Schneier. 2013. Will giving the internet eyes and ears mean the end of privacy? Available at: https://www.theguardian.com/technology/2013/may/16/internet-of-things-privacy-google (accessed on 18 Aug 2018)
5. Elżbieta Andrukiewicz, Scott Cadzow January 17, 2019. IoT Security Standards Gap Analysis. Available at: https://www.enisa.europa.eu/publications/iot-security-standards-gap-analysis DOI:https://doi.org/10.2824/713380
6. Costin, A., Zarras, A. and Francillon, A., 2017, May. Towards automated classification of firmware images and identification of embedded devices. In IFIP International Conference on ICT Systems Security and Privacy Protection (pp. 233-247). Springer, Cham.
7. Gordon Fyodor Lyon. 2009. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure, , USA.
8. Stenberg, Daniel. "Everything-curl." GitBook (2017).
9. Zheng, Z., Webb, A., Reddy, A.N. and Bettati, R., 2018, July. IoTAegis: A Scalable Framework to Secure the Internet of Things. In 2018 27th International Conference on Computer Communication and Networks (ICCCN) (pp. 1-9). IEEE.
10. Hang Guo and John Heidemann. 2018. IP-Based IoT Device Detection. In Proceedings of the 2018 Workshop on IoT Security and Privacy (IoT S&P '18). ACM, New York, NY, USA, 36-42. DOI:https://doi.org/10.1145/3229565.3229572
11. Siby, S., Maiti, R. R., and Tippenhauer, N. O. IoTscanner: Detecting privacy threats in IoT neighborhoods. In Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security (New York, NY, USA, 2017), IoTPTS '17, ACM, pp. 23–30.
12. Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org (accessed on 7 Aug 2019)
13. Steve Deering. ICMP Router Discovery Messages, September 1991. RFC, 1256.
14. John Toebes, Bryan C Turner, and Douglas Jay Walker. 2009. Arrangement in a server for providing dynamic domain name system services for each received request. USPatent7,499,998.
15. Free Software Foundation. GNU Wget 1.20. Available at https://www.gnu.org/software/wget/ (accessed on 7 Aug 2019)
16. Scarpy Developers. Scrapy 1.7. Available at https://docs.scrapy.org/en/latest/intro/overview.html (accessed on 7 Aug 2019)
17. Python Software Foundation. Selenium 3.141. Available at https://pypi.org/project/selenium/ (accessed on 7 Aug 2019)
18. Richardson, Leonard. "Beautiful soup documentation." April (2007).
19. Tenable.Com. 2015. Nessus. Available at: https://www.tenable.com/products/nessus/nessus- professional (accessed on 12 Mar 2018)
20. Openvas.Org. 2012. OpenVAS. Available at: http://openvas.org/ (accessed on 16 Feb 2018)
21. OWASP. 2015. OWASP Internet of Things Project. Available at: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Vulnerabilitiessoftware.html (accessed on 7 May 2018)
22. Tatu Ylonen and Chris Lonvick. 2005. The secure shell (SSH) protocol architecture. RFC, 4251.
23. Jon Postel and Joyce K Reynolds. 1983. Telnet protocol specification. RFC, 854.
24. J. Rosenberg , H. Schulzrinne , G. Camarillo , A. Johnston , J. Peterson , R. Sparks , M. Handley , E. Schooler, SIP: Session Initiation Protocol, RFC Editor, 2002. RFC, 3261.

25. Schulzrinne, H., Rao, A., and Lanphier, R. (1998). Real Time Streaming Protocol (RTSP). RFC, 2326, 1-92.
26. The free encyclopedia Wikipedia. 2013. JetDirect. Available at: https://en.wikipedia.org/wiki/JetDirect (accessed on 13 Jun 2017)
27. für die Master-Prüfung, S.P., des Studiengangs, I.T., und Systeme, N., Müller, J., Schwenk, J., Somorovsky, J. and Mladenov, V., 2016. Exploiting Network Printers.
28. Stas'M Corp. 2018. RouterScan. Available at: http://stascorp. com/load/1- 1- 0- 56 (accessed on 10 May 2018)
29. Google Hacking Database. 2016. Google Hacking Database. Available at: https://www.exploit-db.com/google-hacking-database/ (accessed on 7 May 2018)
30. CERN CERT. 2018. FIR. Available at: https://github.com/ CERN- CERT/FIR (accessed on 25 Jun 2018)
31. Agarwal S., Oser P., Short H., Lueders S. (2017) Internet of Things security. Available from: https://doi.org/10.5281/zenodo.1035034

Sample Availability: The dataset and the source code sample will be available from the authors Sharad Agarwal and Pascal Oser after the completion of the PhD of Pascal Oser, in 2020.