Where do all the idIoTs come from?

Identification of Insecurely Developed IoT devices and a corresponding analysis of Dutch digital markets that sell them

Thesis submitted to the Delft University of Technology in partial fulfillment
of the requirements for the degree of

Master of Science in Management of Technology

by

Swaathi Vetrivel
(4900863)

August 2020

Graduation committee

Chairperson: Prof.dr.M.J.G.(Michel) van Eeten, Section Organization & Governance (TBM)
First Supervisor: Dr.ir.C.(Carlos) Hernandez Ganan, Section Organization & Governance (TBM)
Second Supervisor: Dr.S.T.H.(Servaas) Storm, Economics of Technology and Innovation (TBM)
Advisor: E.R. Turcios Rodriguez, Section Organization & Governance (TBM)

EXECUTIVE SUMMARY

Internet-of-Things (IOT) is a generic term used to describe the growing trend of everyday things like fridges, toothbrushes and bulbs being connected to the internet. IoT devices have sensors, software and other technologies to collect, transfer and exchange data over the internet thus allowing these devices to interact with each other and with their users. The realtime data collected by IoT devices is processed by machine learning algorithms to improve monitoring and surveillance, build better and more personalised features and more accurate prediction models. However, although the increased connectivity provided by IoT improves functionality, efficiency and provides convenience, it also causes new threats and risks since they expose new attack vectors and surfaces. One significant risk posed by IoT devices is the lack of proper access control, there is widespread use of default user credentials, which is leveraged by hackers to gain access to and infect these devices to create botnets, like the 'Mirai' botnet. These botnets are in turn used to malicious activities like launching DDoS attacks, for instance, in 2016, the Mirai botnet launched a massive DDOS attack that caused most of the US West Coast to lose access to several high-profile websites like GitHub, Twitter, Reddit, Netflix, among others.

The attacks caused by Mirai and its variants imposes a negative externality on a third party that is neither the buyer nor the seller. While both consumers and manufacturers of IoT devices can take steps to mitigate this externality but they lack sufficient incentives to do so. At an individual level this behaviour is rational since individual actors in the market act in their own self interest and pursue their individual incentives to maximise their utility. However, collectively these decisions fail to promote common welfare and impose significant social and economic costs on society and point to a market failure. Although some manufacturers took the decision to improve their access control in response to the Mirai botnet, not all did and consequently, there are still many insecurely developed IoT (idIoT) devices sold in the market. However, currently, there is scarce data available on these manufacturers with poor security practices and the retail channels that sell these idIoT devices. Further, there is no empirical research on the extent of information asymmetry and level of transparency for security related information in the market for IoT devices. This research fills these gaps through empirically analysing the market for IoT devices within the Netherlands.

The main objective of this thesis was to identify the manufacturers and online retailers of idIoT devices and thereby empirically analyse the characteristics of the digital market for IoT devices within the Netherlands and examine the behaviour of the actors involved. The main research question was *"How do insecure IoT devices enter the Dutch consumer market, how do retailers present them to consumers and how do consumers evaluate their security?"*

In the first part of the research, in order to identify idIoT devices, IP addresses of Mirai infected devices was obtained from a /15 darknet. From these IP addresses, the corresponding device type and manufacturer was identified through using landing page screenshots, banners and HTML title and header fields gathered from port scans. The results indicate that image based identification is significantly more effective in identifying and labelling IoT devices while the HTML title and header fields are the least useful. The banners were more useful to determine non-IoT devices, however this result could be improved through collection of banners of additional protocols.

In the second part of the research, using the manufacturer and device type of infected IoT devices thus gathered, the e-commerce websites that sell these devices

within the Netherlands were identified through automated google search. From these identified websites, product description data, average ratings, consumer feedback and other fields were scrapped for the infected devices and for other popular IoT devices in the same device type category. To determine if there were any statistically significant differences between these two groups, an independent sample T-test was performed on average ratings, price, total number of reviews, average consumer sentiment (calculated through sentiment analysis) and the number of vulnerabilities associated with the manufacturer (taken from the CVE details database). The results were statistically significant only for average ratings and number of vulnerabilities, both of which were higher for infected devices. The higher average rating for infected device points to lack of transparency of security features in the market for IoT devices. In order to analyse if there was any security related information communicated to consumers in these ecommerce channels, topic modelling was performed and the results indicate that manufacturers market features like power consumption, storage, quality and so on, but not security. Similarly, in order to understand if consumers exhibit any security related concerns, topic modelling was run on the customer reviews. The results did not have any security related concerns and, based on the results, consumers value connectivity, easy installation and performance among others.

Finally, a SVM classification model was built to predict whether a given device is infected or not based on the market data collected, the model had an accuracy of 94.5%. More significantly, the coefficients of SVM model allowed for determining the relative weightages of different attributes of the three actors in the market for IoT devices - the consumer, manufacturer and the ecommerce channel - in determining whether the device belongs to the infected class or the non-infected class. Further, the direction or sign of the coefficient allowed for understanding the predicted class, if the coefficient was postive it belongs to the Mirai infected class and vice versa. The ghest weightage in the model comes from the manufacturer attributes, followed by the intermediary websites while consumer attributes have the least weightage. Within the manufacturer attributes the country where the headquarters is located had the highest weightage.

These findings answered the main research question and the most significant recommendation to improve the security of IoT devices following from the results is to provide security related information to consumers in the ecommerce channels. If security related information is made available to consumers at the time of purchase, they would be able to make more secure choices. However, in order to do so, security information needs to be available and updated by the manufacturers. Since manufacturers currently do not have any incentives to do so, policy intervention is needed to mandate exposure of security information about IoT devices.

ACKNOWLEDGEMENTS

This thesis has been a wonderful learning experience, and I am grateful to have had the opportunity to work and explore such an interesting and relevant field of research. However, I could not have done tackled the challenges of this research alone and I would like to first express my gratitude to my parents, my strongest support system and to whom I owe everything I am.

Next, I would like to sincerely thank Elsa for advertising the research, giving me access to the data and for being generous with her time, help and support throughout the course of the thesis. Thanks Elsa for teaching the ropes of research, for providing valuable writing tips and for reminding me that research requires progress and not perfection.

I am deeply grateful to have had such an ncredibly accomplished committee members - Prof. van Eeten, Prof. Storm and Prof. Ganan, all of whom were kind and supportive throughout this jounrey and helped me improve the thesis and my research skills - due to your critical inputs, advice and suggestions, the thesis has reached a better shape and I have become a better researcher, thank you!

Finally, I would like to thank my friends, old and new, from India and Delft for the many conversations, dinners, walks and breaks that helped keep me going through the highs and lows of the last few months.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 BACKGROUND

1.1.1 Internet of Things

Fridges that send alerts when milk runs low, toothbrushes that provide personalized feedback on brushing techniques, bulbs that can be switched on or off through a mobile app, all of these represent the growing trend of everyday things being connected to the internet. These and other such devices are called the Internet of Things (IoT) and they have undeniably revolutionized the way we live and interact with our physical environment. IoT devices have sensors, software and other technologies to collect, transfer and exchange data over the internet thus allowing these devices to interact with each other and with their users. A 2019 European Commission report on IoT states that IoT represents the next step towards the digitisation of our society and economy, where objects and people are interconnected through communication networks and report on their status and/or the surrounding environment (sha, 2019). The worldwide number of IoT devices is projected to increase to 43 billion by 2023, an almost threefold increase from 2018 and their population is expected to reach 125 billion within the next decade (Anstee, 2019).

Formally, the International Telecommunication Union (ITU) defines IoT as *"a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving inter-operable information and communication technologies"* (ITU, 2012). A more comprehensive definition for IoT, offered by the European Research Cluster on the Internet of Things (IERC), is *"A dynamic global network infrastructure with self-configuring capabilities based on standard and inter-operable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network"*(Ganegedara, 2019).

The growth of IoT devices is closely coupled with the advent of Web 3.0, the third generation of internet services, which focuses on Artificial Intelligence and machine based understanding of data to provide a data-driven and semantic web. The vast volume of data collected by IoT devices in real time can thus be processed by machine learning algorithms to improve monitoring and surveillance, build better and more personalised features and more accurate prediction models. Since almost any human or machine enabled activity can be enhanced by such real time data collection and analysis, IoT devices pervasive both in the consumer and industrial sectors. Industrial IoT devices are used across various industries to increase operational efficiencies and optimise performance through connectivity, automation and data analytics, thus providing opportunities for progression and transformation of manufacturing industries (Rajput and Singh, 2019). These advancements have paved the way for Industry 4.0, the fourth industrial revolution with the characteristics of cyber physical systems (CPS) production, based on heterogeneous data and knowledge integration (Lu, 2017). Further, the market for IoT is predicted to grow considerably alongside the still nascent 5G technology, encouraging many businesses to invest in development and deployment of IoT solutions. In addition, various governments around the world have also launched smart city initiatives that leverage the connectivity and functionalities provided by IoT devices to improve the quality of life and to allow for more sustainable use of limited natural resources.

1.1.2 Security of IoT devices

While the increased connectivity provided by IoT improves functionality, efficiency and provides convenience, it also causes new threats and risks since they expose new attack vectors and surfaces. These risks include but are not limited to unauthorized access and misuse of sensitive data, vulnerable IoT devices being used to launch attacks on other systems and substantial risks to personal safety due to disruption of critical services (FTC, 2015). In addition to lacking adequate security controls, most of these devices cannot be updated and therefore they remain vulnerable to even known security flaws. As more and more IoT devices monitor and manage our cyber physical reality, attacks on these devices can have economic, energetic and physical security consequences that are more severe than the traditional Internet's lack of security, and way beyond the threats posed by attacks to mobile telephony (ser, 2020).

These risks are further aggravated by the poor security in most consumer IoT devices (Munro, 2018). In particular, the lack of strong access control in IoT devices allows attackers to gain access to and infect these devices thereby taking control of them and using them to create botnets. A botnet is a logical collection of internet-connected compromised devices known as 'bots'. These botnets are typically used to launch distributed denial-of-service (DDoS) attacks to overwhelm a target web service or Internet infrastructure with malicious traffic, such that it is incapable of processing legitimate requests (Bhardwaj et al., 2018). In addition, these botnets have also been used to mine cryptocurrencies, for industrial espionage and to steal online banking credentials (Kleinhans, 2018).

In particular, the widespread use of default passwords in IoT devices is leveraged by the botnet 'Mirai' to find, infect and gain control of these insecure devices through random scans of the IP address space. Once a device is thus infected, it is reported to a Command and Control server thereby enabling it to be used in large scale botnets (De Donno et al., 2017). In October 2016, the Mirai botnet launched one of the largest DDoS attacks ever seen on the internet - an attack that reached a magnitude of about 1.2 Terabits per second and left most of the US West Coast incapable of accessing several high-profile websites like GitHub, Twitter, Reddit, Netflix, Airbnb and many others. Since the source code of Mirai was made public in late 2016, it has spawned many variants and quickly emerged as a high-profile security threat. Even amidst the ongoing Covid-19 crisis, Mirai variants continue to emerge - a recent Mirai variant file was detected in April 2020 and is called Covid (Krebs, 2020). Mirai marks a change in the evolution of botnets; the simplicity through which devices were infected and its precipitous growth, demonstrate that novice malicious techniques can compromise enough low-end devices to threaten even some of the best-defended targets (Antonakakis et al., 2017). Large companies invest in security measures to protect themselves against DDoS attacks and have mitiagtion plans in place to prevent the severity of loss from such attacks but such measures can do little in preventing such powerful and voluminous attacks.

DDoS attacks and other problems caused by botnets are a special case of security risks posed by IoT devices since these impose a negative externality to a third party that is neither the buyer nor the seller. When the party that is responsible for the protection of the system does not suffer the costs and consequences of a security failure, then problems arises as in the case of attacks caused by botnets. While both buyers and sellers of IoT devices can take steps to mitigate this externality, they lack sufficient incentives to do so. Individual consumers and manufacturers might invest in better security to protect themselves from attacks but if the device vulnerabilities are used to attack other targets then they are unlikely to invest time, effort or money on fixing the vulnerabilities. This scenario thus represents a market failure and shows that cybersecurity is not merely a technical problem, it has important economic and behavioral dimensions associated with it (Anderson and Moore, 2006).

1.2 THE MARKET FOR IOT DEVICES

An analysis of the market for IoT devices would help in better understanding this market failure and bring to light the various market forces that shape the incentives of the actors involved - the manufacturers that produce the devices, the consumers that buy the devices and the retailers that facilitate the transaction.

Manufacturers

Despite the severe consequences of insecure IoT devices, there is a discernible lack of prioritisation of security of IoT devices by IoT device manufacturers which is influenced by a variety of factors, some of which are listed below (ENISA, 2016).

- Limited device resources
- High complexity of the IoT ecosystem due to the diverse devices, communications, interfaces
- Fragmentation of IoT related standards and protocols
- Pressure on companies to be the first-to-market
- Cost considerations
- Lack of IoT cybersecurity expertise
- Absence of a user interface to directly access IoT devices
- Higher emphasis on functionality and usablity than on security

Thus, manufacturers lack sufficient incentives to address security features since doing so would increase production costs, reduce battery life and delay the time to market (Brass et al., 2017). Moreover, there are different players involved in making these devices, hardware manufacturers, platform suppliers and IoT integrators and there is an inclination to assume that somebody else in the supply chain might have addressed the security concerns (Munro, 2018). Furthermore, consumers are unaware of security aspects of IoT devices which causes scarce market demand for secure IoT products and further reinforces manufacturer's apathy towards security considerations (Storey, 2014).

Consumers

Manufacturers might be incentivised to factor security considerations when consumers demand secure products. However, currently consumers do not prioritise security and instead, reward an early market entry, new functional features, interoperability and availability of complementary goods that are compatible with their existing products (Morgner et al., 2018). Moreover, even if consumers have security concerns, due to the information asymmetry in the market for IoT devices, they are unable to assess the level of security of IoT products (Morgner et al., 2018). This has been described in economic theory as the 'market for lemons', consumers are not willing to pay for something they cannot measure (Akerlof, 1978). Further, although consumers routinely reject security advice, for instance they fail to change the default passwords on their devices, this behaviour is deemed rational from an economic point of view. Following security advice saves them from the direct costs of an attack but it increases their indirect costs, and since these indirect costs are higher relative to the direct costs, not following the advice serves to optimise consumers' utility function (Herley, 2009).

Retailers

Apart from manufacturers and consumers of IoT devices, a key player in the IoT ecosystem are the retailers that act as an intermediary between them and facilitate the sale of these products. Traditionally, these retailers were brick and mortar stores that usually held inventory of the goods sold. However, with the growing prevalence of online shopping, these intermediaries are increasingly ecommerce portals, like amazon.com, bol.com and so on, that provide an online marketplace that connects sellers to buyers. Within EU, the ecommerce market is expected to reach 717 billion euros by the end of 2020, an increase of 12.7% from 2019 (News, 2020). Unlike traditional stores, digital ecommerce markets only facilitate transactions and do not take title to the products sold. Further, these digital markets reduce the search costs for consumers making it easier to find low cost products, which in turn promotes price competition amongst manufacturers (Bakos, 2001).

1.3 PROBLEM STATEMENT

The negative externality imposed by insecurely developed IoT (idIoT) devices with poor access control can be fixed by simple measures, consumers can change the passwords on their devices and manufacturers can generate unique default passwords for their devices. To be fair, some manufacturers took the decision to improve their access control in response to the Mirai botnet however not all did and consequently, there are still many idIoTs sold in the market (Voolf and Cohen, 2020). At an individual level the failure to take these steps to improve access control is rational since individual actors in the market act in their own self interest and pursue their individual incentives. However, collectively these decisions fail to promote common welfare and impose significant social and economic costs on society and point to a market failure. In order to fix this market failure and better manage the security of IoT devices, external policy intervention is needed.

Within the EU, the Cybersecurity Act establishes a cybersecurity certification framework for all ICT products - including IoT devices - which addresses this market failure and provides a means to communicate security features of a product to consumers through certification schemes. The framework has three levels of assurance basic, substantial and high, which indicate the risk associated with the use of the product in terms of probability and impact of an incident. This would make transparent to consumers the security characteristics of the IoT devices and ensure manufacturers adhere to specified security standards thereby increasing trust and security in products and services that are crucial for the Digital Single Market (Comission, 2020). However, the certification imposes a cost on the manufacturers with an associated learning curve and since it is voluntary, not all manufacturers might be inclined to comply. Moreover, manufacturers that fail to implement secure access control in their devices are unlikely to undertake the costs required to build IoT devices that meet the certification standards of security - at least not until the revenue loss from the lack of certification is higher than the cost of getting the certification. And, in the meantime these insecure devices will continue to proliferate the market, posing security risks and imposing severe social and economic costs.

Considering these huge costs, urgent action is needed to address the insecurities of IoT devices and improve the transparency of security features of IoT devices. However, currently there is scarce data available on manufacturers with poor security practices and the retail channels that sell these idIoTs. Further, there is no empirical research on the extent of information asymmetry and level of transparency in the market for IoT devices. This gives us an opportunity to fill these gaps through empirically analysing the market for IoT devices within the Netherlands.

1.4 RESEARCH OBJECTIVE AND QUESTIONS

The objective of this research is to find opportunities for improving the security of IoT devices through identifying the manufacturers and online retailers of insecure IoT devices and thereby empirically analysing the characteristics of the digital market for IoT devices within the Netherlands and examining the behaviour of the actors involved. The main research question is 'How do insecure IoT devices enter the Dutch consumer market, how do retailers present them to consumers and how do consumers evaluate their security?' In order to identify insecure IoT devices, data on devices infected with Mirai will be used. Thus, to achieve the research objective, and answer the main research question, the following sub research questions need to be answered.

- Which IoT devices in the Netherlands are commonly infected with Mirai and who is the manufacturer of these infected IoT devices?
- Through which retail channels do these insecure IoT devices enter the Dutch Consumer market?
- How do manufacturers characterize and present information about the security features of their products in these retail channels?
- How do customer reviews reflect the security concerns of the users of IoT devices?

1.5 RESEARCH APPROACH

This thesis is rooted in empirical research and quantitative data collection, and has two parts. In the first part, the manufacturers of Mirai infected IoT devices within the Netherlands will be identified. In the second part, the ecommerce channels that sell these IoT devices will be traced. The results will be interpreted and analysed to answer the third and fourth research questions and gain empirical insights of information asymmetry and market transparency.

1.6 SCIENTIFIC, PRACTICAL AND MANAGERIAL RELEVANCE

1.6.1 Practical Relevance

The answers to SQ1 and SQ2 will serve as useful input to policymakers to design targeted interventions in these channels to promote a safer IoT ecosystem.

1.6.2 Scientific Relevance

This study contributes to the growing field of Economics of Information Security that studies factors that actors perceive as relevant for security decisions (incentives), their influence on economic actions of individuals and organizations and how these actions lead to emergent properties of the system (Asghari et al., 2016).

1.6.3 Managerial Relevance

Most companies market IoT devices based on the product features and price competition with little weightage to security considerations. Understanding consumer perceptions of IoT devices will help companies better market their products; if there is evidence that consumers value security features it will help companies align their product development and marketing strategy appropriately.

1.7 THESIS STRUCTURE

The second chapter consists of a literature review of methods to identify device type and manufacturer from an IP address and a brief overview of various suggestions proposed to improve security of IoT devices. The third chapter details the methodology followed to answer the research questions, fourth chapter presents the findings and an interpretation of the results and finally, the fifth chapter concludes with a discussion followed by the limitations of the research and recommendations for future research.

2 LITERATURE REVIEW

This section summarizes the findings from the literature review that was carried out with two objectives. First, to identify the existing techniques for device and manufacturer identification in order to answer the first subresearch question and second, to get an overview of the methods proposed to encourage manufacturers to build more secure devices. Further studies that propose methods to nudges consumers into making more secure choices are also outlined.

2.1 MANUFACTURER IDENTIFICATION

The anonymity provided by the internet acts as a double-edged sword, affording privacy for individuals while also complicating the task of identifying the source of a malicious activity. Simply put, by nature of the design of computer networks, there is no straightforward method to glean data on the manufacturer or device type of an IoT device, or of any device on the internet. Nonetheless, the literature does provide mechanisms that can help with identification with varying degrees of accuracy and efficiency.

One method for identification, employed by Meidan et al. (2017) is the application of machine learning algorithms on network traffic data to identify IoT devices within the network. This method leverages the characteristics of the network traffic generated by an IoT device and subsequently classifies the device by make and model with an accuracy of 99.28%. It uses supervised learning and the classifier is trained to distinguish between IoT and non-IoT devices and to associate each IoT device with a particular device class. However, by design, owing to the use of supervised machine learning techniques, the algorithm can only identify devices that were present in its training data and cannot identify new devices on the network.

In contrast, the method illustrated by Miettinen et al. (2017) as part of their security system 'IoT Sentinel', also identifies new devices within the network. Although IoT Sentinel is aimed at effectively mitigating attacks within a network or, failing that, limiting their impact by restricting communications within a network, in order to do so, it employs machine learning to classify devices based on a combination of model and software version of the device. IoT Sentinel generates device-specific fingerprints based on its MAC address and the distinguishable pattern of device-specific communication behaviour during the initial setup process. These fingerprints are then used to classify devices based on the device-type, and its global accuracy with 27 devices was 81.5%. Although 17 devices were identified with an accuracy of 95%, the accuracy for the remaining ten devices was only 50% since these contained different devices from the same vendor intended for the same purpose, two models of smart plugs from TP-link for instance. However, devices from the same vendor but with different purposes like D-Link camera and hubs were accurately distinguished. Nevertheless, since their network consists of only 27 devices, it is tricky to judge IoT Sentinels' performance in larger networks with a wider range of devices.

Conversely, the approach taken by Kumar et al. (2019a) used data collected from about 83M IoT devices spread across 16M real world homes. Such large amounts of data was obtained from user-initiated scans made through WiFi inspector, a tool by Avast, that is included in all of Avast's anti-virus products. The WiFi inspector

runs locally on the user's personal computer and performs network scans of the local subnet to identify devices with remotely exploitable vulnerabilities or with weak credentials. However, to generate a user friendly list of hosts in the network, the WiFi inspector combines a set of expert rules and a supervised classification algorithm that run on the application and transport layer data from the scan and categorizes devices based on the device type. The classification algorithm uses model information available in the web application interfaces or banners, the details broadcasted through UPnP and mDNS and, in cases where manufacturers follow an informal labelling approach, a set of regular expressions that parse out the relevant fields. To obtain the manufacturer information, the WiFi inspector looks up the first 24 bits of the MAC address in the public IEEE Organizationally Unique Identifier (OUI) registry. A problem with this technique however is that the MAC address could be associated with the vendor the network interface card instead of the manufacturer, this was overcome by manually resolving these cases.

In a similar vein, Martin et al. (2016) use the lower order bytes of the MAC address to extract device and model information in addition to identifying the manufacturer. Through analysing data from over two billion 802.11 frames, the protocol used for WLAN (Wireless Local Area) networks, they were able to extract device and model information through data leaked by management frames and discovery protocols. They also distinguish the population and general density of the device, the policies of vendor allocation and use, the exchange of OUI between manufacturers, the discovery of unique models that occur in many OUIs, and the mapping of contiguous address blocks to specific devices. From this mapping they predict finegrained device type and model for unknown devices solely on the basis of their MAC address with 81% accuracy.

Following an entirely different approach, Le et al. (2019), use text processing algorithms to analyse the DNS (Domain Name System) queries sent by IoT devices to identify the devices by vendor and type. DNS is used in any network communication to resolve domain names (host name with the domain suffix, eg., hostname.domain.com) to IP addresses, which is necessary before a device can connect to the server. They illustrate that since majority of the domain names an IoT device queries would belong to the vendor and since different IoT devices from the same vendor would query different domain names, these properties can be used to distinguish between IoT vendor and device types. They were able to achieve an accuracy of 90% for vendor identification and an impressive 99% in identifying the device types. They were also able to achieve an accuracy of 92% for identification of devices that were not present in the training data.

Unlike the more active methods discussed so far, Neshenko et al. (2019) use passive monitoring and measurement techniques to gather information about malicious activities of compromised IoT devices by investigating data traffic collected by a network telescope. Darknet or network telescope refers to a set of routable, allocated, yet unused IP addresses and hence characteristically, all traffic targeting this IP space is unsolicited (Bou-Harb et al., 2017). Using a data driven approach, they were able to locate exploited IoT devices, understand and classify the illicit actions and examine their hosting environments. While they did not identify the device or manufacturer type, they were able to identify the hosting sectors i.e manufacturing, finance, government etc., of the exploited devices. In order to distinguish IoT devices from other internet hosts, they leveraged the search engine Shodan, which is a database of internet connected devices. Shodan crawls the Internet in order to identify and index devices that are connected, storing the collected device IP addresses along with ports and service banner data in a searchable database accessible via the shodanhq.com web interface or via the Shodan API. It also provides the ability to filter using country, hostname, IP address ranges, operating system and ports (Bodenheim et al., 2014).

Similar to Shodan is Censys which is cloud-based service that maintains an up-to-date snapshot of the hosts and services running across the public IPv4 address

space, and also exposes this data through a search engine and API (Durumeric et al., 2015). To gather real time data, Censys continually scans the entire IPv4 public address space across a wide range of ports and protocols and grabs the associated banner - a text message presented on the devices that describes the service running on it. Post validation of this data, it employs a pluggable scanner framework to perform application-layer handshakes and to dissect the handshakes to subsequently create structured data about each host and protocol. The resulting data is post processed and exposed to researchers, through a public search engine, REST API, publicly accessible tables on Google BigQuery, and downloadable datasets, thus ensuring transparency. Censys also uses an extensible annotation framework that enables researchers to programmatically define additional attributes that identify device models and tag security-relevant properties of each host. This allows for public contribution to application scanners to scan additional protocols and also to annotate device types or properties.

Censys was used by Antonakakis et al. (2017) in their seven-month retrospective analysis on the emergence of the Mirai botnet, the evolution of its variants, the competition for vulnerable hosts and the devices that were affected. In order to determine the make and model of the devices infected with Mirai, Censys scans of HTTPS, FTP, SSH, Telnet, and CWMP protocols were used. Nonetheless, they articulate several challenges in accurately labelling an IoT device through this approach. First and foremost, since the Mirai botnet immediately disables common outward facing services like HTTP upon infection, it prevents Censys scans of these infected devices. Moreover, Censys scans can take more than a day to complete in many cases during which time, depending on the DHCP (Dynamic Host Configuration Protocol) configuration of the device, it might be assigned a new IP address. DHCP is a network management protocol used in IP networks to automatically assign IP addresses to devices on a network through the DHCP server. The final challenge is that Censys performs scans of different protocols on different days which makes it difficult to combine banners from multiple services which would increase the specificity of the labels. They overcame these challenges by restricting the analysis to banners that were collected within twenty minutes of performing a scan, thereby mitigating the risk of incorrectly associating banner data from an uninfected device with Mirai infections due to DHCP churn.

Similarly, Cetin et al. (2019) also used Censys to identify the device type as part of their empirical study on the cleanup of IoT malware. They collected, through Censys, raw scan data including the HTML code and banner information, for each IP address of an ISP where an infected host was detected. Their analysis focused on scans of CWMP (port 7547), FTP (port 21), HTTP (port 80 and 8080), HTTPS (port 443), SSH (port 22) and Telnet (port 23 and 2323). However, they were able to accurately label only 28% of the devices through Censys since the remaining 72% of devices lacked banners. To overcome this limitation, they conducted port scans on the unidentified devices using Network Mapper (Nmap) which allowed them to gather banner information of additional ports which are not covered by Censys - port 5000 (UPnP), 8443 (alternative HTTPS), 32400 (Plex media) and 37777 (QSee DVRs). This allowed them to label 36 additional devices.

Taking a more intrusive approach, Yu et al. (2020) identify the firmware of vulnerable devices by logging into these devices using default passwords and accessing their web management portals. Since the login pages of devices are time invariant, similar across manufacturers and contain distinctive information on type and brand of the device, these can be used to fingerprint the associated devices. They grab the content pages and use HTML scrapping techniques to construct a fingerprint that can be used to identify the device. Unsurprisingly, identification based on the internal content of web management pages allowed to achieve an accuracy of 95.97%.

Similarly, Agarwal et al. (2019) propose a method to identify vulnerable IoT devices and identify the manufacturer, device model, and the firmware version cur-

rently running on the device using the page source from the web user interface. Although this also involves gaining access to the device by presumable brute forcing through default password combinations, they were able to get an accuracy of 92.45% for IoT device identification.

Placing emphasis on frugal communication while scanning wide area networks, Tanemo et al. (2020) propose a method to limit the number of target ports that are scanned through extracting relevant information about each IoT device from the scan results. Further, they outline their approach to create an efficient dataset that allowed them to extract information about an IoT-device from a large amount of banner information.

2.2 INCENTIVES FOR SECURITY

As mentioned in the introduction, a policy appropriate to induce manufacturers into building more secure IoT devices would need to overcome the inertia caused by market forces failing to bring about a focus on security. This is elaborated by Jerkins (2017), who, after a process of identifying vulnerable devices attacked by Mirai and subsequently notifying the test bed owners of the vulnerability, concedes that there is not enough incentive for manufacturers, ISPs, or owners of IoT devices to address device insecurity since market forces value low cost and ease of deployment over security. Stressing the lack of market forces or regulatory requirements that would trigger a change from the current state of insecurity, he asserts the need for government intervention backed by a supporting legal framework and combined with a notification approach for vulnerabilities. This would, in his opinion, induce ISPs to mitigate vulnerable devices in their network, motivate manufacturers to improve their security practices and encourage individuals to consider the security of their IoT devices, to avoid the legal consequences or liability for harm caused by their devices.

However, government intervention is not without difficulties as illustrated by Brass et al. (2017), who elaborate on the challenges in implementing existing measures taken by the EU and the US to increase the security of IoT devices. These measures include promoting the principle of "security by design" for IoT manufacturers with the vision of it eventually being extended to "security by default". The first of these challenges is the difficulty in convergence to a core set of standards to support these principles given the diversity of existing and emerging standards in cybersecurity and data protection. These standards range from technical specifications for encryption at device level to cybersecurity risk management at the organisational level which increase the complexity of privacy and security standards that apply to IoT and consequently make it difficult for organizations to adhere to the princples. The second challenge is that owing to the diverse applications of IoT, the standards developed are within sectoral verticals, instead of an encompassing standard across verticals. Further, the 'light touch' regulatory approcach to IoT makes the task of ensuring compliance to a reasonable level of security difficult since these principles are non-binding. While noting that security by design has a large ambit that makes sole reliance on top down measures insuffucuent, they encourage governments to

> ...search deeper in their policy toolbox to enable the institutional capacity of private and public entities to coordinate and respond in an adaptive nmanner to rapidly evolving security and privacy challenges. Thus, governments must consider their wider "orchestration" and "mobilisation" role in order to "activate networks for public problem solving".

They suggest government led training programmes in security for providers of government contracts and small and medium size organisations who cannot afford the costs of implementing and upgrading cybersecurity measures to circumvent risks

posed by IoT. They also suggest that the government take measures to simplify information sharing between private enterprises and government agencies that work on security of interconnected cyber and physical infrastructures. In order to allow the insurance market to better assess exposure and model cybersecurity risks, they propose that governments use positive incentives to promote a wider adoption of information assurance schemes in the private sector.

Taking a more investigative approach, Angrishi (2017) analysed the vulnerabilities that were exploited by IoT botnets in major DDoS incidents and provides recommendations for manufacturers, in addition to end users and ISPs, to mitigate IoT related cyber risks. His recommendations for manufacturers include mandating a unique strong default password on the devices, hard coding the devices to enable connection only to private IPv4 addresses or to the manufacturer's website while blocking communication with all other domains and IP addresses, provision for periodic contact with manufacturer or seller's site to check for security updates and a reduced functionality if the device has not connected to the manufacturer's website for a specified duration of time. He also proposes that laws should be formed to ensure that manufacturers are responsible for monitoring and implementing safety best practices on their devices. Further, he suggests that IoT devices be certified for security by national or international regulatory bodies which could help with promoting security awareness amongst device manufacturers, vendors and end users alike.

In contrast, Storey (2014) refers to the void of responsibilities left by the lack of willingness amongst international legislative bodies to impose standards for IoT. He argues that the void created has put the onus on the manufacturers to decide which security systems are needed in their devices unless the data collected by the device falls under the jurisdiction of a specific governing body like healthcare providers. However, from a manufacturers perspective, he says, security might not be the first concern since it would add to cost of production, not create a significant increase in the product's value as perceived by customers, add a layer of complexity to the device, affect the performance and might make the user interface more difficult to navigate. He points to the first mover advantage in new technologies which push security considerations that increase the product's time to market to the backseat for manufacturers. Further, he notes that the lack of consumer awareness about security issues which reinforces manufacturer's reluctance to address security issues. He states that since more critical infrastructure like power generation plants and electricity grids are being brought online, there is an urgent need to ensure these facilities are protected against cyber or terrorist attacks. The key step, according to him, to protect such infrastructure is ensuring proper identification and authentication of all devices and the suggested method to achieve this is through use of a secure element, that cannot be copied or tampered with and which holds a cryptographic key unique to the device, within each connected device. He also notes that vendors and service providers should take the lead in deploying security solutions which would allow individual users to take control their identity and equip them in turn to take control of their personal devices.

Similarly, Morgner et al. (2019) investigate into manufacturers incentives for increased sustainable security in the development of IoT products and propose the use of mandatory security labels on devices. These labels would state the manufacturer's willingness to provide future security updates and also explicitly mention when security updates are not guaranteed. They hypothesise that the use of such labels would influence consumer buying decisions and thereby motivate manufacturer's to provide security updates on their products. They also conducted a user study on the importance of such security update labels to consumer choice and decision with over 1400 participants. Their results indicate that the presence of a security update label, indicating until which date updates are guaranteed accounts, has a 8% to 35 % impact on consumer choice. And, among products with a high perceived security risk, like smart home cameras, the availability of updates seems

to be twice as important as other highly ranked product attributes (like field of view and resolution). Their results also show that the provisioning time for security updates, the time taken for a product be patched upon discovery of a vulnerability, accounts for about 7% to 25% of impact on consumer's choices. They note that the labels are intuitively understood by consumers, do not need third party product assessments prior to release and additionally, incentivise manufacturers to provide sustained security support.

Accounting for security considerations in the design of IoT devices, Medeiros et al. (2018) provide a list of good practices and associated actions that could be taken from both the developer's and user's perspective to ensure safety of IoT devices. The guidelines presented can serve both as way for a company to improve its security stance through taking the actions suggested, and as a way for third party evaluation of a company's security stance. The practices are mapped onto categories such as Information security, Access and credentials, Disclosure, Privacy and transparency and User notification. Their information security guidance for manufacturers includes security protocols, updated cryptography and vulnerability checks on IoT devices and applications, robust mechanisms for distributing updates and correct vulnerabilities, evaluation of security risks and compliance of outsources service and cloud providers and finally, minimal usage of physical inputs, outputs and hardware interfaces like USBs by applications. In the access and credential guidance category, they suggest manufacturers mandating strong passwords and authentication by default, restricting usage of administrative passwords for administrative purposes, manufacturer support or multi factor authentication for password recovery, countermeasures against brute force and abusive login attempts, user notification on password change and outlier login attempts on the device, encryption of stored authentication credentials. On the disclosure, privacy and transparency front, their suggested good practices include manufacturers limiting data collection to bare minimum needed for device operation, making manufacturer's data retention policy and lifetime of personal information storage publicly available, ability for users to reject manufacturer's policy, anonymized information collection by applications for storing at servers. They also suggest guidelines for manufacturers for user notifications such as a communication process to inform users about security problems, privacy issues, product termination or device discontinuity, security events and operational faults.

In direct contrast to proactive considerations of security during design, Wu et al. (2019) suggest a method to incentivise retrospective detection of the security status of IoT devices which might induce manufacturers to fix detected issues and thereby prioritise security. The method suggested involves the use of distributed detectors to overcome the difficulty in conducting a comprehensive, centralized security appraisal of IoT devices. Their proposed method, called SmartRetro, is a block chain powered incentive platform that would incentivise distributed detectors to participate in vulnerabilty analysis and share the detection results. The consumers of SmartRetro would receive automatic security updates when a vulnerability in their installed IoT system is detected. Under this scheme, the developers of IoT are responsible for constructing and maintaining the underlying blockchain and verifying the vulnerability upon detection. However, although their results indicate that the system is technically feasible and economically viable, they do not mention what incentives the developers might have to take part in the scheme given the additional costs (for developement and maintainence) that would be incurred by them. Nonetheless, if the product manages to gain popularity and reaches a tipping point through some initial momentum, that might serve as motivation for other manufacturers to participate.

Neisse et al. (2017) also propose the use of block chain to increase the security of IoT devices, however in their approach, block chain powered smart contracts are used to enhance the transparency and traceability of cybersecurity certification information and to support trusted exchange of such information. Their work is

presented in light of the EU Cybersecurity Act (EuropeanComission, 2018) passed in June 2019, which established a EU certification framework for information and communications technology (ICT) digital products, services and processes to ensure application of a common cybersecurity certification. The framework is voluntary and seeks to establish a central standard instead of member countries adopting separate standards. The proposed approach seeks to balance the need to include data pertinent to a cybersecurity certificate (as a result of a successful certification process) without end user involvement with the need to provide a unified view of the security level of an IoT device throughout its lifecycle. The development of a block chain powered platform, in their opinion, would allow us to meet both those demands of a security assessment platform.

To understand if IoT device manufacturers ship products with good security features and explain the user controlled security features to the consumers, (Blythe et al., 2019) empirically analyse the security information and cyber hygiene advice that is communicated to consumers of IoT devices. They do so by collecting the user manuals and associated support pages for 270 consumer IoT devices produced by 220 different manufacturers. The two researchers then independently read and coded the collected material through a bottoms up approach, identifying all the security features mentioned in these sources and then mapping them to the guidelines found in the UK Government's Secure by Design Code of Practice (CoP) for IoT devices. Their findings indicate that there is very little publicly available information on the security features of IoT devices, on average only four of the 16 security features mentioned in the CoP were discussed in these pages. They also highlight the lack of standardization in the communication of security related features of IoT devices to consumers and argue for the need for government intervention in this space to provide assurances about security of IoT devices.

Nudges for improving security behaviour

Nudging is based on choice architecture and promotes the idea that the manner in which a choice is presented will affect the decision outcome. This section summarizes the findings of key papers related to nudges for improving cybersecurity behaviour.

In order to understand the effects of notifications on security behaviour of consumers van Bavel et al. (2019) conducted an online experiment that drew on the concepts of protection motivation theory (PMT) and used two types of notifications. The first notification was framed as a coping message and gave advice to participants on minimizing their exposure to risk while the second notification was a threat appeal that highlighted the potential negative consequences of not taking appropriate security precautions. Their findings show that while both messages nudged consumers into making more secure choices, the coping message had better results and was as effective as both messages combined. This thus places emphasis on protective coping behaviour and highlights the need for interventions to focus on informing consumers on effective actions to better protect themselves online.

Boehmer et al. (2015) extend and add to research on PMT by proposing and examining the role of a new explanatory variable personal responsibility in the protective behaviour of college students. The results from the first part of their study show that personal responsibility can explain the additional variance observed in protective behaviour of participants after taking into account the effects of traditional threat and coping appraisal variables. The second part of their study built on this result and examined the possibility of influencing personal responsibility through intervention and experimental manipulation. This experimental manipulation showed evidence of a causal relationship between personal responsibility and protective behaviour amongst their participants. Overall, their results indicate that a sense of personal responsibility and a belief in third party responsibility can influence protective behaviour and manipulating the norms related to personal re-

sponsibility is associated with an increase in intentions to engage in such protective behaviour.

Drawing on elements of co-creation and the MINDSPACE framework that brings together different factors influencing behaviour change from various economic and psychological models of behavior change, Coventry et al. (2014) suggest a structured approach for organizations to identify and design nudges that might promote more secure behavioral practices. The approach is termed SCENE and involves in order, scenario elicitation, co-creation of nudges, election of nudges for further development, nudge prototyping and evaluation of the prototypes. These nudges are designed to influence security behaviors at the specific point in the interaction where decisions relevant to security must be made.

A crucial observation from this literature review is that there is scarce work focusing on the role of ecommerce intermediaries in improving the security of IoT devices sold on their portals. Further there is little empirical evidence barring Blythe et al. (2019) on the information asymmetry and lack of transparency of security related features of IoT devices.

3 RESEARCH METHODOLOGY

3.1 DEVICE IDENTIFICATION

This section describes the methodology followed to identify and collect device type and manufacturer information on Mirai infected IoT devices within the Netherlands.

3.1.1 Infected Device Identification

In order to identify Mirai infected devices within the Netherlands, data from Surfnet's Network Telescope was used. Surfnet [1] is an organization that offers high-quality network services to all Dutch educational and research institutions and also hosts a /15 Network telescope. A darknet or network telescope is a system that allows us to observe the network wide events through monitoring of traffic directed to unused IP addresses.

IP addresses and Darknets

An IP address is a unique address that identifies a device on a local network or on the Internet. Similar to phone numbers in telephony, IP addresses are the crucial backbone of internet routing and IP addresses are essential for any device to connect to the internet. An IP address has four sets of numbers between 0 and 255 separated by three dots, for example the IP address of google.com is 172.217.2.110. IP addresses can thus range from 0.0.0.0 to 255.255.255.255 and this entire IP address space is managed globally by the Internet Assigned Numbers Authority (IANA). However, not all of these IP addresses are in active use, and darknet refers to the 'dark' portion of the web that does not host any active services and (Cymru, 2015). Thus, any traffic that is destined to this darknet or the corresponding set of routable, allocated, but unused IP addresses is unsolicited or malicious (Bou-Harb et al., 2017). Surfnet's darknet monitors such unused but allocated IP addresses within the /15 network, although the exact address range is not available publicly (Kumar et al., 2019b). The packet capture data of the traffic hitting Surfnet's darknet is available close to realtime and from this the IP addresses that match the Mirai fingerprint are extracted for further processing to identify the corresponding device type and manufacturer.

Mirai Fingerprint

Communication over the internet is powered by multiple protocols and, prominent among these are IP (Internet Protocol), that uses IP addresses for source and destination identification and TCP (Transport Communication Protocol), which uses sequence numbers for orderly packet processing. While the IP protocol allows us to establish connections, the TCP protocol helps in maintaining an established connection and provides for orderly collection and transmission of packets. Typically, TCP sequence numbers are generated randomly at the beginning of each session. However, either due to design or oversight, the Mirai's scanning algorithm generates TCP sequence numbers as integer representations of the destination IP address.

1 https://www.surf.nl/en

Since the chances of a randomly generated sequence number matching the destination address incidentally is $1/2^{32}$ (Antonakakis et al., 2017), the packets matching this fingerprint are considered to originate from scans of Mirai infected devices and the associated IP addresses are counted as Mirai infected IP addresses.

3.1.2 Device type and Manufacturer identification

Now that we have the Mirai infected IP addresses, the next step is to identify which device the IP address belongs to and to further identify the type of device. If the device type is IoT, then we would also like to identify the manufacturer of the IoT device. The identification technique used in this research is active in nature, which implies that packets are sent to the IP address and the corresponding responses are collected and analyzed for identification. This identification is done through collecting the banners and landing page screenshots from all the open ports of a given IP address. While the IP address identifies the address of the system within a network, ports identify the process or network service running within a system. To make a comparison to traditional telephony, if an IP address is the phone number of an organization, ports are the extension numbers that are dialled to reach various departments or employees within the organization. Each IP address can have a maximum of 65535 ports and consequently 65535 services running simultaneously. The standard ports are the ports associated with common services like Hypertext Transfer Protocol (HTTP) on port 80 and 8080, and Hypertext Transfer Protocol Secure (HTTPs) on port 443 and 8443. HTTP refers to the protocol that is predominantly used for communication between web browsers and web servers, HTTPs is an extension of HTTP and allows for secure communication.

An overview of the steps followed for identification are listed below, and the following sections elaborate on the methodology further.

- Using the gowitness tool [2] licensed under GNU General Public v3 License, the landing page screenshots of each IP address on the standard ports 80, 443, 8080 and 8443 are obtained.
- Using masscan (Graham, 2019), a publicly available tool, banners of all open ports of an IP address is gathered. Masscan scans all ports (0-65535) of an IP address and interacts with the application running at each open port to collect the associated banners.
- Using the results of masscan, gowitness is run again to obtain screenshots of all identified open ports for each IP address. These screenshots and banners contain information that can be leveraged to identify the device type and manufacturer.

Another source of information is the db file generated by each gowitness run, that contains among others, the HTTP headers, title, and SSL certificate fields from every successful screenshot grab. It is worth noting that although we grab information for all standard and open ports, the infected packets might originate from *any, some or all* of these ports.

Banners

Banners are messages that are displayed to users once a connection is established with a device. These banners are configurable by device administrators and are typically used to warn users about the consequences of unauthorized access to the device or to present information regarding the device or the services that is running on the port. Figure 3.1 shows an example of a SSH banner message that is displayed when a user tries to establish a SSH connection with the server, and

2 https://github.com/sensepost/gowitness

displays information about the server (Red Hat Enterprise Linux Server release 6.5), SSH refers to the SSH protocol that is widely used for secure remote access to a system.

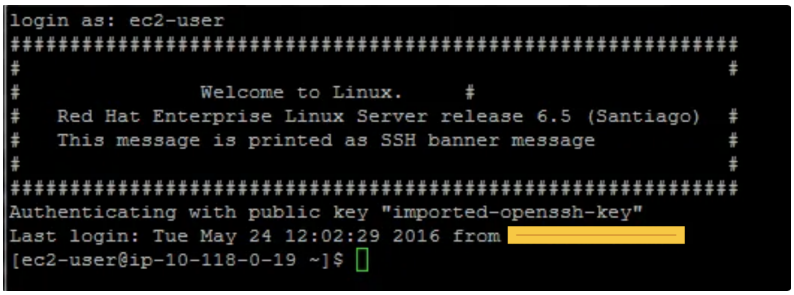

Figure 3.1: SSH Banner (Charles, 2018)

Thus, these banners might provide details of the device which can be used for to identify the device type and manufacturer. In order to collect banners from the open ports of a device, a publicly available tool, masscan (Graham, 2019) was used. Masscan scans all ports (0-65535) of an IP address and interacts with the application running at each open port to collect the associated banners. These banners collected through masscan were first cleaned up by removing the hexadecimal characters and date fields and broken down into words and then passed to a counter that counts the number of occurrences of each unique word. These words and the associated frequency of occurrence were sorted in descending order of frequency (most frequent to least frequent) and manually analysed to find distinct strings or patterns that can help with identifying the corresponding device type or manufacturer, for instance, some banners contain the manufacturers name which can be used for manufacturer identification and others have information related to the running process or OS which can be used to narrow down the device type. After these distinguishable patterns were identified, a ruleset was created with mappings of patterns to device types and/or manufacturer. The ruleset was created through the python DurableRules [3] framework and contains the regular expressions to match the pattern and the corresponding information on the manufacturer and device type, the ruleset has been added to the Appendix 6.1

Landing Page Screenshots

Landing page refers to the page that is shown to the user after a successful connection is established with an IP address. For instance, most home routers have 192.168.0.1 as the default IP address to access the routers admin panel and change configurations. Typing this address into a browser or typing it along with the port number 80 as 192.168.0.1:80, will return the routers landing page which typically has input fields for username and password; upon entering the correct username and password device access is granted. An example of a routers landing page screenshot is provided in figure 3.2, and as can be observed from the figure, the landing page has information on the manufacturer (TP-Link) and also provides details on the device type (450M Wireless N Router, Model No. TL-WR940N).

Thus, landing pages are extremely useful to collect the information we need on device types and manufacturers. Therefore, using a tool named gowitness[4], licensed under GNU General Public v3 License, the landing page screenshots were collected from all the Mirai infected IP addresses on the standard ports 80, 443, 8080 and 8443. Additionally, gowitness was also run to collect screenshots from the other open ports that was identified by the masscan tool (for collecting banners).

3 https://github.com/jruizgit/rules
4 https://github.com/sensepost/gowitness

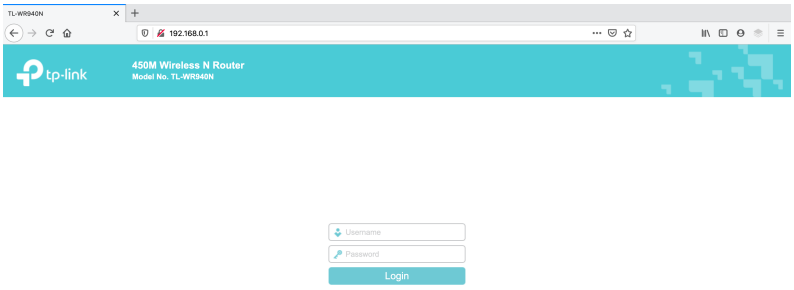

Figure 3.2: Landing page screenshot of a home router

In order to reliably identify manufacturer and device type from the landing page screenshots, first, a large chunk of the images were processed manually (in a group effort) to identify and label the device type and manufacturer info where possible. In some cases, the screenshot contained the name of the manufacturer or device, however, in most cases, we got the information through Google image search which returns a list of websites that have the same or similar images. In the successful cases, manual analysis of the resulting websites gave the needed info. In some cases where a straightforward reverse image search did not yield usable results, image search of the logo or a smaller more feature rich part of the landing page screenshot was used for identification.

Next, for each image, a 16 bit hash was generated using the perception hashing technique (imagehash.phash) publicly available under the python imagehash library [5]. Perception hashing creates analogous hashes when input images are homogeneous and therefore, in our scenario screenshots of similar devices will map to the same hash value.

In the next step, using this labelled data of image hashes, manufacturer and device type info, as yet unidentified screenshots can be labelled. The simplest way would be to compare the hash value of the new screenshot with available hashes and if there is a match, the new image can be tagged with the manufacturer and device type info from the labelled data. However, given that similar images will have a similar perception hash value, this comparison can be extended to similar hashes to facilitate a more expansive identification process.

To that end, the value of the hamming distance between the hashes was used. Hamming distance value is a measure of the number of bit positions where the input strings have different values, a value of zero denotes exact match while a value of one denotes a change in one position. A function to calculate hamming distance was written in python, the definition has been added to the appendix 6.2. Since all the hashes were of the same size (16 bit), the hamming distances were not normalized. For a test set of 97 images, exact matching allowed labelling of 21 images (hamming distance = 0), while using a hamming distance value less than three allowed for (accurately) labelling an additional 32 (33%) images. Hence, for the rest of the dataset, a hamming distance value of less than 3 was taken for matching.

5 https://github.com/JohannesBuchner/imagehash

Gowitness Report

Each run of gowitness creates a db file that can be used to create a report of each run and this db contains the following fields, hash refers to perception hash for the screenshot image captured.

- url
- final_url
- screenshot_file
- response_code
- response_code_string
- headers
  - Server
  - Expiry
  - Retry Count
  - Content-type
- ssl_certificate
  - peer_certificates
  - cipher_suite
- page_title
- hash

In the first step, a python script was written that parses the db file for each day and collects the available fields for each entry. In cases where multiple dissimilar entries were present in the file for each unique IP address and port combination, all the entries were combined and added. From this, the titles were extracted, sorted based on frequency of occurrence and manually analysed for distinguishable patterns that can be used for identification, similar to the process described for the banners. In the same manner, the header and SSL fields were also analysed and used for labelling wherever they contained such identifiable information. These distinguishable patterns were mapped to the corresponding device types and manufacturers through using the python DurableRules framework, this ruleset is also added to the appendix 6.3

3.2 E-COMMERCE CHANNEL IDENTIFICATION

This section describes the methodology followed to identify the ecommerce channels that sell (insecure) IoT devices within the Netherlands.

3.2.1 Website identification

In order to find the list of ecommerce websites that sell insecure IoT devices, the first step was an automated google search. This search was done for a subset of 48 devices (refer table 3.1) from the list of all infected IoT devices identified and only contains devices where the manufacturer is also known. Additionally, routers, switches and NAS were excluded because although these are popularly grouped under IoT devices, they are not representative of typical consumer IoT devices and consequently both the manufacturer and consumer considerations with regard to these devices might differ from typical consumer IoT devices. Furthermore, since

these are network connection oriented devices, they need to have open ports for their functioning which implies that there is higher likelihood that these devices are not themselves infected but are merely listening on the open ports that were scanned.

Table 3.1: Manufacturer and device type combinations used for google search

ABUS Surveillance camera	Maginon camcorder
ABUS DVR	Milestone Video Surveillance Camera (XProtect)
Airspace CCTV	Mobotix MOBOTIX M25
Apexis Network Camera	Modeo MR60 Nettv
Aras Xyclop Camera	NoVus IP camera
Avtech IP Camera	Phillips Hue smart lights
Avtech DVR	Reolink NVR
cabletech DVR TR-008-4HV	Ronin Telecom IP Camera
Vu+ solo2	Sannce IP Camera
Vu+ solo4k	Sansco NVR Security Camera
Vu+ soloSE	SMA SunnyWebBox
Fibaro Home Centre	Smartwares Network camera
Fibaro Home Centre 2	Sompy Alarm System
Foscam IP Camera	Sony Ipela SNC-CH160
HD-Network Camera (ESCAM)	Uniview Unv IP Camera
HikVision Camera	Vacron IP Camera
HikVision DVR	VACRON NVR
HikVision IP Camera	Vimar Elvox Video Door entry
HoneyWell Smart home	X10 AirSight Xx34A
Interlogix IP Camera	Xiong Mai DVR
Interlogix TruVision NVR	Xiong Mai NVR
Loxone Home automation	Xiong Mai IP Camera
MAGINON IPC-250 HDC	Zhejiang Dahua IP Camera
Maginon Security Camera	Zhejiang Dahua IR PTZ Dome Camera

For the automated google search, a python script was written using the freely available google search package [6]. The keywords used for the search were a combination of the manufacturer and device name concatenated with the English terms ”buy ”, ”buy online ” and the equivalent Dutch terms ”kopen ”, ”koop online”; the Dutch terms were included to get a more representative list of websites within the Netherlands. For each of the 48 identified infected devices, the script runs four searches and collects the search results links from which the website names were extracted and added to a python data frame, in total the script returned 951 search results. However, unsurprisingly the data frame had multiple duplicate websites and once these were removed, there were 210 unique websites.

These unique websites were then sorted based on frequency of their occurrence and subsequently, the top 80% of most frequent websites (about 70) was extracted for further analysis, this list is added in Appendix 6.4. The motivating factor for choosing only the top 80%, apart from time considerations, was to ensure that the websites analysed were the popular ecommerce sites within the Netherlands since those that appeared fewer than three times in the results were dropped. The bottom 20% of websites that were not considered for further analysis is added in Appendix **??**

These 70 websites were manually checked to filter out non-ecommerce sites, and only the set of 15 ecommerce websites that ship to the Netherlands were taken for further analysis, these are listed in table 3.3. Websites like marktplaats.nl that

6 https://pypi.org/project/google-search/

merely link to other ecommerce sites and do not offer direct sales were ignored. Additionally, since the aim is to compare products across multiple manufacturers, individual manufacturer websites were discarded since they only sell only the specific device.

The list of websites identified, their corresponding tranco ranking, and frequency of occurence in the results is presented in table 3.3, sorted in ascending order of tranco rankings. Tranco ranking [7] provide the ranking of websites hardened against manipulation by malicious actors. For each of the websites, the tranco ranking was collected using the freely available python package tranco [8]. A ranking of -1 indicates that no corresponding entry was found in the tranco list. In the case of aliexpress.com, the status is partial success because scrape of the generic device search was successful but the specific device search encountered captcha checks for some searches.

Table 3.2: Ecommerce websites in the top 80% of identified websites

Website name	Tranco Ranking	Count
amazon.com	18	20 (1.25%)
nl.aliexpress.com	81	26 (1.62%)
amazon.co.uk	124	15 (0.93%)
amazon.de	161	19 (1.18%)
bol.com	3762	119 (7.45%)
amazon.nl	10331	1 (0.06%)
coolblue.nl	17128	46 (2.69%)
mediamarkt.nl	55488	20 (1.25%)
beslist.nl	58459	55 (3.44%)
onlinecamerashop.nl	832129	19 (1.19%)
maginon.com	-1	24 (1.5%)
ipcam-shop.nl	-1	35 (2.19%)
bewakingscamera-winkel.nl	-1	8 (0.5%)
camerashop24.nl	-1	15 (0.93%)
365cam.nl	-1	3 (0.18%)
en.robbshop.nl	-1	9 (0.56%)
voipshop.nl	-1	5 (0.3%)

3.2.2 IoT product Listings

Once the websites that sell these devices were identified, the next step was to identify the individual product listings in each of the website so that details regarding the price, product name and description, average ratings, total number of ratings and reviews could be gathered. For this purpose, web scraping scripts was written for each of the website. These scraping scripts extract the required data from these websites, leveraging the handy python Beautiful soup [9] library for pulling data out of HTML files. HTML (Hypertext Markup Language) is the markup language used for displaying results in a web browser, it defines the structure and content of a webpage. HTML uses tags for defining how content must be displayed within a webpage, and typically, most websites use the same tags across all of their webpages. Hence, for each website, first, the tags that contain the relevant information was identified by looking at the HTML page source of the webpage and then those tags were used in the scraper script to collect the needed information. The python HTTP requests library [10] was used to send requests to the server, each request returns a response object that contains the content the webpage. A point to note is

7 https://tranco-list.eu/
8 https://pypi.org/project/tranco/
9 https://www.crummy.com/software/BeautifulSoup/bs4/doc/
10 https://requests.readthedocs.io/en/master/user/quickstart/

that some of the results were in Dutch and these were converted to English using the googletrans library [11] which is a free and unlimited library that implements the Google Translate API. The exact steps followed are listed below.

1. Search for the keyword (device and manufacturer combination in 3.1) on the website and return the web page containing the results
2. Collect all the links displayed in the results upto a maximum of 20 links
3. Send a request for each of these links and parse the response page to identify the needed information
4. If results are not in English, call google translate API to convert the content to English
5. Write each link and all the collected information (in English) into a file

Nonetheless, web scraping is not always successful since most websites have checks in place to prevent automated access. In order to overcome the checks - at least partially - random headers from a list of 12 different headers were used in the requests call. A HTTP request header is the information that is sent by a client to a server containing details on the information expected in the response. Thus, when different headers are used, the server gets requests from seemingly different clients thereby bypassing some of the checks. In addition, random delays between 1 and 10 seconds was added between each consecutive requests. However, despite these tricks, some servers still tagged the request as automated and returned a page containing captcha check, therefore it was not possible to collect information from those websites.

Table 3.3: Scrape status

Website name	Scrape status
amazon.com	Fail : Captcha check
nl.aliexpress.com	81 Partial success
amazon.co.uk	Success
amazon.de	Success
bol.com	Fail : Returns 406 error
amazon.nl	Success
coolblue.nl	Success
mediamarkt.nl	Fail : Captcha check
beslist.nl	Success
onlinecamerashop.nl	Fail : Captcha check
maginon.com	Fail : Captcha check
ipcam-shop.nl	Success
bewakingscamera-winkel.nl	Success
camerashop24.nl	Success
365cam.nl	Success
en.robbshop.nl	Fail : Link identification fail
voipshop.nl	Success

Across all websites, a total of 2116 listings were collected using the search term of infected devices. Furthermore, in order to allow for comparison between (known) infected devices and other devices, a generic search containing only the device type from 4.5 was also done on each of the identified websites. The steps followed were the same as those followed for the infected device listings with only a change in the keyword. For each website, these results were also collected separately, and across all the websites, generic device type searches returned 2049 listings.

11 https://pypi.org/project/googletrans/

Table 3.4: Generic search terms

Surveillance camera
DVR
CCTV
Network Camera
IP Camera
ip set top box
Smart home hub
NVR
Security Camera
IP camcorder
smart media player
smart lights
Video Door entry system
Dome Camera

3.2.3 Data cleanup

Once the product listings of Mirai infected devices and other popular devices were collected, the next step was the clean out the listings. This was necessary because although the searches were done for specific device types, unsurprisingly, the results contained listings of other products as well. Therefore, these results were manually analysed to remove non-IoT product listings like batteries, data cables, remote controls and waterproof dome covers for outdoors surveillance cameras. Additionally, some devices like dash cameras were removed since in most cases they do not offer direct internet connectivity. However, spy cameras were retained since they are a subset of IP cameras with additional functionality and design for obscurity. After this cleanup, the specific devices searches contained 527 listings and the generic searches contained 1762 listings.

Moreover, although the infected device search contained the manufacturer name, in some cases the resulting product was from a different manufacturer. Thus, in the next step, for both sets of listings, the manufacturer had to identified and added. This task is non-trivial since most online websites do not provide explicit information on manufacturers and hence for each of the listing the product title and description were manually analysed to identify the manufacturer. Nonetheless, in some cases, it was not possible to identify the manufacturer name from the website listing and the manufacturer field for these entries were left blank. Once the manufacturers were thus identified, both sets of results were filtered to separate listings of Mirai infected device manufacturers from other popular device manufacturers which resulted in a final list of 142 unique listings of Mirai infected devices and 1098 unique listings of other popualr devices.

3.2.4 Additional data collection

Since the security of IoT devices is the primary variable of interest, to aid further analysis, for each identified manufacturer, the number of known vulnerabilities was collected and added from the CVE details website [12] wherever the information was available. The webiste collects CVE vulnerability data various sources like National Vulnerability Database (NVD) maintained by National Institue of Standards and Technology, exploits from www.exploit-db.com, vendor statements, additional ven-

12 https://www.cvedetails.com/

dor supplied data and from Metasploit modules data from the Metasploit computer security project run by the Boston, Massachusetts-based security company Rapid7.

Moreover, to understand the sentiment of consumers for each product, sentiment analysis was performed on each individual review of a product. Sentiment analysis uses Natural Language Processing techniques to determine the attitue or sentiment expressed in a particular topic, whether is positive, negavtive or neutral (Bakshi et al., 2016). Sentiment analysis was done using a pretrained classifier "en-sentiment" available as part of the python Natural Language Processing (NLP) library Flair [13]. Flair utilizes a pre-trained model to detect and prints a label of positive or negative for each review in addition to an integer between 0 and 1 that indicates the degree of confidence. A result of "positive (0.9)" indicates that the review is positive with high certainty. In order to get the sentiment scores for each product, an average of the sentiment value was taken, the sum of the sentiment value for each review of the product was taken and divided by the total number of reviews. The positive and negative labels were taken as signs in calculating the sum, that is, "positive (0.9)" was taken as +0.9 while "negative (0.7)" was taken as -0.7. These averaged sentiment scores were also added to the data set and the final data set contained the following fields.

1. Search term: The term that was used for the search, device and manufacturer combination in case of specific searches and the device type in case of generic searches.
2. Website Link: The link to the specific product listing page.
3. Product Name: The title of the product as it is listed on the web page.
4. Price: The price of the product in Euros, in cases where the prices were not in Euros, python Currency Converter library [14] was used to convert it to Euros.
5. Average Ratings: The average rating of the product on the website (where it is available). The scale of the rating varied from 1 to 5, 1 to 10 and 1 to 100 across different websites. In order to get a uniform scale for comparison, ratings on the scale of 5 and 100 were converted into their corresponding value on a scale of 10.
6. Total Ratings: The total number of ratings available on the website. However, across websites there is a discrepancy in how this value is calculated. Some websites count all ratings, while some websites only count the ratings that have an associated review. Even though the total number of ratings could act as reasonable proxy for device popularity, since it was not possible to normalise this field meaningfully across websites, it was not considered for any further analysis.
7. Reviews: Contains all the reviews for each listing where ever reviews are available.
8. Product Description: The description of the product in the website listing. Some websites also additionally provide product specifications, where this was available, it was also appended to the product description field.
9. Manufacturer: The manually identified manufacturer of the device. However, despite the manual effort it was not possible to identify the manufacturer for some of the devices from the website page.
10. Number of CVEs: The number of known vulnerabilities per manufacturer where ever an entry for the manufacturer was present in the CVE Details database.

13 https://github.com/flairNLP/flair
14 https://pypi.org/project/CurrencyConverter/

11. Number of Products: The total number of products for each manufacturer where ever an entry for the manufacturer was present in the CVE Details database.

12. Sentiment score: The sentiment score for each listing where ever reviews are available. Sentiment score was obtained for each review and the average was added to each listing.

13. Manufacturer Country: The country that hosts the headquarters of the manufacturer was added wherever the information was available.

14. Class: A binary field with 1 for Mirai infected devices and 0 for other popular devices.

3.2.5 Topic Modelling

The data collected from web scraping of the ecommerce websites has two fields of textual data - the product description and customer reviews. Product description is the information that is provided to consumers by the sellers or manufacturers and customer reviews contain the qualitative feedback that of consumers. An analysis of this data will allow us to understand what security information about the product, if any, is marketed to consumers and how consumer perceive the products and if there are any security related concerns that they mention. Owing to large size of the dataset, manual analysis and categorization of the data was not possible and hence topic modelling was performed on both these sets of data. Topic modelling is a type statistical modelling for discovering abstract 'topics' that occur in a collection of documents. The topic modelling technique used is the Latent Dirichlet Allocation (LDA) topic model, a three-level hierarchical Bayesian model, in which each document in a collection is modeled as a finite mixture over an underlying set of topics and each topic in turn is modeled as an infinite mixture over an underlying set of topic probabilities (Blei et al., 2003). Simply put, LDA assumes that each document can be described by a distribution of topics and each topic can in turn be described by a set of associated words (Ganegedara, 2019).

The LDA program first divides the sentences into a group of words and stop words are removed from this group. Stop words are words like a, is, etc., which are often found in the text but do not add any semantic meaning. The list of available stop words was expanded with words returned by the LDA algorithm but which were too generic to be associated with a particular topic (buy, find, etc.).

This list of words is passed to a lemmatization function. Lemmatization refers to the use of vocabulary and morphological analysis of words, to eliminate infectious events and return the dictionary base or form of a word called lemma citep sanderson2010christopher. This helps convert different forms of a word into the same basic form, allowing for a more heterogeneous collection of words.

For this collection of words, a glossary of integer indexes is created for each word using the python dictionary function available in the gensim corpora library. In the next step, the data is processed to create a corpus of ordered pair representations, with a unique global integer id for each word contained in the text and the corresponding frequency of occurrence. This corpus is then transferred to the LDA wrapper function available under the python gensim package that prints words and the corresponding weightages identified from the data.

There are two user-defined parameters that are entered into the LDA model, the number of topics (k) and the number of words for each topic. Once a set of topics has been generated, a coherence test can be used to assess the quality of the results based on the distance between words on the same topic. However, since it is difficult to assess the number of subjects in our a priori data set, coherence score was generated for different k values and the best one was from those available. The ouput of LDA is a set of words and the associated the weightages, the results need

to interpreted manually to label the latent topic that is indicated by the given set of words.

In addition to LDA, another topic modelling that provides better results for a semi-supervised approach were security related words can be input as seed or anchor words were run. Since the identification of these anchor words was based out of our knowledge and experience and through consultation with experts, the validity of the methodology is not sound and therefore this has been added to the Appendix 6.6.

3.2.6 A model for classification

In order to understand the factors that influence the security of these devices, a Linear Support Vector Machine (SVM) model was built to classify the devices into two classes - Mirai infected and uninfected using the python scikit-learn library and thereby understand the weightages of different features of the dataset. A logistic regression model was also tried but since the SVM model had better accuracy rate, only those results and methodology are presented.

SVM is a supervised machine learning algorithm that plots each data point in an n-dimensional space where n represents the number of features and the corresponding feature values are taken as the coordinate values. It then classifies the dataset by finding the hyper place that best differentiates two classes. The linear SVM uses a linear kernel function, which assumes that the data is linearly seperable. Although other polynomial kernel functions like gamma achieved a higher accuracy rate and precision, since it is not possible to determine the weights of each feature when using a non-linear kernel function, they were ignored in favour of the linear kernel.

Table 3.5: Results of SVM classifier for varying C values

C value	Accuracy	Precision Class 0	Precision Class 1
1.00E-05	0.887	1	0
0.0001	0.886	0.89	0
0.001	0.886	0.89	0
0.01	0.886	0.89	0
0.1	0.924	0.95	0.7
1	0.945	0.96	0.82
10	0.924	0.95	0.7
100	0.928	0.95	0.73
1000	0.928	0.95	0.73
10000	0.928	0.95	0.73

In order to run SVM on the dataset, the categorical variables manufacturer country and ecommerce website had to be converted into numerical representation. To do so, the OneHotEncoder available within scikit-learn was used. This encodes the data using a one-hot (aka 'one-of-K' or 'dummy') encoding scheme and creates a binary column for each category. For our dataset, there was 17 different countries and eight ecommerce websites and hence the one hot encoding resulted in 18 binary columns for the manufacturer country, an added category for data points where manufacturer country is Unknown and eight binary columns for websites. Further, the model assumes all values are numerical and hold meaning but in our dataset there are some missing values since not all features were available for all devices. These missing values were therefore handled using the Imputer also available in the scikit-learn library which fills in missing values using the mean of other values for the feature. In order to judge the accuracy and precision of the model, the dataset was divided into testing and training sets, 20% of the dataset was taken as the test set and the remaining 80% was used for training the model. The input parameters to the model the C parameter which provisions for control over the

tradeoff between the model accuracy and maximization of the decision function's margin. A low C value encourages a simpler decision function and a larger margin at the cost of training accuracy. In order to determine the best C value for our data, the model was run for C values ranging from $10^{-}5$ to 10^{5} (3.5) and the best results were obtained for a C value of 1 with an overall accuracy of 0.945 and a precision of 0.96 for class 0 and 0.82 for class 1 which contains Mirai infected devices.

Once the model was built the magnitude and direction of the coefficients for the various features was collected and analysed to determine the relative weightage of different features of each actors.

4 FINDINGS

4.1 INFECTED DEVICE IDENTIFICATION

This chapter presents the results from the analysis on infected IP addresses collected between 26th November 2019 and July 7th 2020. Figure 4.1 shows a graph of the count of Mirai infected IP addresses per day from Surfnet's network telescope collected between 26th November 2019 and July 7th 2020.

Figures 4.1 and 4.2 give an overview of the number of infected IPs that originate within the Netherlands and the associated scans. The number of scans might not linearly rise with an increase in the number of IPs since only available open ports for each IP are scanned.

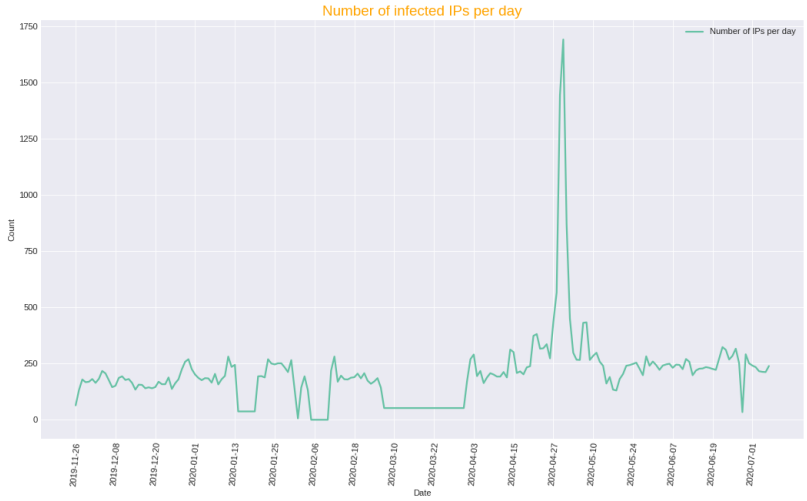

Figure 4.1: Number of infected IPs within NL per day

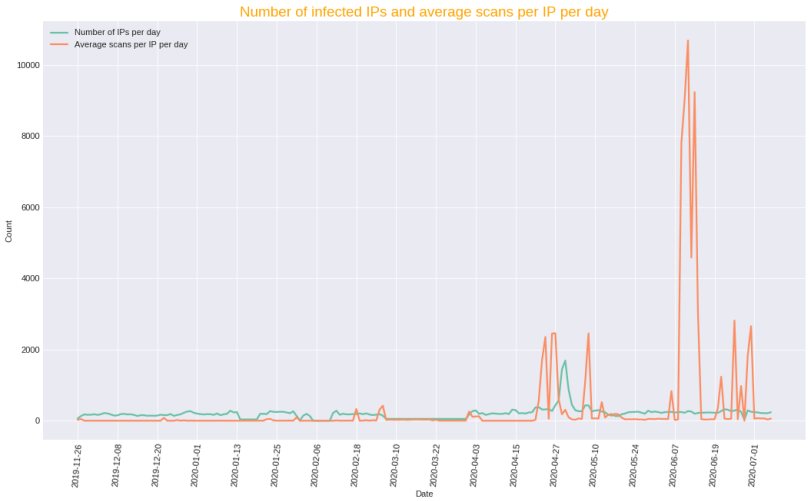

Figure 4.2: Number of scans per IP per day

In order to understand the coverage of labelling from each of the three data sources listed in the methodology, graphs were plotted for each. Figure 4.3 shows the number of devices identified IoT or non-IoT each day using screenshot data. Devices that cannot be confidently ascertained to be either IoT or non-IoT are classified as unknown. Figure 4.4 presents the same data but normalised for the total number of available screenshots each day.

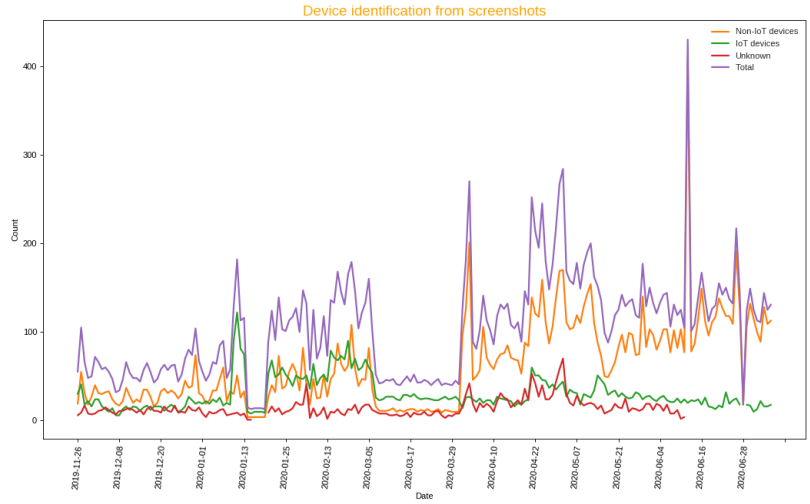

Figure 4.3: Device identification from screenshots

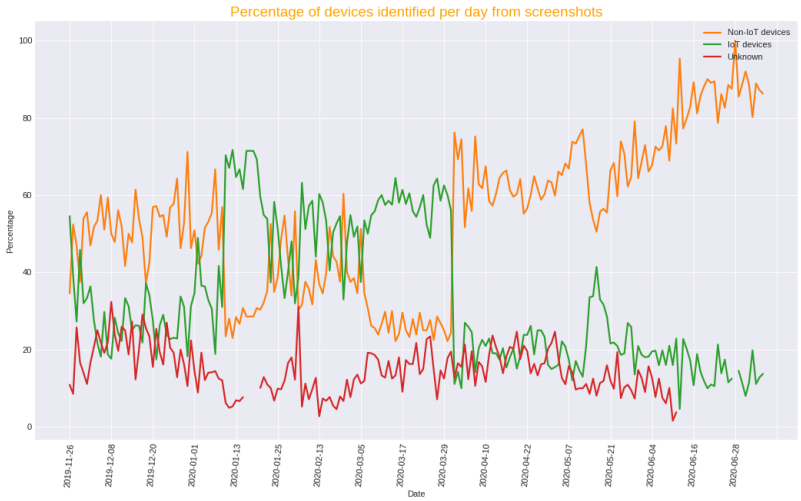

Figure 4.4: Percentage of devices identified per day from screenshots

Figure 4.5 presents the number of identifications possible through banner data and 4.6 presents the corresponding percentages.

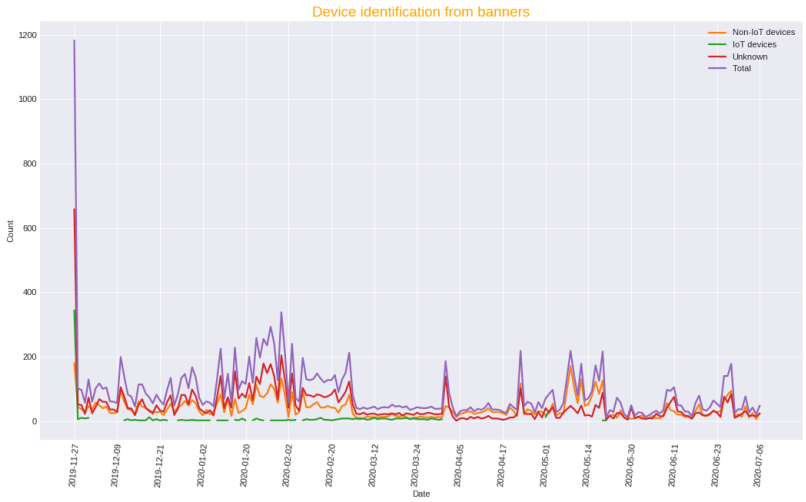

Figure 4.5: Device identification from banners

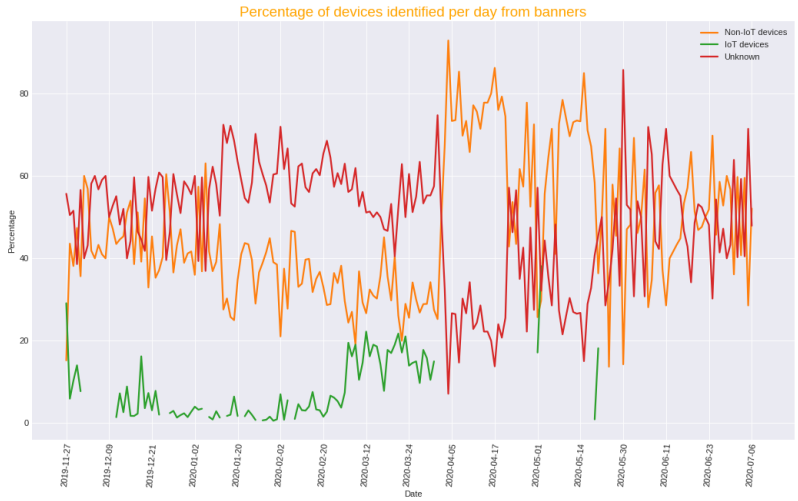

Figure 4.6: Percentage of devices identified per day from banners

The HTML title based identification is presented in figures 4.7 and 4.8.

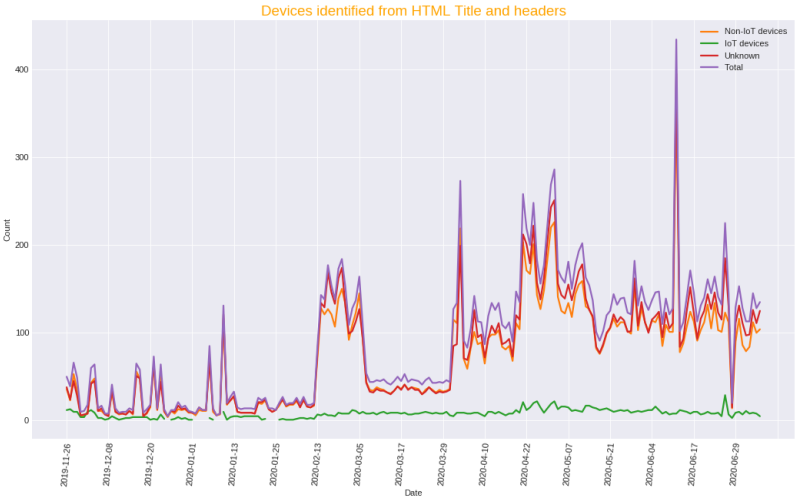

Figure 4.7: Device identification from HTML title

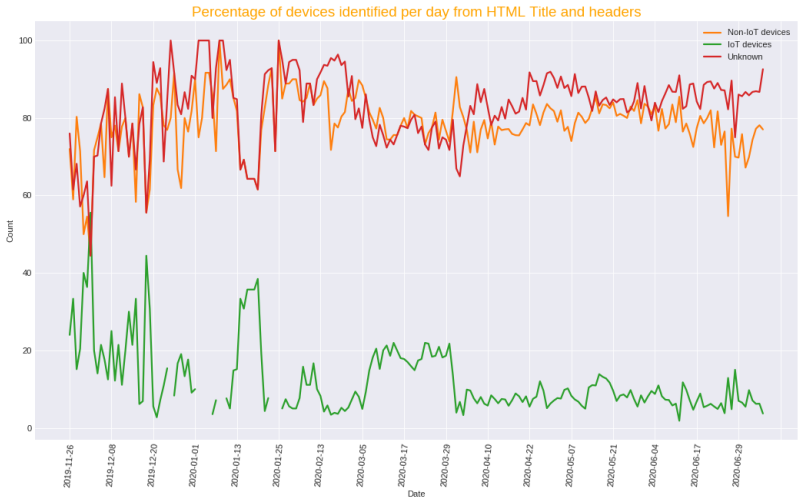

Figure 4.8: Percentage of devices identified per day from HTML title

Table 4.1 shows a summary of the results identification across all three methods.

Table 4.1: Results of device identification across all three methods

	Screenshots	Banners	Gowitness DB
IoT devices			
IP Camera/DVRs/NVRs	249	1583	3*1658
Smart Home Automation	1022	380	
Routers	1218	490	
QNAP	156		
Non IoT devices			
Webservices	2479	8520	2272
Error Page/Unknown	34559	7336	32676
Total number of entries	39683	18309	36606

The following table 4.2 presents unique combinations of Manufacturer, device type that were identified from the available dataset through all the methods combined. – Give count for each device and manufacturer

4.2 ONLINE DISTRIBUTION CHANNELS IDENTIFICATION

4.2.1 Which ecommerce websites sell (infected) IoT devices?

For performing the google search to identify websites that sell infected IoT devices, a subset of 48 devices from the list of infected devices presented in table 4.2 was used. This list is presented in table 3.1 and only contains devices where manufacturer is also known. Moreover, it does not contain routers, switches and NAS because, although these are typically grouped under IoT devices, they are network connectivity oriented devices and therefore they need to have open ports for their functioning. This implies that there is higher likelihood that these devices are not themselves infected but are merely listening on the open ports that were scanned.

The list of websites identified, their corresponding ranking, information on the scrape and frequency of occurence in the results is presented in table 3.3, sorted in

Table 4.2: Manufacturer and device types identified IoT devices

Manufacturer	Device Type	Manufacturer	Device Type
Alphatronics b.v	IP Transceiver	Netgear	Netgear WNDR3700
Asus	Router	NoVus	NoVus IP camera
AVM	FritzBox Router	OctoPrint	3D Printer
Avtech	IP Camera, DVR	pfsense	Router/Firewall
cabletech	DVR TR-008-4HV	QNAP	QNAP QTS
Calian Ltd	SDTS_modulator-83	Resol	DL2 Datalogger
Ceru Co. Ltd	Enigma 2 Set-up box	Sansco	NVR Security Camera
Cisco	Login Page, Cisco ASDM	Synology	Synology DiskStation
Domoticz	Home automation system	TP Link	Gigabit Broadband VPN Router R60
Draytek	Vigor Router, Switch	Ubiquiti Inc	Router (EdgeMax)
Fibaro	Home Centre 2	Unknown	Cccam 2.2.1 Server
Fritz Box	Router	Unknown	Video Recorder
Grandstream	UCM6202 IP PBX	Unknown	Surveillance Camera
HD-Network	Camera (ESCAM)	Vacron	IP Surveillance Camera
HomeWizard	Smart Home System	Vigor	Router
Huawei	Router (model HG659)	Vimar	Elvox Video Door entry system
Interlogix	TruVision NVR	WatchGuard	Access Points
Lavid Technology	Pix-Link AC1200M	X10 Wireless Technol	IP Camera
Linksys	Linksys Smart Wifi L	Xiong Mai	DVR, NVR, IP Camera
Loxone	Home automation	Zhejiang Dahua Technology	IP Camera
Maginon	Security Camera, camcorder	Ziggo	Wi-Fi Modem
MikroTik	Router	ZTE	Router
Milestone	Video Surveillance C	ZyXel	Wireless Router
		Unknown	IP Camera

ascending order of rankings. The rankings are from the tranco ranking [1] which provides the ranking of websites hardened against manipulation by malicious actors. The freely available python package tranco [2] was used for collecting the ranking. A ranking of -1 indicates that no corresponding entry was found in the tranco list. In the case of aliexpress.com, the status is partial success because scrape of the generic device search was successful but the specific device search encountered captcha checks for some searches.

After the data from device specific search was cleaned out and filtered only for infected devices from known vulnerable manufactuers, a total of 142 entries were present. The websites where these products are sold and the associated count is presented in table 4.3. The infected device type and manufacturer and the associated count is presented in table 4.4.

Table 4.3: Count of infected devices found per website

Website	Count
www.amazon.nl	44
www.coolblue.nl	31
www.beslist.nl	26
www.amazon.de	17
www.ipcam-shop.nl	10
www.bewakingscamera-winkel.nl	6
www.camerashop24.nl	5
nl.aliexpress.com	3

- table 4.5 device type and website

1 https://tranco-list.eu/
2 https://pypi.org/project/tranco/

Table 4.4: Count of infected devices found per website

Manufacturer and device type	Count
Foscam IP Camera	47
Reolink NVR	11
Avtech IP Camera	11
Avtech DVR	10
NVR	8
Network Camera	6
HikVision Camera	5
IP Camera	5
ABUS Surveillance camera	4
Surveillance camera	3
Dome Camera	3
Interlogix IP Camera	3
HikVision IP Camera	2
Vu+ solo4k	2
Maginon Security Camera	2
Security Camera	2
CCTV	2
Fibaro Home Centre 2	2
Sannce IP Camera	2
Video Door entry system	1
HD-Network Camera (ESCAM)	1
Phillips Hue smart lights	1
Apexis Network Camera	1
Fibaro Home Centre	1
ABUS DVR	1
HikVision DVR	1
Uniview Unv IP Camera	1
DVR	1
IP camcorder	1
Sony Ipela SNC-CH160	1
Mobotix MOBOTIX M25	1

As outlined in the methodology, on each website, in addition to a search for infected devices and manufacturer combination, a generic search using only the device type was also performed. The list of 14 generic device type search terms used are presented in table 4.5. Since most websites list their best selling products first, the devices returned in this generic search can be considered as representative of the popular devices in each of the categories.

Table 4.5: Generic search terms

Surveillance camera
DVR
CCTV
Network Camera
IP Camera
ip set top box
Smart home hub
NVR
Security Camera
IP camcorder
smart media player
smart lights
Video Door entry system
Dome Camera

4.3 COMPARISION BETWEEN INFECTED AND POPULAR DEVICES

Once the results from both the searches were cleaned, it allowed for comparing the characteristics of both sets of data and testing for statistically significant differences.

4.3.1 Average ratings

In order to understand if there is a difference in customer perception of the devices, an independent sample t-test was performed on the average ratings across both sets of devices. Since the datasets are not of equal lengths, the t-test was done for unequal variance.

The t-statistic value is 2.41 and the result is statistically significant with a pvalue = 0.019 which provides us enough evidence to accept the alternate hypothesis the samples have different means. The boxplot of the average ratings is presented in figure 4.9. Interestingly, the average rating of infected devices has a higher mean value than the generic devices. However, this is in line with other studies that have found that average ratings are do not converge with other indicators of quality like Consumer Reports quality scores (De Langhe et al., 2016)

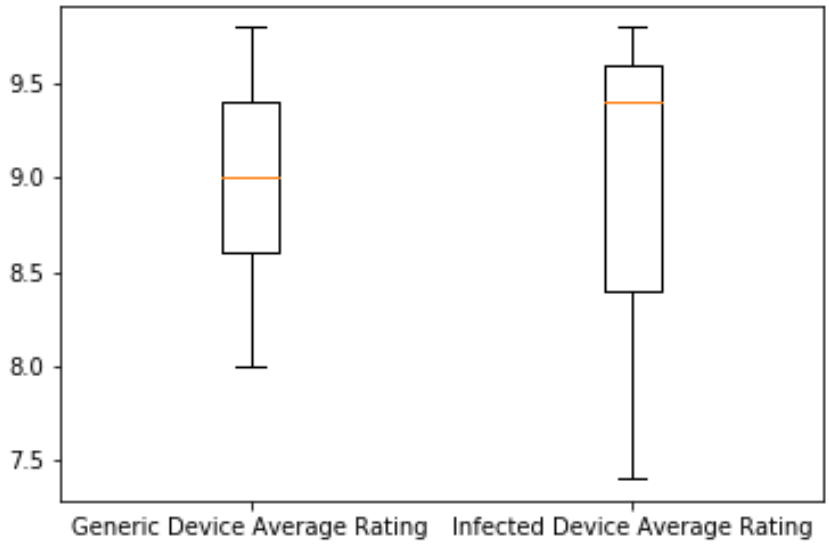

Figure 4.9: Average ratings across both sets of data

4.3.2 Average sentiment score

In order to test if there is difference in consumer sentiment between the two groups of products, an independent sample t-test (with unequal variance) was done on the available average sentiment score of each group. The results are not statistically significant, the p-value is 0.77 and the t-value is 0.29. Although not statistically significant, corresponding box plot in figure 4.10 shows that similar to average ratings, the mean sentiment score of infected devices is higher than that of the generic device.

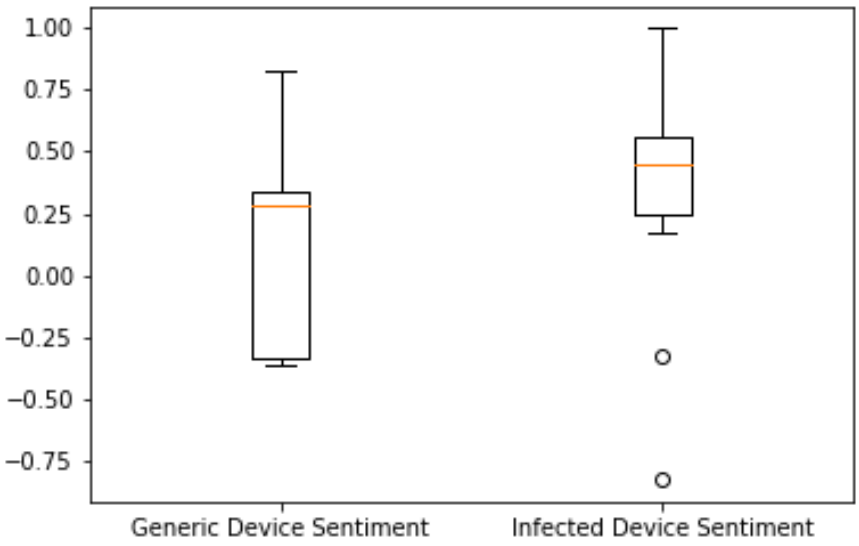

Figure 4.10: Average sentiment score across both sets of data

4.3.3 Number of known vulnerabilities

Next, in order to understand if there is a statistically significant difference in the security of the infected devices when compared to the generic devices, an independent sample t-test (with unequal variance) was done for the the number of known vulnerabilities of each manufacturer in both the lists. The results are statistically significant with a p-value of $1.4e^{-5}$ and a t-statistic value of 4.64. The corresponding box plot is presented in 4.11. The higher mean value for CVEs of infected device also serves to validate that the infected devices identified are indeed more vulnerable.

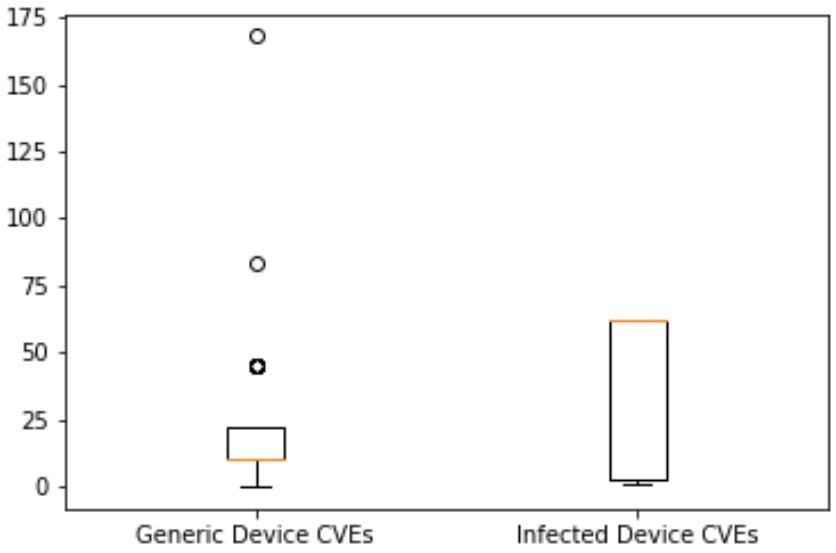

Figure 4.11: Number of vulnerabilities across both sets of data

In addition, t-test was also performed to test for difference in prices and the number of total reviews, however these results were not statistically significant and hence are not presented.

4.4 TOPIC MODELLING

4.4.1 LDA - Customer reviews

The best coherence score (0.3) for the reviews of infected devices was associated with three topics. The words returned and the corresponding weightages are presented below. The topics referred could be 'App based connectivity', 'Network Issues' and 'CCTV performance'.

1. app, connect, easy, video, set
   (0.029, 0.023, 0.022, 0.021, 0.017)
2. problem, work, wifi, make, time
   (0.019, 0.016, 0.014, 0.013, 0.012)
3. camera, quality, system, cable, picture
   (0.045, 0.036, 0.020, 0.019, 0.016)

The highest coherence score (0.373) for reviews of generic devices was for a 17 topics. The words returned and the associated weightages are listed below. As evident

from the list, there is considerable overlap in the words and these were grouped together for topic identification. The topics indicated could be 'Easy camera installation', 'Smart light connectivity', 'Cheap camera', 'App based connectivity', 'Set top box issues', 'Smart TV functionality', 'Physical camera specifications', 'Product Configuration and Update', 'Cloud storage', 'Audio design', 'Camera battery performance', 'Picture Quality', 'Device control'.

1. Installation, easy, cable, cam, plug
   (0.040, 0.032, 0.030, 0.026,0.023)
2. lamp, light, connect, switch, spot
   (0.081, 0.061, 0.038, 0.035,0.033)
3. camera, software, cheap, network, run
   (0.081, 0.038, 0.020, 0.018,0.018)
4. operate, app, show, connection, set
   (0.028, 0.027, 0.022, 0.021,0.019)
5. set, problem, play, star, device
   (0.053, 0.033, 0.030, 0.026,0.021)
6. device, box, work, video, tv
   (0.098, 0.053, 0.035, 0.026, 0.025)
7. connect, find, install, smartphone, include
   (0.038, 0.031, 0.022, 0.020, 0.019)
8. light, time, ring, quality, mount
   (0.025, 0.022, 0.022, 0.021,0.020)
9. product, connection, app, work, configure
   (0.042, 0.032, 0.031, 0.026, 0.025)
10. product, update, connection, internet, excellent
    (0.026, 0.020, 0.019, 0.019, 0.019)
11. app, video, wifi, cloud, home
    (0.042, 0.041, 0.033, 0.032, 0.022)
12. sound, hear, connect, design, perfect
    (0.069, 0.024, 0.023, 0.021, 0.020)
13. camera, time, system, battery, day
    (0.049, 0.022, 0.021, 0.020, 0.019)
14. quality, night, picture, excellent, price
    (0.087, 0.039, 0.036, 0.031, 0.028)
15. control, device, work, button, switch
    (0.051, 0.037, 0.028, 0.025,0.023)
16. set, camera, system, battery, time
    (0.033, 0.033, 0.030, 0.026, 0.024) (
17. work, easy, small, quality, product
    (0.061, 0.044, 0.030, 0.026, 0.024)

Despite the low coherence scores, these topics can serve as a meaningful proxy of customer focus. Taken together they indicate that customers value easy installation and connectivity, battery life, image quality, user friendly device control and access. Interestingly, the tenth topic indicates that customers value product updates. However, this need not neccessarily mean security updates, it could also be updates to product functionality.

4.4.2 LDA - Product Descriptions

In a similar manner, in order to judge the topics present in product descriptions, LDA was run on both sets of data.

The best coherence score for LDA on product descriptions was 0.54, and was assocaiated with three topics. From this it could be postulated that the topics are 'Device power consumption and compatability', 'PoE and night vision' and 'Surveillance camera image quaility'.

1. power, smartphone, wifi, compatible, operation
   (0.029, 0.027, 0.027, 0.023, 0.021)
2. night, poe, vision, easy, high
   (0.038, 0.025, 0.021, 0.017, 0.017)
3. video, camera, image, support, surveillance
   (0.063, 0.030, 0.028, 0.025, 0.021)

For LDA run on product descriptions of generic devices, the best coherence score was 0.46 associated with 14 topics, the words and weightages are given below. After grouping similar terms, the underlying topics indicated could be 'Disk storage for cameras', 'NVR resolution', 'Smartlight colors', 'Night vision', 'Smart home control', 'Surveillance Camera quality', 'Motion detection', 'IP Connectivity', 'Outdoor camera audio support' and 'Intruder detection'.

1. Camera, disk, system, hard, connect
   (0.087, 0.058, 0.057, 0.052,0.039)
2. video, output, record, network, resolution
   (0.096, 0.081, 0.079, 0.055, 0.054)
3. wireless, set, view, camera, image
   (0.059, 0.039, 0.037, 0.037, 0.023)
4. light, color, wifi, smart, meet
   (0.089, 0.068, 0.041, 0.028, 0.028)
5. night, video, sensor, mode, power
   (0.043, 0.037, 0.032, 0.027,0.024)
6. control, smart, home, type, power
   (0.070, 0.044, 0.038, 0.031, 0.026)
7. build, type, app, wifi, power
   (0.045, 0.039, 0.035, 0.035, 0.027)
8. free, time, limited, system, security
   (0.115, 0.113, 0.051, 0.048, 0.041)
9. night, vision, meter, led, cable
   (0.079, 0.061,0.052, 0.034, 0.031)
10. image, camera, backup, surveillance, quality
    (0.060, 0.035, 0.030, 0.030, 0.023)
11. detect, easy, motion, device, cloud
    (0.023, 0.019, 0.018, 0.015, 0.014)
12. image, ip, connect, function, power
    (0.069, 0.049, 0.035, 0.034, 0.027)
13. support, audio, function, motion, outdoor
    (0.080, 0.064, 0.064, 0.057,0.054)

14. card, support, video, alarm, detection
    (0.037, 0.036, 0.035, 0.032, 0.025)

Additionally, the results from the semi-supervised CorEx topic modelling added in 6.6, indicates an output related to security for product descriptions of uninfected devices - encryption of the hard disks.

4.5 SVM CLASSIFICATION MODEL

The SVM classification model was run using a linear function which allows us to access the coefficients of each feature in the dataset. A linear SVM creates a hyperplane that uses support vectors to maximise the distance between the two classes. The SVM coefficients represent the vector coordinates that are orthogonal to the hyperplane and their direction indicates the predicted class, a positive coefficient implies class 1, which represents Mirai infected devices. These features and the corresponding coefficients rounded up to three decimal points are captured in table 4.6.

Table 4.6: Coefficients for each feature in the SVM classifier

Feature	Coefficient
Price	-0.005
Average Ratings	0.487
Number of Products	-0.037
Number of CVEs	0.034
Average Sentiment	-0.387
Total number of reviews	0.018
Belgium	-0.485
Brazil	0
Canada	-1
China	0.813
France	-1
Germany	1.129
Hong Kong	2.959
Italy	0
Japan	1
Netherlands	-1.848
Poland	1.819
South Korea	-0.54
Sweden	0
Switzerland	0
Taiwan	2
US	-2.573
Ukraine	0
Unknown	-2.275
amazon.nl	1.711
ipcam-shop.nl	-1.05
coolblue.nl	1.496
camerashop24.nl	-0.231
bestlist.nl	0.088
bewakingscamera-winkel.nl	-0.275
amazon.de	-1.667
nl.aliexpress.com	-0.072

– Explain more in detail about interpreting coefficients at least one paragraph

In order to understand the weightage of the features of each actor in the market for IoT devices, the consumers, manufacturers and the ecommerce intermediaries, the corresponding coefficients were grouped and their absolute sum was calculated as shown in table 4.7. Since the price of the device, average ratings, average sentiment and the total number of reviews play a role in consumer's buying decision, these are taken to be consumer attributes. For the manufacturer attributes, in addition the country of the manufacturer, the number of products a given manufacturer has and the total number of reported vulnerabilities are considered. Although the price of the device can be an attribute of the manufacturer as well, since the underlying driver for price considerations is market supply and demand, which in turn is reflected in consumer buying decisions, it was added to the consumer attributes. The only intermediary attribute in the model is the ecommerce website. A model with the tranco ranking was tested but it since it had a zero coefficient it was discarded in the final model.

From table 4.7, it can be seen the highest weightage in the model comes from the manufacturer attributes, followed by the intermediary websites while consumer attributes have the least weightage. This shows that consumer perceptions of IoT devices are scarcely affected by the security of the device. Within the manufacturer attributes the weightage of number of products and number of vulnerabilities of a manufacturer only amount to 0.071 of the total manufacturer weightage of 19.512, with the rest being determined by the country of the manufacturer. From table 4.6, it can be seen that devices from manufacturers whose headquarters are in Hong Kong (2.959), Taiwan (2), Poland(1.819), Germany(1.129) and China(0.0813) have a higher weightage in the positive direction, that is these devices belong to the Mirai infected device class while devices from manufacturers with headquarters in US (-2.573), Netherlands (-1.848), South Korea (-0.54) and Belgium (-0.485) belong to the uninfected class. This indicates that security posture of manufacturers might be influenced by the country they are based out of. Similary, for the intermediaries it can be observed that devices purchased on amazon.nl (1.711), coolblue.nl (1.496) and bestlist.nl (0.088) have weightage in the positive direction while those bought from ipcam-shop.nl (-1.05), amazon.de (-1.667), bewakingscamera-winkel.nl (-0.275), camerashop24.nl (-0.231) and nl.aliexpress.com (-0.072) have a negavtive coefficients indicating that these belong to class 0, devices that are not infected by Mirai. Table 4.8 shows the coefficient values from the model for each ecommerce website along with the corresponding Tranco ranking. It is interesting to note that three of the websites that have negative coefficient values (Class 0) have no corresponding Tranco ranking, and the two that have an entry in the Tranco database have a higher ranking than those of websites with positive coefficients. However, since the sample set is small we cannot reliably form conclusions on the corelation between website popularity and number of vulnerable IoT devices that are sold on the website.

Table 4.7: Feature coefficients grouped by actors

Consumer	
Price	0.005
Average Ratings	0.487
Average Sentiment	0.387
Total number of reviews	0.018
	0.897
Manufacturer	
Number of Products	0.037
Number of CVEs	0.034
Belgium	0.485
Brazil	0
Canada	1
China	0.813
France	1
Germany	1.129
Hong Kong	2.959
Italy	0
Japan	1
Netherlands	1.848
Poland	1.819
South Korea	0.54
Sweden	0
Switzerland	0
Taiwan	2
US	2.573
Ukraine	0
Unknown	2.275
	19.512
Intermediaries	
amazon.nl	1.711
ipcam-shop.nl	1.05
coolblue.nl	1.496
camerashop24.nl	0.231
bestlist.nl	0.088
bewakingscamera-winkel.nl	0.275
amazon.de	1.667
nl.aliexpress.com	0.072
	6.59

Table 4.8: Ecommerce intermediary coefficients and corresponding Tranco ranking

Ecommerce Website	Weightage	Tranco Ranking
amazon.nl	1.711	10331
amazon.de	- 1.667	161
coolblue.nl	1.496	17128
ipcam-shop.nl	- 1.05	-1
bewakingscamera-winkel.nl	- 0.275	-1
camerashop24.nl	- 0.231	-1
bestlist.nl	0.088	58459
nl.aliexpress.com	- 0.072	81

5 CONCLUSIONS AND DISCUSSION

The main objective of the thesis was to examine how insecure IoT devices enter the Dutch consumer market and thereby empirically evaluate and analyse the characteristics of the market for IoT devices and the actors involved to make recommendations for policy interventions to improve the security of IoT devices. The findings presented in Chapter 4 provide answers to the main and sub research questions and in this chapter the key findings are summarized in order to draw relevant conclusions and make policy recommendations.

RQ1: Which IoT devices in the Netherlands are commonly infected with Mirai and who is the manufacturer of these infected IoT devices?

It was found that the most commonly infected IoT devices in the Netherlands were network connected cameras varyingly termed as IP cameras, Surveillance cameras, Security cameras, Dome cameras and CCTV all of which provide the same underlying functionality, NVRs and DVRs. The results from the device identification also has smart lights, smart media players, IP set top box, video door entry systems and smart home hubs but they were relatively lesser in number. The manufacturers of these devices were also identified and interestingly none of the manufacturers of Mirai infected devices are located within the Netherlands.

RQ2: Through which retail channels do these insecure IoT devices enter the Dutch Consumer market?

The list of ecommerce channels that sell Mirai infected IoT devices were identified through an online google search and it was found that these devices are sold both on the popular ecommerce sites like Amazon and the less popular ones like bewakingscamera-winkel.

RQ3: How do manufacturers characterize and present information about the security features of their products in these retail channels?

Through analysis of the product descriptions from the ecommerce channels which is the information that manufacturers present to consumers to market their product, it was found that manufacturers do not market any security related features. The topics output by the topic modelling algorithm showed that both manufacturers of Mirai infected devices and other popular devices advertise technical features of the product and the corresponding performance and functionality attributes. In addition, it was observed that there is a statistically significant difference in the number of known vulnerabilities of manufacturers of infected IoT devices versus those of other popular IoT devices, the former group has a higher number of vulnerabilities than the latter which indicates that manufacturers of devices with poor access control do indeed have poor security orientation.

RQ4: How do customer reviews reflect the security concerns of the users of IoT devices?

The results of topic modelling on customer reviews indicate that consumers of both Mirai infected devices and other popular devices are more concerned about product performance, quality, ease of use and connectivity. However, one of topics for other popular devices contained the term update - more specifically product updates when taken in context - which implies that consumers do care about keeping their product up to date. Although security updates are also pushed as product updates, it is not possible to conclude with the available information whether these updates are motivated by security considerations. Moreover, through comparison of the average ratings of infected IoT devices and other popular devices, it was observed that the average ratings were higher for the infected devices which shows

that consumer perceptions do not reflect the security of the device. Sentiment analysis of the consumer reviews however did not show any statistically significant difference in sentiment across the two groups.

Further, the SVM model built to classify Mirai infected devices from other IoT devices shows that the highest contributing factors are the manufacturer attributes, the number of known vulnerabities, the number of products that the manufacturer has and most significantly the country headquarters of the manufacturer. In addition, the model also indicates that there is higher likelihood of infected IoT devices being sold in the ecommerce websites amazon.nl, coolblue.nl, bestlist.nl since these sites have a negative coefficient. Moreover, the results from the model show that consumer attributes contribute very little weightage to the model, which is again an indication that consumer perception of IoT devices is unaffected by the device's security level. Interestingly, the coefficient of average ratings is positive which indicates that devices with higher ratings are more likely to belong to the infected device category, which is in line with the results of the t-test.

5.1 DISCUSSION

The identification of Mirai infected devices and manufacturers shows us the devices and manufacturers that have the poorest security orientation - since access control is one of the most fundamental security features. One of EU wide risk mitigation strategies for improving the cybersecurity of 5G networks is to minimise the exposure to risks stemming from the risk profile of individual suppliers (Group, 2020). To that end, knowledge of these manufacturers and the various e-commerce websites through which these enter the Dutch consumer market could serve as input for policy makers to design targeted interventions.

An interesting observation from this research is that, although the EU Cybersecurity Act that also includes cybersecurity certification for IoT devices was effective from June 2019, none of the IoT devices analysed made any references to the certification. Moreover, from the results of the SVM model it was shown that the country of the manufacturer influences the security of the devices, therefore better import restrictions and market surveillance techniques can be designed to ensure that products imported from foreign countries adhere to EU cybersecurity standards.

Moreover, the supply chain of IoT devices involves different companies manufacturing various components, and in some cases these companies are geographically distributed. Additionally, some devices are sold as White-label products which increase the difficulty of identifying the entity responsible for security features. Given this complexity, it might be a more parsimonious use of resource to design interventions targeted at ecommerce intermediaries which would in turn influence manufacturers through trickle down effect. These intermediaries could also link to a centralized database of vulnerabilities of each manufacturer so that consumers have pertinent information on the security posture of manufacturers prior to buying the devices. Additionally, in scenarios where the devices pose a significant threat to security - with cooperation from the online retailers - it might be possible to track the owners of infected devices and perform product recall or ensure devices are updated for key vulnerabilities.

Since these marketplaces only act as intermediaries between manufacturers and consumers, they typically escape liability law. But, in recent times there has been cases where courts have found online retail stores liable for defective products sold on their platform (Beach, 2019). Historically, strict product liability has not been applicable to designers, manufacturers, and/or retailers of digital products since the consequences have been mostly economic damages (Dean, 2018). Nevertheless, owing to the increasing non-economic costs associated with insecurely developed IoT devices like lost access to crucial services, damage to private property etc., pol-

icy makers will have to take into consideration the allocation of responsibility for harms caused by vulnerable IoT devices.

Furthermore, the finding that average ratings for infected devices are higher than those of generic devices provides empirical proof that the market fails to reflect the security status of devices. However, these ratings can be manipulated, companies sometimes pay people to give high ratings and write positive reviews (Aral, 2013). Nonetheless, this highlights the need to increase consumer awareness about security features of IoT devices. To that end, policy could mandate that online retailers provide a field for security standards of all IoT and digital products. It could then be upto the manufacturer to populate the field based on their security features. This would increase customer visibility of security features and also serve to nudge manufacturers into prioritising the security of their products.

Additionally, the ecommerce websites could display notifications containing security related advice to consumers buying IoT/ICT products on steps that they could take to better protect themselves and ensure the products meet good security standards. As studies by van Bavel et al. (2019) and van Bavel and RodrÃguez-Priego (2016) show, such coping messages are effective in nudging consumers to make more secure choices. Although presenting such advice might dissuade some consumers from buying the devices, in the long run displaying such notifications might improve the reputation of the ecommerce channels which could in turn serve as an incentive for them to display these notifications. Further, studies have shown that consumers ignore security advice when it is deemed to be marketing related, and hence the security advice might be more effective when it is displayed by the ecommerce websites since consumers trust these channels (Redmiles et al., 2016).

In order to ensure real action is taken to improve the security of IoT devices, interventions need to target the actors in the market with the most power - the consumers. They should be aware of security features of IoT devices and have security related information available to ascertain the security level of the devices prior to purchase. However, currently as the results from the topic modelling show there is no security related information presented to consumers in ecommerce channels. Displaying security labels, certificates, or security advice in the ecommerce channels on purchase of IoT devices could in turn motivate manufacturers to prioritise security in order to increase their revenue. However, in order to do so, manufacturers should be willing and able to provide such information on security of their devices. This is chicken and egg problem can be solved through policy intervention that mandates that manufacturers provide such information and ecommerce sites provide the appropriate fields. Although existing research has identified mechanisms to increase consumer awareness about security, the role of ecommerce intermediaries is overlooked. Since they act as key players connecting consumers and manufacturers, they provide an effective means of intervention and displaying the security information on these channels would be the most effective way to ensure consumers making informed decisions at their moment of purchase.

The predominant practice in software industry is to ship features first with security pushed to later releases. However, considering that most of these IoT devices do not have a provision for update, it is imperative that they are developed to sustain with better inbuilt security features following security by design principles. Failure to do so would cause these idIoTs to be prime targets for hackers and the alternative of discarding these devices to buy an upgraded version with better security features creates e-waste that when not recycled properly, wastes limited and precious natural resource. Furthermore, in the aftermath of the Covid-19 pandemic as the world increasingly relies on online services, the consequences of insecurely developed IoT devices are higher and urgent action is needed to improve the security of these devices.

5.2 LIMITATIONS AND FUTURE RESEARCH

Although the methodology described allowed for identifying and labelling device types and manufacturers, is has its limitations. The first limitation is the use of Mirai fingerprint to filter infected IP addresses. Although packets that match this fingerprint can be positively identified as originating from a Mirai infected machine, it cannot be said that *all* mirai probes will match the fingerprint. Since the abnormality isn't critical to the botnet's scanning functionality, a conscientious hacker could have modified the source code to use a random sequence number for the probes, while retaining the rest of the Mirai's stateless scanning algorithm. The next limitation, arises from NAT which implies that we cannot be certain that the devices identified and labelled are indeed the devices that are infected. Moreover, high occurrence of a given device type and manufacturer combination cannot be used as an indication of a higher infection rates of these devices. Owing to DHCP churn, it is possible that the same infected device appears in the dataset under a different IP address. In addition, even without DHCP churn, although IP addresses are scanned only once a day, it is possible that the same underlying device appears in the dataset in subsequent days with the same IP address. Furthermore, masscan, the tool used for collecting banners sometimes fails to collect banners from all open ports. Hence, banner data from protocols like uPnP and RTSP are missing from the dataset. From publicly available scan data from services like Shodan, it is evident that these protocols contain rich information that can be used for device and manufacturer identification. Other available tools like zgrab2 [1] also do not collect data from these protocols by default. The needed modules need to therefore be added to these tool to enable collection of these data. Further research could aim to enhance these zgrab2 or create a custom banner grabber to also collect banners from these services.

The search for websites was done using the freely available google search library which by default searches on google.com, this could explain the relatively high frequency of amazon.com in the results. Furthermore, although the python script was executed from a server, the laptop used to access the server was used to shop on amazon.de multiple times while (almost) never on amazon.nl. The laptop browser history and/or cookies could have influenced the search results and explain the higher frequency of amazon.de in the results. Although manual search on google.nl did not return significantly different results, future work could try to extend the library for search on google.nl. Additionally, searches from a virtual box or a similar environment could help escape the influence of browser history and cookies. Moreover, data was only collected from those websites where it was possible to evade the captcha checks and use of proxy servers could help overcome the captcha check issues that were encountered. Moreover, google search results are personalized based on a user's browsing history, which implies that there might be some other ecommerce websites that are presented to some users but which were not present in the automated search results. Additionally, this analysis only provides a static view of websites that currently list these products. Although some websites list products that are out of stock, other websites do not which limits the search results.

The initial idea for the research, before Covid-19 pandemic, was to contact the local retailers of IoT devices to understand their perspective of security and its corresponding influence on their choice of IoT devices to stock. However, owing to the social distancing measures in place and the related uncertainty, the study was done on the online channels within the Netherlands instead. Future studies could aim to expand the methodology followed to other countries in the EU in order to better understand the EU digital single market for IoT devices. This could also be extended to other countries which would help in comparing and analysing the differences in the market across countries.

1 https://github.com/zmap/zgrab2

Further experiment could be done on the effective design of holistic coping messages to nudge consumers into making more secure IoT product choices. A general advice to change the password of devices upon purchase might mitigate the impact of Mirai and its variants, provided consumers are effectively nudged into doing so. However, other vulnerabilities beyond poor access control might require a more targeted advice and further research could aim to identify and design appropriate coping messages.

The results of the model indicate that there is a higher likelihood of infected devices sold on certain ecommerce sites, however it does not explain the causal mechanism behind this. It could be that these websites are more popular and therefore have a higher number of products being sold in their site, but further research is needed to explore and identify the causation.

6 RULESETS

6.1 BANNER BASED IDENTIFICATION

This section presents the rulesets used for labelling the banners.

```
with ruleset('labelb'):

    @when_all(m.banner.imatches('.*Ubuntu.*') | m.banner.
        imatches('.*Apache.*') | m.banner.imatches('.*Debian
        .*') | m.banner.imatches('.*nginx.*')
             | m.banner.imatches('.*CentOS.*') | m.banner.
                imatches('.*lighttpd.*') | m.banner.imatches
                ('.*lighttpd.*') | m.banner.imatches('.*squid
                .*')
             | m.banner.imatches('.*httpd.*') | m.banner.
                imatches('.*ftpd.*') | m.banner.imatches('.*
                Dovecot.*') | m.banner.imatches('.*imap.*'))
    def isServer(c):
        global cl
        cl = cl+1
        print ("isserver")
        update_flag("0", c.m.ts, c.m.ip, c.m.port)
        update_device("Server", c.m.ts, c.m.ip, c.m.port)

    @when_all(m.banner.imatches('.*Avtech.*'))
    def isAvtech(c):
        global cl
        cl = cl+1
        update_mfg("Avtech", c.m.ts, c.m.ip, c.m.port)
        #update_mfg_device("Avery Berkel", "Weighing Machine
            ", c.m.ts, c.m.ip, c.m.port)
        update_flag("1", c.m.ts, c.m.ip, c.m.port)

    @when_all(m.banner.imatches('.*mediarouter.*'))
    def isHuw(c):
        global cl
        cl = cl+1
        update_mfg("Huawei", c.m.ts, c.m.ip, c.m.port)
        update_device("Router", c.m.ts, c.m.ip, c.m.port)
        #update_mfg_device("Avery Berkel", "Weighing Machine
            ", c.m.ts, c.m.ip, c.m.port)
        update_flag("1", c.m.ts, c.m.ip, c.m.port)

    @when_all(m.banner.imatches('.*Avery Berkel.*ScaleType
        .*'))
    def isAveryB(c):
        global cl
        cl = cl+1
        update_mfg("Avery Berkel", c.m.ts, c.m.ip, c.m.port)
```

```
    update_device("Weighing Machine", c.m.ts, c.m.ip, c.m
        .port)
    #update_mfg_device("Avery Berkel", "Weighing Machine
        ", c.m.ts, c.m.ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*Edge.*'))
def isUBNTE(c):
    global cl
    cl = cl+1
    update_mfg("Ubiquiti Networks", c.m.ts, c.m.ip, c.m.
        port)
    update_device("Router", c.m.ts, c.m.ip, c.m.port)
    #update_mfg_device("Ubiquiti Networks", "Router", c.m
        .ts, c.m.ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*UBNT Router UI.*'))
def isUBNT(c):
    global cl
    cl = cl+1
    update_mfg("Ubiquiti Networks", c.m.ts, c.m.ip, c.m.
        port)
    update_device("Router", c.m.ts, c.m.ip, c.m.port)
    #update_mfg_device("Ubiquiti Networks", "Router", c.m
        .ts, c.m.ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*Fritz.*'))
def isFritz(c):
    global cl
    cl = cl+1
    update_mfg("AVM", c.m.ts, c.m.ip, c.m.port)
    update_device("FritzBox Router", c.m.ts, c.m.ip, c.m.
        port)
    #update_mfg_device("AVM ", "FritzBox Router", c.m.ts,
         c.m.ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)


@when_all(m.banner.imatches('.*Ziggo.*'))
def isZiggoG(c):
    global cl
    cl = cl+1
    update_mfg("Ziggo", c.m.ts, c.m.ip, c.m.port)
    update_device("Wi-Fi Modem", c.m.ts, c.m.ip, c.m.port
        )
    #update_mfg_device("Ziggo", "wifi-modem", c.m.ts, c.m
        .ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*Ziggo.*TC7210.Z.*'))
def isZiggo(c):
    global cl
    cl = cl+1
    update_mfg("Ziggo", c.m.ts, c.m.ip, c.m.port)
```

```
    update_device("Technicolor TC7210 Wi-Fi Modem", c.m.
        ts, c.m.ip, c.m.port)
    #update_mfg_device("Ziggo", "Technicolor TC7210 wifi-
        modem", c.m.ts, c.m.ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*Vigor Router.*'))
def isVigor(c):
    global cl
    cl = cl+1
    update_mfg("Vigor", c.m.ts, c.m.ip, c.m.port)
    update_device("Router", c.m.ts, c.m.ip, c.m.port)
    #update_mfg_device("Vigor", "Router", c.m.ts, c.m.ip,
        c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*JAWS\/1.0.*'))
def isIPCamera(c):
    global cl
    cl = cl+1
    update_flag("0", c.m.ts, c.m.ip, c.m.port)
    update_device("IP Camera", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*TP-LINK Gigabit Broadband
    VPN Router R600VPN.*'))
def isTPLink(c):
    global cl
    cl = cl+1
    update_device("Gigabit Broadband VPN Router R600VPN",
        c.m.ts, c.m.ip, c.m.port)
    update_mfg("TP Link", c.m.ts, c.m.ip, c.m.port)
    #update_mfg_device("TP Link", "Gigabit Broadband VPN
        Router R600VPN", c.m.ts, c.m.ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*ZTE.*corp.*'))
def isZTE(c):
    global cl
    cl = cl+1
    update_mfg("ZTE", c.m.ts, c.m.ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*Microsoft*IIS*'))
def isMicrosoft_IIS(c):
    global cl
    cl = cl+1
    update_flag("0", c.m.id)
    update_mfg("Microsoft", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*Avtech.*'))
def isAvtech(c):
    global cl
    cl = cl+1
    update_mfg("Avtech", c.m.id)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)
```

```
@when_all(m.banner.imatches('.*Domoticz.*'))
def isDomoticz(c):
    global cl
    cl = cl+1
    update_mfg("Domoticz", c.m.ts, c.m.ip, c.m.port)
    update_flag("1", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*Apache.*'))
def isApache(c):
    global cl
    cl = cl+1
    update_flag("0", c.m.ts, c.m.ip, c.m.port)

@when_all(m.banner.imatches('.*nginx.*'))
def isNginx(c):
    global cl
    cl = cl+1
    update_flag("0", c.m.ts, c.m.ip, c.m.port)
```

6.2 HAMMING DISTANCE FUNCTION

```
def hamming_distance(chaine1, chaine2):
    return sum(c1 != c2 for c1, c2 in zip(chaine1, chaine2))
```

6.3 HTML TITLE BASED IDENTIFICATION

This presents the rules used for labelling based on the HTML titles.

```
with ruleset('labeldb'):

    @when_all(m.hdrs.imatches('.*LCAD03FLN.*'))
    def isLinkSys(c):
        global cl
        cl = cl+1
        update_mfg("Linksys", c.m.date, c.m.ip, c.m.port)
        update_device("Dome Camera - LCAD03FLN", c.m.date, c.
            m.ip, c.m.port)
        update_flag("1", c.m.date, c.m.ip, c.m.port)

    @when_all(m.title.imatches('.*Domoticz.*'))
    def isDomoticz(c):
        global cl
        cl = cl+1
        update_mfg("Domoticz", c.m.date, c.m.ip, c.m.port)
        update_flag("1", c.m.date, c.m.ip, c.m.port)

    @when_all(m.title.imatches('.*cpanel.*') | m.title.
        imatches('.*whm.*') | m.hdrs.imatches('.*cpanel.*'))
    def iscPanel(c):
        global cl
        cl = cl+1
        update_mfg("cPanel", c.m.date, c.m.ip, c.m.port)
        update_flag("0", c.m.date, c.m.ip, c.m.port)
```

```
@when_all(m.title.imatches('.*Ziggo TC7210.Z.*'))
def isZiggo(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Ziggo", "Technicolor TC7210 wifi-
       modem", c.m.date, c.m.ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("Ziggo", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*HomeWizard.*'))
def isHomeWizard(c):
    global cl
    cl = cl+1
    #update_mfg_dev("HomeWizard", "Smart Home System", c.
       m.date, c.m.ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("HomeWizard", c.m.date, c.m.ip, c.m.port)
    update_device("Smart Home System", c.m.date, c.m.ip,
       c.m.port)

@when_all(m.title.imatches('.*NZBGet.*'))
def isNZBGet(c):
    global cl
    cl = cl+1
    #update_mfg_dev("NZBGet", "Usenet Downloader", c.m.
       date, c.m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_mfg("NZBGet", c.m.date, c.m.ip, c.m.port)
    update_device("Usenet Downloader", c.m.date, c.m.ip,
       c.m.port)

@when_all(m.title.imatches('.*RouterOS router
   configuration page.*'))
def isMikroTik(c):
    global cl
    cl = cl+1
    #update_mfg_dev("MikroTik", "Router", c.m.date, c.m.
       ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("MikroTik", c.m.date, c.m.ip, c.m.port)
    update_device("Router", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*Synology.*') | m.title.
   imatches('.*Diskstation 414.*'))
def isNASS(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Synology", "Disk Station NAS", c.m.
       date, c.m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_mfg("Synology", c.m.date, c.m.ip, c.m.port)
    update_device("Disk Station NAS", c.m.date, c.m.ip, c
       .m.port)

@when_all(m.title.imatches('.*Fritz.*') | m.ssl.imatches
   ('.*fritz.*'))
```

```
def isFritz(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Fritz Box", "Router", c.m.date, c.m.
        ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("Fritz Box", c.m.date, c.m.ip, c.m.port)
    update_device("Router", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*Vigor.*') | m.ssl.imatches
    ('.*Vigor.*') | m.hdrs.imatches('.*vigor.*'))
def isDraytek(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Draytek", "Vigor Router", c.m.date,
        c.m.ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("Draytek", c.m.date, c.m.ip, c.m.port)
    update_device("Vigor Router", c.m.date, c.m.ip, c.m.
        port)

@when_all(m.title.imatches('.*HUAWEI.*'))
def isHuawei(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Huawei", "Home Gateway HG659", c.m.
        date, c.m.ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("Huawei", c.m.date, c.m.ip, c.m.port)
    update_device("Home Gateway HG659", c.m.date, c.m.ip,
         c.m.port)

@when_all(m.title.imatches('.*OctoPrint.*'))
def isOctoPrint(c):
    global cl
    cl = cl+1
    #update_mfg_dev("OctoPrint", "3D Printer", c.m.date,
        c.m.ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("OctoPrint", c.m.date, c.m.ip, c.m.port)
    update_device("3D Printer", c.m.date, c.m.ip, c.m.
        port)

@when_all(m.title.imatches('.*EdgeOS.*'))
def isUBNT(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Ubiquiti Networks", "Router", c.m.
        date, c.m.ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("Ubiquiti Networks", c.m.date, c.m.ip, c.m
        .port)
    update_device("Router", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*IP.*CAMERA.*') | m.title.
    imatches('.*IP.*Cam.*'))
```

```
def isIPCamera(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Unknown", "IP Camera", c.m.date, c.m
        .ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_device("IP Camera", c.m.date, c.m.ip, c.m.port
        )

@when_all(m.ssl.imatches('.*plex.*'))
def isPlex(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Asus", "AiCloud, Cloud Storage", c.m
        .date, c.m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_mfg("Plex", c.m.date, c.m.ip, c.m.port)
    update_device("Media Server", c.m.date, c.m.ip, c.m.
        port)

@when_all(m.ssl.imatches('.*router.*asus.*'))
def isAsusRouter(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Asus", "AiCloud, Cloud Storage", c.m
        .date, c.m.ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)
    update_mfg("Asus", c.m.date, c.m.ip, c.m.port)
    update_device("Router", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*AiCloud.*'))
def isAiCloud(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Asus", "AiCloud, Cloud Storage", c.m
        .date, c.m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_mfg("Asus", c.m.date, c.m.ip, c.m.port)
    update_device("AiCloud, Cloud Storage", c.m.date, c.m
        .ip, c.m.port)

@when_all(m.title.imatches('.*Dagizmo.*'))
def isDagizmo(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Dagizmo", "HOA Cloud Storage", c.m.
        date, c.m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_mfg("Dagizmo", c.m.date, c.m.ip, c.m.port)
    update_device("HOA Cloud Storage", c.m.date, c.m.ip,
        c.m.port)

@when_all(m.title.imatches('.*fastly.*error.*'))
def isFastly(c):
    global cl
    cl = cl+1
```

```
    #update_mfg_dev("Fastly", "Cloud Provider Error Page
        ", c.m.date, c.m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_mfg("Fastly", c.m.date, c.m.ip, c.m.port)
    update_device("Cloud Provider Error Page", c.m.date,
        c.m.ip, c.m.port)

@when_all(m.title.imatches('.*Cryptshare.*'))
def isCryptshare(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Cryptshare", "File sharing service",
         c.m.date, c.m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_mfg("Cryptshare", c.m.date, c.m.ip, c.m.port)
    update_device("File sharing service", c.m.date, c.m.
        ip, c.m.port)

@when_all(m.title.imatches('.*SABnzbd.*'))
def isSABnzbd(c):
    global cl
    cl = cl+1
    #update_mfg_dev("SABnzbd", "News Reader", c.m.date, c
        .m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_mfg("SABnzbd", c.m.date, c.m.ip, c.m.port)
    update_device("News Reader", c.m.date, c.m.ip, c.m.
        port)

@when_all(m.title.imatches('.*Shell In A Box.*'))
def isShell(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Creator - Markus Gutschke", "Web
        Based Terminal Emulator ", c.m.date, c.m.ip, c.m.
        port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
    update_device("Web Based Terminal Emulator ", c.m.
        date, c.m.ip, c.m.port)

@when_all(m.hdrs.imatches('.*TP-LINK.*R600VPN.*'))
def isTPLink(c):
    global cl
    cl = cl+1
    update_mfg("TP-Link", c.m.date, c.m.ip, c.m.port)
    update_device("Gigabit Broadband VPN Router R600VPN",
         c.m.date, c.m.ip, c.m.port)
    update_flag("1", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*Index of.*'))
def isFileSystem(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Unknown", "File System", c.m.date, c
        .m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)
```

```
    update_device("File System", c.m.date, c.m.ip, c.m.
        port)

@when_all(m.title.imatches('.*Deluge.*'))
def isTorrent(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Deluge", "Bit Torrent Client", c.m.
        date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("Deluge", c.m.date, c.m.ip, c.m.port)
    update_device("Bit Torrent Client", c.m.date, c.m.ip,
         c.m.port)

@when_all(m.title.imatches('.*Cisco.*ASDM.*') | (m.title.
    imatches('.*Fireware.*')))
def isFireWall(c):
    global cl
    cl = cl+1
    update_device("Firewall", c.m.date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*phpMyAdmin.*'))
def isSQLServer(c):
    global cl
    cl = cl+1
    #update_mfg_dev("phpMyAdmin", "SQL Server", c.m.date,
         c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("phpMyAdmin", c.m.date, c.m.ip, c.m.port)
    update_device("SQL Server", c.m.date, c.m.ip, c.m.
        port)

@when_all(m.title.imatches('.*Usermin.*'))
def isUsermin(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Usermin", "Unix Web Interface", c.m.
        date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("Usermin", c.m.date, c.m.ip, c.m.port)
    update_device("Unix Web Interface", c.m.date, c.m.ip,
         c.m.port)

@when_all(m.title.imatches('.*Plesk.*'))
def isWebControlPanel(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Plesk", "Web Control Panel", c.m.
        date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("Plesk", c.m.date, c.m.ip, c.m.port)
    update_device("Web Control Panel", c.m.date, c.m.ip,
        c.m.port)

@when_all(m.title.imatches('.*RabbitMQ.*'))
```

```
def isRabbitMQ(c):
    global cl
    cl = cl+1
    #update_mfg_dev("RabbitMQ", "Message Broker Software
        ", c.m.date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("RabbitMQ", c.m.date, c.m.ip, c.m.port)
    update_device("Message Broker Software", c.m.date, c.
       m.ip, c.m.port)

@when_all(m.title.imatches('.*yii.*'))
def isWebApp(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Yii", "Web Application Framework", c
        .m.date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("Yii", c.m.date, c.m.ip, c.m.port)
    update_device("Web Application Framework", c.m.date,
        c.m.ip, c.m.port)

@when_all(m.title.imatches('.*Metabase.*'))
def isBusAnalytics(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Metabase", "Business Analytics Tool
        ", c.m.date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("Metabase", c.m.date, c.m.ip, c.m.port)
    update_device("Business Analytics Tool", c.m.date, c.
       m.ip, c.m.port)

@when_all(m.title.imatches('.*DDOS-GUARD.*'))
def isDDoSGuard(c):
    global cl
    cl = cl+1
    #update_mfg_dev("DDoSGuard", "Web Hosting Service and
         DDoS Protection", c.m.date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("DDoSGuard", c.m.date, c.m.ip, c.m.port)
    update_device("Web Hosting Service and DDoS
       Protection", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*E-Business Suite.*') | m.
    title.imatches('.*Oracle.*'))
def isBusApp(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Oracle", "Application", c.m.date, c.
       m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_mfg("Oracle", c.m.date, c.m.ip, c.m.port)
    update_device("Application", c.m.date, c.m.ip, c.m.
       port)
```

```
@when_all(m.title.imatches('.*DirectAdmin.*') | m.title.
    imatches('.*EHCP.*') | m.title.imatches('.*LiteSpeed
    .*') | m.title.imatches('.*Vesta.*'))
def isWebControlPanel(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Unknown", "Web Control Panel", c.m.
        date, c.m.ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_device("Web Control Panel", c.m.date, c.m.ip,
        c.m.port)

@when_all(m.title.imatches('.*Bitclick.*') | (m.title.
    imatches('.*HCB.*Group.*')) | (m.title.imatches('.*
    Alibaba.*')) | (m.title.imatches('.*mips.tv.*')) |(m.
    title.imatches('.*manga.*')) | (m.title.imatches('.*
    bol.com.*')) | (m.title.imatches('.*Nielsen.*')) | (m.
    title.imatches('.*axeba.*')) | (m.title.imatches('.*
    recrur.*')) | (m.title.imatches('.*wordpress.*') | m.
    ssl.imatches('.*braatheneiendom.*')))
def isWebsite(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Unknown", "Website", c.m.date, c.m.
        ip, c.m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
    update_device("Website", c.m.date, c.m.ip, c.m.port)

@when_all(m.ssl.imatches('.*Microsoft.*'))
def isMicrosoft(c):
    global cl
    cl = cl+1
    update_device("Microsoft Server", c.m.date, c.m.ip, c
        .m.port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*Outlook.*') | (m.title.
    imatches('.*Webreus.*')) | m.title.imatches('Webmail
    .*') | m.title.imatches('.*Apache.*') | (m.title.
    imatches('.*Best VPS.*')) | (m.title.imatches('.*
    Virtuozzo.*')) | (m.title.imatches('.*SSD VPS.*')) | (
    m.title.imatches('.*Linux.*')) | (m.title.imatches('.*
    Synapse.*')) |
    (m.title.imatches('.*Surfnet.*')) | (m.title.imatches
        ('.*nginx.*')) | (m.title.imatches('.*3CX.*console
        ')) | (m.title.imatches('.*XAMPP.*')) |
    (m.title.imatches('.*Web.*Server.*')) | m.title.
        imatches('.*IIS.*Server.*') | m.title.imatches('.*
        Spark.*Server.*') | m.title.imatches('.*Linkdroid
        .*Server.*'))
def isServer(c):
    global cl
    cl = cl+1
    update_device("Server", "o", c.m.date, c.m.ip, c.m.
        port)
    update_flag("o", c.m.date, c.m.ip, c.m.port)
```

```
@when_all(m.title.imatches('.*VPN.*'))
def isVPN(c):
    global cl
    cl = cl+1
    update_device("VPN", c.m.date, c.m.ip, c.m.port)
    update_flag("0", c.m.date, c.m.ip, c.m.port)

@when_all(m.title.imatches('.*error.*') | m.title.
    imatches('.*Unauthorized.*') | m.title.imatches('.*
    Unavailable.*') | m.title.imatches('.*not.*found.*') |
    m.title.imatches('.*Forbidden.*') | m.title.imatches
    ('.*Bad.*Gateway.*') | m.title.imatches('.*Bad.*
    Request.*'))
def isErrorPage(c):
    global cl
    cl = cl+1
    #update_mfg_dev("Unknown", "Error Page", c.m.date, c.
        m.ip, c.m.port)
    update_device("Error Page", c.m.date, c.m.ip, c.m.
        port)
    update_flag("2", c.m.date, c.m.ip, c.m.port)
```

6.4 top 85% of websites

This contains the name of the website, the number of times it appeared in the results and the cumulative percentage. The total number search results was 951.

```
'https://www.amazon.com', 46, 4.83
'https://www.bol.com', 41, 9.14
'https://www.maginon.com'm 41, 13.45
'https://www.milestonesys.com', 34, 17.03
'http://www.xiongmaitech.com', 33, 20.5
'https://www.vimar.com', 32, 23.86
'https://www.beslist.nl', 31, 27.12
'https://www.interlogix.com', 26, 29.86
'https://www.marktplaats.nl', 25, 32.49
'https://www.avtech.com.tw', 24, 5.01
'https://www.loxone.com', 24, 37.53
'https://nl.aliexpress.com', 23, 39.95
'https://www.amazon.in', 19, 41.95
'https://www.youtube.com', 18, 43.84
'https://www.novuscctv.com', 17, 45.63
'https://krebsonsecurity.com', 16, 47.31
'https://shop.loxone.com', 16, 49.00
'https://sec-consult.com', 13, 50.36
'https://www.zdnet.com', 12, 51.62
'https://manuals.fibaro.com', 12, 52.89169295478443
'https://www.coolblue.nl', 12, 54.15352260778128
'https://www.ipcam-shop.nl', 11, 55.31019978969505
'https://ipvm.com', 10 , 56.36
'https://www.amazon.co.uk', 10, 57.41
'https://www.onlinecamerashop.nl', 10, 58.46
'https://us.dahuasecurity.com', 10, 59.51
'https://www.bewakingscamera-winkel.nl', 9, 60.46
'https://www.mediamarkt.nl', 8, 61.30
'https://www.dahuasecurity.com', 8, 62.14
'https://www.sedsystems.ca', 7, 62.88
'https://nl.hardware.info', 7, 63.61
'https://www.flipkart.com', 7, 64.35331230283911
'http://www.chillingeffects.org', 7, 65.08
'http://www.vacron.com', 7, 65.82
'https://www.camerashop24.nl', 6, 66.45
'https://cambodia.desertcart.com', 6, 67.08
'https://www.x10.com', 6, 67.71
'https://m.nl.aliexpress.com', 5, 68.24
'https://www.home-assistant.io', 5 , 68.77
'https://www.desertcart.com.kw', 5, 69.29
'https://www.fibaro.com', 5, 69.82
'https://www.voipshop.nl', 5, 70.35
'https://www.kieskeurig.nl' , 4, 70.76
'https://www.globenewswire.com', 4, 71.18
'https://tweakers.net' , 4, 71.60
'https://en.robbshop.nl', 4, 72.029
'https://www.dectdirect.nl', 4, 72.45
'https://www.voipango.nl', 4, 72.87
'https://www.synology.com', 4, 73.29
'https://www.lorextechnology.com', 4, 73.71
'https://www.cctvwinkel.nl', 4, 74.13
```

```
'https://www.nchsoftware.com', 4, 74.55310199789695
'http://epg.com.pt', 4, 74.97
'http://vuplustv.com', 4, 75.39
'https://kodi.wiki', 4, 75.81
'https://dutch.alibaba.com', 3, 76.13
'http://www.zoobelli.com', 3 , 76.44584647739 22
'https://www.365cam.nl', 3, 76.76
'https://www.dektec.com', 3, 77.07
'https://www.newtec.eu', 3, 77.39
'https://www.linkedin.com', 3, 77.70
'https://www.calian.com', 3 , 78.02
'https://www.landashop.com', 3 , 78.33
'https://www.vergelijk.nl', 3, 78.65
'https://www.xiongmaitech.com' , 3, 78.96
'https://www.security.nl', 3, 79.28
'https://www.aliexpress.com', 3, 79.6
'https://www.bestbuy.com' , 3, 79.9
'https://www.pinterest.com', 3 , 80.23
```

6.5 BOTTOM 20% OF WEBSITES

This contains the websites that were not considered for further analysis.

https://www.epine.nl
https://apps.apple.com
https://richinaction.com
https://www.deltalight.com
https://www.consumer.ftc.gov
https://www.securityworldmarket.com
https://www.bnsdistribution.eu
https://myteleurel.com
https://shop.primacom.nl
https://www.betaalbare-domotica.nl
http://lacavernedalunbaba.com
http://www.vedicom.nl
http://www.btvtechniek.nl
https://www.kvviko.nl
http://amatco.co
https://www.security.honeywell.com
http://beauty-virgin.com
https://www.want.nl
http://dronekopen.info
https://www.acoba.com
https://www.beveiligingswinkel.nl
https://www.ezviz.eu
https://belize.desertcart.com
http://www.vacron.jp
http://kokos.ovesenterprise.ro
https://dir.indiamart.com
https://www.pentestpartners.com
https://www.digitalcameraworld.com
https://www.banggood.com
http://swry.robertaabitidasposa.it
https://www.gereedschapland.nl
https://www.telecomshop.nl
https://nl.qwe.wiki
https://www.thalesgroup.com
https://www.megateh.eu
https://www.amazon.ae
https://www.networkwebcams.co.uk
https://habr.com
https://gsglobalsecurity.com
https://www.vacron-eg.com
https://www.voipsupply.com
https://www.bestbuy.ca
https://www.cctvcalculator.net
https://www.qnapsecurity.com
https://routerantenna.blogspot.com
http://www.pix-link.com
https://www.whitelabelhaircare.com
https://www.voipon.co.uk
https://www.sourcesecurity.com
https://www.obj.ca
https://gathering.tweakers.net

https://clipchamp.com
https://www.helpnetsecurity.com
http://areadymedia.com
https://calpere.com
https://www.imotionsecurite.com
https://lumendatabase.org
https://www.smarthomemagazine.nl
https://www.startpagina.nl
https://www.novuscctv.com
https://wildcamerakopen.nl
https://cilo.nl
https://www.techsmith.com
https://www.safewise.com
https://www.cameranu.nl
https://obsproject.com
https://webshop.slv.nl
https://pflege-schierling.de
https://maxict.nl
http://www.authinx.com
https://apps.dtic.mil
https://www.qnap.com
https://www.indiamart.com
https://boingboing.net
https://www.mediakind.com
http://www.camera-sdk.com
https://fibarobenelux.com
http://www.avi-store.com
https://www.videosurveillance.com
https://novelsat.com
https://www.dtsdigitalcctv.co.uk
http://zmlkawy.com
https://www.marktplaats.nl
https://www.domoticz.com
https://www.apowersoft.nl
https://www.cnn.com
https://nl.dissnornim.net
https://webcamera.io
https://www.ebay.co.uk
https://www.ipphone-warehouse.com
https://www.interlogix.com.au
https://gocart.zengcheng123.com
https://www.activeonline.com.au
http://mitrelli.com
https://www.snuffelhoek.nl
https://www.safety.com
http://dl.multiservizispeedy.it
https://organicswadeshi.com
https://www.centralpoint.be
https://www.theverge.com
https://en.wikipedia.org
https://egypt.souq.com
https://www.ip-camerawinkel.nl
https://saudi.souq.com
http://bluelinetalkradio.com
https://www.world-of-satellite.co.uk
https://www.supra-space.de

https://www.powerplanetonline.com
https://botswana.desertcart.com
https://www.velleman.eu
https://www.wmrecorder.com
https://www.pbtech.co.nz
http://downloads.eminent-online.com
https://www.audiovolt.nl
https://www.lorextechnology.co.uk
https://www.thehomeautomationstore.com
https://openpli.org
http://softjapan.co.jp
https://pro.sony
http://enokvirgulino.com.br
https://www.datona.nl
http://stanne.com
https://lavid.en.alibaba.com
https://www.panasonic.com
https://blog.alterdesk.com
https://albania.desertcart.com
https://www.metshop.nl
https://www.jumia.com.ng
http://cowa-carparts.nl
https://camlytics.com
https://www.hrb-beveiligingen.nl
https://www.shiftcomputers.nl
https://www.televes.com
http://www.hts.bg
https://www.eycom.nl
https://www.centralpoint.nl
https://www.beveiligingenzo.nl
https://www.aldi.co.uk
https://www.aldi.nl
https://www.konigelectronic.com
https://www.larcoz.nl
https://www.amazon.ca

6.6 COREX TOPIC MODELLING

In order to better analyse if any security related topic is present, topic modelling with anchor words related to security was performed using the Correlation Explanation (CorEx) model. CorEx uses an information-theoretic approach to learning latent topics over documents and seeks to identify maximally informative topics as encoded by their total correlation (Gallagher et al., 2017). And, unlike LDA, CorEx topic model makes few assumptions about the latent structure of the data, and flexibly incorporates domain knowledge through user-specified "anchor words."

CorEx topic modelling was done for product descriptions and customer reviews from infected devices and also for those from the other popular devices from the generic search, amounting to four sets of data. For each dataset, first the punctuations from the text were removed and then the sentences were broken into a list of commas separated words. This list of words was then passed to a function that removes stop words. Stop words are words like 'is, a, an, the, in' etc., which occur extremely frequently in text but are of little value in understanding the semantic structure of the text. For our analysis, the list of stop words available in the python Natural Language Toolkit (NLTK) library was extended with words output as a result of CorEx but which were too generic to be associated with any one particular topic, these include words like "like", "make", "sure", "means", "to", "want", "choose", "enter" and so on. This set of words was then passed to a CountVectorizer available as part of the python scikit-learn library [1]. CountVectorizer converts this collection of words to a vector of term/token counts. From this vectorizer object of counts, the corresponding feature names or words are extracted using the vectorizer.get_feature_names function, also available as part of the scikit learn library [2] and the CorEx topic model was run on this collection of words.

In order to arrive at a optimal number of topics for each dataset, a graph was plotted for varying number of topics, 'n' ranging from zero to ten for descriptions and zero to 14 for reviews. Since CorEx identifies topics based on total correlation, the graph plots the contribution of each additional topic to the total correlation. It was experimentally observed that when the additional correlation explained by a topic is less than 3, then there is little significance to the topic. Therefore, the number of topics was taken to be n where the coorelation of (n+1) was less than 3. These graphs are shown in images 6.1, 6.2, 6.3 and 6.4.

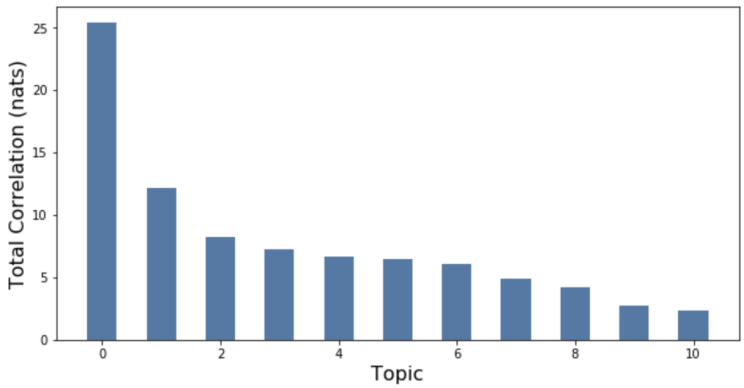

Figure 6.1: Correlation for product description - infected devices

The optimal number of topics was found to be 14 for both sets of reviews, nine for the product descriptions of infected devices and ten for product description of other popular devices. The semi-supervised CorEx modell was also done using words related to security as the seed topics. These words were 'patch', 'update', 'secure',

1 https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
2 https://scikit-learn.org/stable/modules/feature_extraction.html

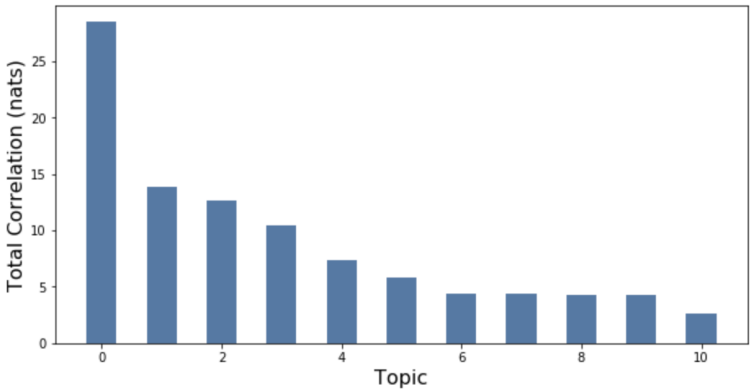

Figure 6.2: Correlation for product description - other popular devices

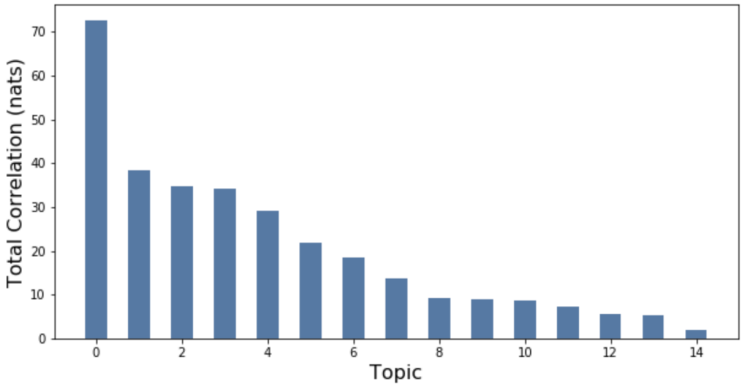

Figure 6.3: Correlation for customer reviews - infected devices

'protect', 'encryption', 'protocol', 'security', 'safe'. Both the unsupervised and semi-supervised CorEx modelling algorithm return for each topic, a set of words and the associated weightages. The results are given below.

Infected devices Product description

Unsupervised 0: carry,width,cm,height,apple,depth,apps,location,manufacturer,fisheye
1: dimensions,264,manual,specifications,audio,p2p,compression,temperature,input,format
2: interface,best,provide,expansion,purchase,care,consider,bnc,recorders,contact
3: reolink,fits,5mp,work,super,nvr,live,videos,3g,anytime
4: angle,fps,dynamic,smartphone,onvif,lens,nas,upnp,pan,sd
5: access,gb,protocol,panel,base,memory,wireless,country,saves,months
6: battery,event,pixels,days,assembly,monitor,image,alarm,possible,storage
7: infrared,images,microphone,dark,built,camera,night,viewing,app,resolution
8: indoor,room,115,colors,comes,180,ezlink,noise,ap,assured

Semi - supervised 0: secure,security,encryption,protocol,cut,wpa,lux,802,humidity,ce
1: nas,onvif,av,warranty,dynamic,pan,tilt,support,upnp,zoom
2: temperature,input,mac,output,os,consumption,dc,local,ddns,level
3: client,hd,days,app,compression,night,dimensions,264,battery,dark
4: supports,surveillance,devices,ipad,web,access,directly,panel,internet,3g
5: fps,frames,diameter,ipcam,psia,dns,eptz,fisheye,pixels,windows
6: disk,hard,movement,recorder,recording,pre,anytime,sata,usb,nvr
7: item,leds,super,easy,vision,videos,wide,amazon,external,cmos
8: type,apple,carry,width,cm,height,manufacturer,depth,apps,garantie

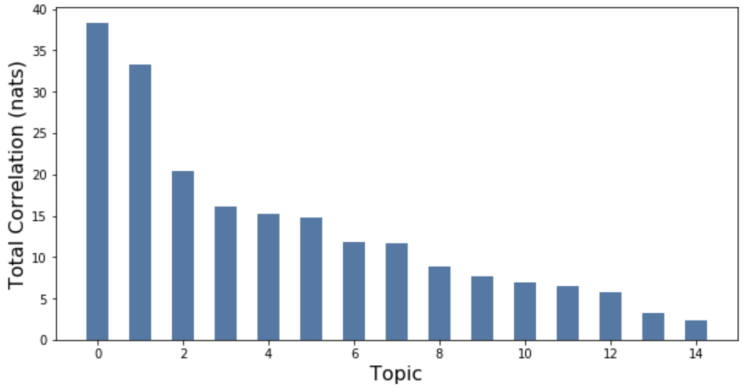

Figure 6.4: Correlation for customer reviews - other popular devices

Other popular devices Product description

Unsupervised
0: browser,264,ipad,iphone,sata,inputs,problem,local,tb,tune
1: recorder,images,connect,meters,sharp,menu,cameras,channel,disk,hard
2: carry,apps,guarantee,manufacturer,width,height,garantie,depth,platform,years
3: advice,obligation,country,address,ntsc,professional,playback,contact,input,1ch
4: alexa,apple,google,assistant,cm,code,wit,voor,ifttt,home
5: power,buyer,nenjoy,returns,china,possible,benefit,sale,sellers,clear
6: wdr,supplier,lpr,horiz,boxed,nightvision,extended,wi,fi,ean
7: card,sd,pan,tilt,detection,support,function,motion,battery,email
8: 20m,cabling,f2,supplies,correction,times,extra,0vp,making,maximum
9: 720,pixels,assembly,alarm,client,android,dns,1920,1080,fps

Semi - supervised 0: encryption,security,overwritten,complete,movement,offer,capacity,digital,memory,con
1: manual,264,ir,auto,ddns,agc,mobile,balance,compression,illumination
2: battery,storage,seconds,event,device,videos,push,model,second,glass
3: pan,tilt,card,sd,detection,function,motion,support,onvif,poe
4: images,cameras,recorder,resolution,sharp,connect,viewing,infrared,meters,image
5: voor,van,en,op,het,met,wifi,kleur,ondersteuning,quot
6: equipped,cable,features,point,order,connected,connecting,leds,select,makes
7: apple,carry,windows,apps,height,assistant,cm,width,type,depth
8: advice,obligation,country,address,laptop,months,maintained,professional,contact,ntsc
9: nee,kaart,bediening,hoogte,xaoen,xaoip,dag,gewicht,bewegingsdetectie,xaox

Infected devices Customer Reviews

Unsupervised 0: days,easily,won,white,setup,unfortunately,cell,got,better,apps
1: monitor,including,android,away,extremely,attached,overall,case,notifications,big
2: far,view,recording,second,satisfied,bit,connected,video,picture,ideal
3: software,recordings,internet,install,problems,point,viewing,options,cameras,phone
4: fine,object,dhcp,turn,100,code,enable,obviously,feeling,worst
5: download,unit,think,year,buying,alerts,way,cloud,router,brilliant
6: perfectly,right,desktop,provided,directly,screw,understand,brand,took,kit
7: detection,review,little,using,bought,having,months,don,light,thing
8: problem,points,half,models,does,mode,higher,quick,managed,heavy
9: optional,reasons,deduction,triggers,impossible,mains,hope,20,life,massive
10: open,screws,times,cheaper,lots,mounting,pay,impressed,total,amazing
11: stream,angle,imagine,moves,period,skewed,qr,combination,chosen,netwerk
12: systems,ago,person,subscription,voice,probably,terms,waterproof,smart,battery

13: program,app,opening,images,chrome,receiving,pity,directions,transfer,het

Semi - supervised
0: security,update,software,great,recordings,connected,video,problems,needed,point
1: correct,qr,code,network,waterproof,model,wanted,purchase,area,home
2: bit,installation,comes,720p,storage,ip,screws,hd,slot,object
3: changed,surveillance,vision,hardware,money,questions,lot,value,mac,highly
4: meters,larger,sensor,terms,indoors,strong,optical,alternatives,piece,date
5: attention,youtube,qnap,led,manufacturers,c1,similar,format,viewed,cam
6: direct,client,pretty,space,holes,item,objects,feeling,existing,enabled
7: perfectly,install,installed,switch,connection,supply,new,internet,server,ordered
8: ghz,thank,smartphones,independently,handling,quiet,band,12,switched,tells
9: detection,visible,using,function,available,ir,sound,clearly,range,instructions
10: send,problem,pc,know,does,hard,try,browser,work,drive
11: cheaper,kit,perfect,second,directly,lots,reliable,feature,desktop,zoom
12: download,alerts,got,certain,included,different,build,decided,remotely,amazing
13: car,base,later,footage,outside,required,sufficient,lose,professional,material

Other Popular devices Customer Reviews

Unsupervised
0: say,price,unfortunately,power,cable,stars,small,able,play,amazon
1: return,kit,drive,start,base,isn,turned,according,save,fee
2: comparison,risk,language,1000,knowledge,module,256,servers,saw,265
3: access,absolutely,need,comes,free,connection,screws,support,micro,settings
4: function,complete,insert,code,storage,directly,store,offer,option,alert
5: cameras,image,recording,quality,detection,camera,videos,live,motion,vision
6: network,long,work,know,purchase,recommend,time,far,annoying,bought
7: great,latest,list,decided,screen,possibilities,button,voice,ok,explained
8: review,don,definitely,little,alexa,using,real,phone,led,sound
9: hear,possible,photo,example,available,app,switch,select,hope,provided
10: een,contains,browser,te,build,en,burglars,het,chose,maar
11: automatically,mobile,smoothly,clock,downside,sharing,carport,ball,summary,baby
12: services,family,handy,goes,calling,development,mount,response,research,agenda
13: advantage,nice,place,interested,technology,living,automation,capacity,badly,remember

Semi - supervised 0: security,secure,update,safe,10,doesn,check,router,finally,protect
1: note,appears,think,servers,main,happened,error,subscription,explain,following
2: software,liked,install,wanted,sent,market,extremely,wifi,hand,systems
3: unfortunately,short,configure,installation,recording,cameras,images,does,impression,signal
4: voice,decided,hung,differences,al,versatility,choice,showed,000,maar
5: power,use,price,able,cable,instructions,simple,detection,say,absolutely
6: personal,unless,tape,month,lack,fits,een,rings,te,van
7: turn,darker,react,guy,30,buzz,e27,350,interface,led
8: door,heard,manually,addition,self,360,continuous,click,certain,installing
9: control,compared,status,exactly,blocked,hoping,flash,euro,mag,mentioned
10: read,device,recognize,list,added,reset,currently,usual,ask,states
11: little,start,state,link,programming,fix,appear,multiple,stop,come
12: picture,decent,needed,issue,password,include,technical,experience,screw,sharp
13: function,problems,minutes,small,connected,download,using,alexa,cloud,need

Although google translate was used to convert the Dutch text to English, in some cases it fails for smaller words.

BIBLIOGRAPHY

(2019). The internet of things.

(2020). Secure and safe internet of things.

Agarwal, S., Oser, P., and Lueders, S. (2019). Detecting iot devices and how they put large heterogeneous networks at security risk. *Sensors*, 19(19):4107.

Akerlof, G. A. (1978). The market for "lemons": Quality uncertainty and the market mechanism. In *Uncertainty in economics*, pages 235–251. Elsevier.

Anderson, R. and Moore, T. (2006). The economics of information security. *science*, 314(5799):610–613.

Angrishi, K. (2017). Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets. Technical report.

Anstee, D. (2019). Rise of the internet of things (iot).

Antonakakis, M., April, T., Bailey, M., Bursztein, E., Cochran, J., Durumeric, Z., Alex Halderman, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y., Antonakakis Tim April, M., Bernhard Elie Bursztein, M., Cochran Zakir Durumeric Alex Halderman Luca Invernizzi, J. J., Kallitsis, M., Kumar, D., Lever Zane Ma, C., Mason, J., and Sullivan Kurt Thomas, N. (2017). *Understanding the Mirai Botnet*.

Aral, S. (2013). The problem with online ratings.

Asghari, H., van Eeten, M., and Bauer, J. M. (2016). Economics of cybersecurity. In *Handbook on the Economics of the Internet*. Edward Elgar Publishing.

Bakos, Y. (2001). The emerging landscape for retail e-commerce. *Journal of economic perspectives*, 15(1):69–80.

Bakshi, R. K., Kaur, N., Kaur, R., and Kaur, G. (2016). Opinion mining and sentiment analysis. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 452–455. IEEE.

Beach, H. (2019). "sellers" beware: Online marketplaces could see increased liability for allegedly defective products.

Bhardwaj, K., Miranda, J. C., and Gavrilovska, A. (2018). Towards IoT-DDoS Prevention Using Edge Computing. Technical report.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Blythe, J. M., Sombatruang, N., and Johnson, S. D. (2019). What security features and crime prevention advice is communicated in consumer iot device manuals and support pages? *Journal of Cybersecurity*, 5(1):tyz005.

Bodenheim, R., Butts, J., Dunlap, S., and Mullins, B. (2014). Evaluation of the ability of the shodan search engine to identify internet-facing industrial control devices.

Boehmer, J., LaRose, R., Rifon, N., Alhabash, S., and Cotten, S. (2015). Determinants of online safety behaviour: Towards an intervention strategy for college students. *Behaviour & Information Technology*, 34(10):1022–1035.

Bou-Harb, E., Lucia, W., Forti, N., Weerakkody, S., Ghani, N., and Sinopoli, B. (2017). Cyber meets control: A novel federated approach for resilient cps leveraging real cyber threat intelligence. *IEEE Communications Magazine*, 55(5):198–204.

Brass, I., Tanczer, L., Carr, M., and Blackstock Word, J. (2017). Title: "Regulating IoT: Enabling or Disabling the Capacity of the Internet of Things?". Technical report.

Cetin, O., Ganan, C., Altena, L., Kasama, T., Inoue, D., Tamiya, K., Tie, Y., Yoshioka, K., and van Eeten, M. (2019). Cleaning Up the Internet of Evil Things: Real-World Evidence on ISP and Consumer Efforts to Remove Mirai. Internet Society.

Charles, K. (2018). How-to display a warning banner before the login prompt.

Comission, E. (2020). The eu cybersecurity certification framework.

Coventry, L., Briggs, P., Jeske, D., and van Moorsel, A. (2014). Scene: A structured means for creating and evaluating behavioral nudges in a cyber security environment. In *International conference of design, user experience, and usability*, pages 229–239. Springer.

Cymru, T. (2015). The darknet project, 2015.

De Donno, M., Dragoni, N., Giaretta, A., and Spognardi, A. (2017). Analysis of DDoS-capable IoT malwares. In *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017*, pages 807–816. Institute of Electrical and Electronics Engineers Inc.

De Langhe, B., Fernbach, P. M., and Lichtenstein, D. R. (2016). Navigating by the stars: Investigating the actual and perceived validity of online user ratings. *Journal of Consumer Research*, 42(6):817–833.

Dean, B. (2018). An exploration of strict products liability and the internet of things. *SSRN Electronic Journal*.

Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. (2015). A Search Engine Backed by Internet-Wide Scanning.

ENISA (2016). Baseline security recommendations for iot in the context of critical information infrastructures.

EuropeanComission (2018). Cybersecurity act.

FTC (2015). Iot privacy security in a connected world.

Gallagher, R. J., Reing, K., Kale, D., and Ver Steeg, G. (2017). Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.

Ganegedara, T. (2019). Intuitive guide to latent dirichlet allocation.

Graham, R. (2019). robertdavidgraham/masscan.

Group, N. C. (2020). Cyber security of 5g networks : Eu toolbox of risk mitigating measures.

Herley, C. (2009). So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 workshop on New security paradigms workshop*, pages 133–144.

ITU (2012). Y.iot-overview.

Jerkins, J. A. (2017). Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017*. Institute of Electrical and Electronics Engineers Inc.

Kleinhans, J.-P. (2018). Improving iot security in the eu.

Krebs, B. (2020). Mirai covid variant disregards stay at home orders.

Kumar, D., Garg, D., Alperovich, G., Kuznetsov, D., Gupta, R., Shen, K., Case, B., and Durumeric, Z. (2019a). *All Things Considered: An Analysis of IoT Devices on Home Networks*.

Kumar, S., Vranken, H., van Dijk, J., and Hamalainen, T. (2019b). Deep in the dark: A novel threat detection system using darknet traffic. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4273–4279. IEEE.

Le, F., Ortiz, J., Verma, D., and Kandlur, D. (2019). Policy-Based Identification of IoT Devices' Vendor and Type by DNS Traffic Analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11550 LNCS, pages 180–201. Springer Verlag.

Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of industrial information integration*, 6:1–10.

Martin, J., Rye, E., and Beverly, R. (2016). Decomposition of MAC Address Structure for Granular Device Inference.

Medeiros, L. S., Zuvanov, F., de Mello, F. L., and Strauss, E. (2018). IoT Information Security Evaluation for Developers and Users. *Journal of Information Security and Cryptography (Enigma)*, 4(1):16.

Meidan, Y., Bohadana, M., Shabtai, A., David Guarnizo, J., Ochoa, M., Tippenhauer, N. O., and Elovici, Y. (2017). ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis.

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A. R., and Tarkoma, S. (2017). IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. In *Proceedings - International Conference on Distributed Computing Systems*, pages 2177–2184. Institute of Electrical and Electronics Engineers Inc.

Morgner, P., Freiling, F., and Benenson, Z. (2018). Opinion: Security lifetime labels-overcoming information asymmetry in security of iot consumer products. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 208–211.

Morgner, P., Mai, C., Koschate-Fischer, N., Freiling, F., and Benenson, Z. (2019). Security Update Labels: Establishing Economic Incentives for Security Patching of IoT Consumer Products.

Munro, K. (2018). Why is consumer iot insecure?

Neisse, R., Hernández-Ramos, J. L., Matheu, S. N., Baldini, G., and Skarmeta, A. (2017). Toward a Blockchain-based Platform to Manage Cybersecurity Certification of IoT devices. Technical report.

Neshenko, N., Husak, M., Bou-Harb, E., Celeda, P., Al-Mulla, S., and Fachkha, C. (2019). Data-Driven Intelligence for Characterizing Internet-Scale IoT Exploitations. In *2018 IEEE Globecom Workshops, GC Wkshps 2018 - Proceedings*. Institute of Electrical and Electronics Engineers Inc.

News, E. (2020). Ecommerce in europe: 717 billion in 2020s.

Rajput, S. and Singh, S. P. (2019). Identifying industry 4.0 iot enablers by integrated pca-ism-dematel approach. *Management Decision*.

Redmiles, E. M., Kross, S., and Mazurek, M. L. (2016). How i learned to be secure: a census-representative survey of security advice sources and behavior. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 666–677.

Storey, A. (2014). There's nothing 'smart' about insecure connected devices. *Network Security*, 2014(7):9–12.

Tanemo, F., Osaki, M., Waki, H., Ishioka, Y., and Matsushita, K. (2020). A method of creating data for device-information extraction by efficient wide-area-network scanning of iot devices. In *2020 International Conference on Information Networking (ICOIN)*, pages 643–648. IEEE.

van Bavel, R. and RodrÃguez-Priego, N. (2016). Nudging online security behaviour with warning messages: results from an online experiment. Technical report, Joint Research Centre (Seville site).

van Bavel, R., Rodríguez-Priego, N., Vila, J., and Briggs, P. (2019). Using protection motivation theory in the design of nudges to improve online security behavior. *International Journal of Human-Computer Studies*, 123:29–39.

Voolf, D. and Cohen, R. (2020). Naming shaming web polluters: Xiongmai.

Wu, B., Xu, K., Li, Q., Liu, Z., Hu, Y. C., Zhang, Z., Du, X., Liu, B., and Ren, S. (2019). SmartCrowd: Decentralized and automated incentives for distributed IoT system detection. In *Proceedings - International Conference on Distributed Computing Systems*, volume 2019-July, pages 1106–1116. Institute of Electrical and Electronics Engineers Inc.

Yu, D., Zhang, L., Chen, Y., Ma, Y., and Chen, J. (2020). Large-scale iot devices firmware identification based on weak password. *IEEE Access*, 8:7981–7992.