



**Advanced Cyber-Threat Intelligence, Detection, and Mitigation
Platform for a Trusted Internet of Things
Grant Agreement: 786698**

D6.1 State-of-the-art on profiling, detection and mitigation

Work Package 6: Title of Work package

Document Dissemination Level

P	Public	<input checked="" type="checkbox"/>
CO	Confidential, only for members of the Consortium (including the Commission Services)	<input type="checkbox"/>

Document Due Date: 28/02/2019

Document Submission Date: 05/03/2019



Co-funded by the Horizon 2020 Framework Programme of the European Union



Document Information

Deliverable number:	D6.1
Deliverable title:	State-of-the-art on profiling, detection and mitigation
Deliverable version:	1.0
Work Package number:	WP6
Work Package title:	State-of-the-Art on profiling, detection and mitigation (M10)
Due Date of delivery:	28/02/2019
Actual date of delivery:	28/02/2019
Dissemination level:	PU
Editor(s):	Stavros Shiaeles (CSCAN)
Contributor(s):	Stavros Shiaeles, Keltoum Bendiab, Julian Ludlow, Salam Ketab, Muhammad Ali, Abdulrahman Alruban (CSCAN) Liza Charalambous, George Boulougaris, Michael Skitsas (ADITESS) Dimitrios Kavallieros, Vasiliki-Georgia Bilali, George Kokkinis (KEMEA) Nicholas Kolokotronis, Costas Vassilakis, Spiros Skiadopoulos, Christos Tryfonopoulos, Konstantinos Limniotis, Christos-Minas Mathas, Sotirios Brotsis (UOP)
Reviewer(s):	Dimitris Kavalieros (KEMEA) Gohar Sargsyan (CGI)
Project name:	Advanced Cyber-Threat Intelligence, Detection, and Mitigation Platform for a Trusted Internet of Things
Project Acronym	Cyber-Trust
Project starting date:	1/5/2018
Project duration:	36 months
Rights:	Cyber-Trust Consortium

Version History

Version	Date	Beneficiary	Description
0.05	22/12/2018	CSCAN	Tentative ToC proposed
0.10	28/12/2018	CSCAN	Deliverable's ToC finalised
0.20	15/02/2019	CSCAN	Sections added
0.30	20/02/2019	ADITESS	Sections added
0.40	21/02/2019	UOP	Sections added
0.50	22/02/2019	KEMEA	Sections Added
0.60	23/02/2019	CSCAN	Formatting of the document and send to review
0.70	27/02/2019	CSCAN	Review received and applying changes
1.00	05/03/2019	CSCAN	Final Submission

Acronyms

ACRONYM	EXPLANATION
ACL	Access Control List
AES	Advanced Encryption Standard
API	Application Programming Interface
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
AV	Antivirus
BGP	Border Gateway Protocol
BLE	Bluetooth Low Energy
BS	Base Stations
BSN	Base Station Network
C&C	Command and Control
CAM	Content Addressable Memory
CII	Critical Information Infrastructure
CoAP	Constrained Application Protocol
COM	Communication Port
CPU	Central Processing Unit
CSP	Communication Service Provider
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DM	Data Mining
DNS	Domain Name System
DOS	Denial of Service
DPI	Deep Packet Inspection
EC-GSM-IoT	Extended Coverage-GSM-IoT
ED	End Device
EGP	Exterior Gateway Protocol
eMTC	enhanced Machine Type Communication
ETL	Extract Transform Load
FHE	Fully Homomorphic Encryption
FTP	File Transfer Protocol
FWSM	Firewall Services Module
GDPR	General Data Protection Regulation
GHZ	Gigahertz
GSM	Global System for Mobile communications
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGS	Integration Gateway Services
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
IPv6	Internet Protocol version 6

IRC	Internet Relay Chat
ISM	Industrial, Scientific, and Medical Radio Band
ISP	Internet Service Provider
ITS	Intelligent Transportation System
JPG / JPEG	Joint Photographic Experts Group
KNN	K-nearest neighbour
Li-Fi	Light Fidelity
LoRa	Long Range PHY and WAN
LPWA	Low Power Wide Area
LPWAN	Low Power Wide Area Network
LTE	Long-Term Evolution
LTE-MTC	LTE-Machine Type Communication
M2M	Machine to Machine
MAC	Media Access Control
MCU	Microcontroller Unit
MDM	Mobile Device Management
MDMP	Mobile Device Managers Plus
MDU	Multi-dwelling Units
MS	Microsoft
NAT	Network Address Translation
NB-IoT	Narrow-Band IoT
NIC	Network Driver Interface
NIDS	Network intrusion detection system
OS	Operational System
P2P	Peer-to-peer
PCAP	Packet Capture
PMIC	Power Management Integrated Circuit
PPDM	Privacy Preserving Data Mining
PPT	Polynomial Probabilistic Time
PS	Profiling Service
QID	Quater in die
RAM	Random-access memory
RF	Radio Frequency
RFC	Request for Comments
RFIC	Radio Frequency Integrated Circuit
RIP	Routing Information Protocol
RSA	Rivest, Shamir, and Adelman cryptosystem
RTOS	Real-Time Operating System
SAVE	Static Analyser for Vicious Executables
SDA	Smart Device Agents
SDK	Software Development Kit
SDN	Software-Defined Networking
SGA	Smart Gateway Agents
SLAAC	Stateless Address Auto Configuration
SMB	Server Message Block
SMC	Secure Multiparty Computation
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOM	Self-Organizing Maps
SSH	Secure Shell

SSL	Secure Sockets Layer
SSO	Single Sign-On
STUN	Session Traversal Utilities NAT
TCP	Transmission Control Protocol
TV	Television
UDP	User Datagram Protocol
UI	User Interface
UNB	Ultra Narrow Band
USB	Universal Serial Bus
VM	Virtual Machine
VPN	Virtual Private Network
Wi-Fi	Wireless Fidelity
ZK	Zero-Knowledge

Table of Contents

Executive Summary.....	11
1. Introduction	12
1.1 Purpose of the document.....	12
1.2 Relations with other activities in the project	12
1.3 Structure of the document	12
2. IoT Devices profiling methods	14
2.1 Introduction.....	14
2.2 Device Profiling.....	14
2.3 SDA with Cloud Services	18
2.4 SDA operating on Linux-based distribution.....	19
2.5 SDA App	20
2.6 Services.....	21
2.6.1 Computer Services	21
2.6.2 Router Services	22
2.6.3 Camera Services.....	22
2.6.4 Smartphone Services & Tablet Services	22
2.6.5 Gateway Services	23
2.6.6 Categories Services	23
2.7 IoT Connections	24
2.7.1 Short-range wireless	24
2.7.1.1 Bluetooth mesh networking and Bluetooth low energy.....	24
2.7.1.2 ZigBee.....	24
2.7.1.3 Z-wave	24
2.7.1.4 Wireless (Wi-Fi).....	25
2.7.1.5 IPv6 Low-power wireless Personal Area Network (6LowPAN)	25
2.7.1.6 Thread	25
2.7.1.7 Light Fidelity (Li-Fi)	25
2.7.2 Cellular Technologies.....	25
2.7.2.1 Extended Coverage-GSM-IoT (EC-GSM-IoT)	25
2.7.2.2 Narrow-Band IoT (NB-IoT).....	26
2.7.2.3 LTE-Machine Type Communication (LTE-MTC) and enhanced Machine Type Communication (eMTC).....	26
2.7.3 Long range Wireless Technologies	26
2.7.3.1 LoRa (LoRa PHY and LoRaWAN)	26
2.7.3.2 Sigfox.....	26
2.7.3.3 Weightless.....	27
2.8 Log files.....	27

2.9	Network profiling.....	30
2.9.1	Network Fundamentals	31
2.9.2	Profiling the Network.....	32
2.9.3	Passive Network Protocol Capture and Analysis	34
2.9.4	Protocols	35
2.9.4.1	Internet Protocol (IP)	36
2.9.4.2	Address Resolution Protocol (ARP).....	36
2.9.4.3	Internet Control Message Protocol (ICMP).....	37
2.9.5	Ports.....	37
2.9.6	Volume Traffic & Muddy Filtering	38
3.	State of the art in malware detection and mitigation	42
3.1	Introduction.....	42
3.2	Malware.....	42
3.3	Classification of Malware	43
3.3.1	Viruses	43
3.3.2	Worms	44
3.3.3	Trojans	44
3.3.4	Bots	45
3.3.5	Ransomware	45
3.3.6	Backdoors	46
3.3.7	Spyware and Adware	46
3.3.8	Rootkits	46
3.4	Malware Analysis Techniques	46
3.4.1	Basic Static Analysis	47
3.4.2	Advance Static Analysis.....	47
3.4.3	Basic dynamic analysis.....	48
3.4.4	Advanced Dynamic Analysis	49
3.5	Signature-Based Techniques	49
3.5.1	Snort.....	50
3.5.1.1	Pcap.....	51
3.5.1.2	Decode and Pre-processing	51
3.5.1.3	Detection engine.....	51
3.5.2	Suricata	52
3.5.2.1	Development and features	52
3.5.2.2	IDS/IPS.....	53
3.6	Behaviour Based Techniques.....	53
3.6.1	Bro IDS	53
3.6.1.1	Components of Bro IDS.....	54

3.6.1.2	Machine learning concepts and definitions.....	54
3.7	Evasion of Malware and Anti-Evasion Approaches	56
3.7.1	An overview of Evasion Approaches and Malware Camouflage Evolution.....	56
3.7.1.1	Encryption	57
3.7.1.2	Oligomorphism.....	57
3.7.1.3	Polymorphism	58
3.7.1.4	Metamorphism	58
3.8	Anti-Evasion Approaches.....	59
3.8.1	Malware Deobfuscation	59
3.8.2	Unpacking	59
3.8.3	Binary rewriting and editing	60
3.8.4	Malware binary reconstruction	60
3.8.4.1	Malware Unpacking	60
3.8.4.2	Malware Normalization	60
4.	The quest for privacy in the IoT	62
4.1	Introduction to IoT and its Applications	62
4.1.1	Smart Homes	62
4.1.2	Healthcare.....	62
4.1.3	Supply Management.....	62
4.2	The need for Privacy-preserving data mining	63
4.2.1	User Privacy	65
4.2.2	Privacy Issue in Data Mining	66
4.2.3	Confidentiality Issues in Data Mining	66
4.2.4	Semi-Honest Adversaries.....	66
4.3	Heuristic-Based Techniques and tools	66
4.3.1	Data Perturbation	68
4.3.2	Cryptographic Technique.....	68
4.3.3	Blocking based technique	68
4.3.4	Condensation Approach	68
4.3.5	Hybrid technique	69
4.3.6	Data Anonymization	69
4.4	Cryptography-Based Techniques and tools.....	69
4.4.1	Secure Multiparty Computation	70
4.4.2	Security in the multiparty computation	70
4.4.2.1	Adversarial power.	70
4.4.2.2	Feasibility of secure multiparty computation.	71
4.4.3	Homomorphic Encryption Techniques	71
4.4.3.1	Homomorphic Encryption.....	72

4.4.3.2	Somewhat Homomorphic Encryption.....	72
4.4.3.3	Fully Homomorphic Encryption	72
4.4.3.4	Limitations and Generations.....	73
4.4.4	Zero-knowledge proofs.....	73
4.5	Reconstruction-Based Techniques and tools	74
5.	Conclusion.....	78
6.	References	80

List of Figures

Figure 2.2: SDA Monitoring through Cloud Networks.....	18
Figure 2.3: SDA Hybrid Monitoring through Cloud Networks	19
Figure 2.4: SDA operating on Linux Based Device	19
Figure 2.5: Hybrid Mobile app development	20
Figure 2.6: The IoT Network Environment [10]	31
Figure 2.7: The TCP/IP Protocol Suite [174]	32
Figure 2.8: Example of SiLK Workflow [85]	34
Figure 2.9: Sensor Functionality [85].....	35
Figure 2.10: High-Level Cyber-Trust Network Monitoring Approach.....	35
Figure 2.11: IPv4 and IPv6 Differences	36
Figure 2.12: The ARP Packet Structure	37
Figure 2.13: Common TCP/UDP Ports [63]	38
Figure 2.14: An example Smart Home and its network edge [18]	39
Figure 2.15: The Cyber-Trust MUD environment.....	39
Figure 2.16: Results of MUD analysis on IoT devices [65]	40
Figure 3.1: Malware trend (Source: [112])	43
Figure 3.2: Ways that a virus can add itself to the host code	44
Figure 3.3: Malware Analysis Method.....	47
Figure 3.4: Example of sandbox architecture	49
Figure 3.5: Image containing Signature of Worm	50
Figure 3.6: signature-based system.....	50
Figure 3.7: Snort Engine [82]	51
Figure 3.8: Suricata architecture [83].....	52
Figure 3.9: Malware Detection Techniques.....	53
Figure 3.10: Bro IDS working	54
Figure 3.11: Visual representation of the relationship between data-related fields.....	56
Figure 3.12: Phases of Malware for development of stealth methodologies [99]	57
Figure 3.13: Malware detector	60
Figure 3.14: Malware Normalization and Signature Comparison [70].....	61
Figure 4.1: The framework of privacy-preserving data mining [95].....	64
Figure 4.2: Summary of the critical aspects of IoT privacy [124]	65
Figure 4.3: A classification of the developed privacy preserving data mining algorithms [45]	67
Figure 4.4: Somewhat Homomorphic Encryption [92].....	72
Figure 4.5: Fully Homomorphic Encryption (FHE) [92].....	73
Figure 4.6: The fragile balance among data privacy and data utility (source [68])	75
Figure 4.7: The mechanism of PPDM	75
Figure 4.8: Classification hierarchy of PPDM techniques based on the location of the computation.....	76

List of Tables

Table 2.1: Correlations between Services and Service Categories.....	23
Table 2.2: The six principle areas are, including examples of applications.	33
Table 3.1: Tools that plays a vital role for performing dynamic behaviour analysis	49
Table 3.2: Bro IDS	54
Table 3.3: Structure of an encrypted virus [13].....	57
Table 3.4: Major evasion techniques	58

Executive Summary

This report is a contractual deliverable within the Horizon 2020 Project Cyber-Trust: Advanced Cyber-Threat Intelligence, Detection, and Mitigation Platform for a Trusted Internet of Things. It provides the state-of-the-art in the areas covered by work package WP6 of Cyber-Trust project, which include IoT devices profiling methods, malware detection and mitigation techniques, as well as the quest for privacy in the IoT. First, the report explores different technologies and approaches for profiling IoT devices in diverse platforms and services. In brief, IoT device profiling methods vary based on the platform and the applications they run and the services they offer. Also, the report provides current techniques and approaches for malware detection and mitigation. Malware typically includes viruses, worms, Trojans, bots, ransomware, and rootkits. To detect such malware, there are two main types of malware detection methods are discussed in this deliverable; these include mainly signature and anomaly-based techniques. The first compare software's signatures against existing database repository that hold a collection of pre-defined malware signatures. While the anomaly-based, the behaviour of the software/device (or even malware) is monitored against the defined set of requirements and against security policy which is a baseline model for normal behaviour of the system. The deliverable also provides the quest for privacy in the IoT as such devices generate a large volume of data in which there is a need for privacy-preserving, especially with data mining process to prevent any sensitive leakage of the confidential data.

Further, existing heuristic-based techniques and tools are presented, for instance, data perturbation, cryptographic techniques, blocking based techniques, hybrid and data anonymization schemes. Finally, it should be noted that this deliverable discusses content in which some of them are technical by nature (e.g. network profiling, techniques, malware detection mitigation methods). We believe readers with technical knowledge (such as CIOs/IOs, network/security experts, IT department staff with at least some entry-level security expertise, personnel of LEA, ISAO and ISAC) will be able to benefit from the full extent of this deliverable. Non-technical readers might have to skip the technical parts of the content (especially during their first reading).

1. Introduction

Internet of Things (IoT) consists of heterogeneous internet-based devices, which generates an enormous volume of data, this include sensors, smart devices and other industrialised modules. IoT also introduces laudable, presents an exponential increase in the complexity of the network, and in terms of cybersecurity, it creates a more vulnerable topology because of the increased complexity, principally due to security problems arising from embedded devices and other legacy hardware. Further, with the emerging of IoT technologies, malware and criminals can target such devices by exploiting the underlying services and existing vulnerabilities in which this introduces number of security risks. This vulnerability challenge is what Cyber-Trust aims to investigate and address, to both support the growth of IoT while mitigating the effects of complexity and vulnerability when protecting IoT devices. Therefore, profiling such devices for identifying potential misbehaviour or infection by malware is critical. In addition, IoT generates an enormous volume of data and mining data from those devices requires a safe transition such information via an untrusted network (i.e. Internet), in which user data becomes venerable to a variety of potential attacks. Therefore, privacy-preserving becomes a vital requirement in such activity, especially with the existing threats and the increasing number of malware targeting such devices. Hence, this report offers an overview and state-of-the-art in those areas.

1.1 Purpose of the document

The Cyber-Trust project aims to develop an innovative cyber-threat intelligence gathering, detection, and mitigation platform, tackle the grand challenges towards securing the ecosystem of IoT devices. This document aims to deliver State-of-the-art on profiling, detection and mitigation regarding advanced Cyber-Threat Intelligence, Detection, and Mitigation Platform for a Trusted Internet of Things project.

1.2 Relations with other activities in the project

The review of IoT profiling, those malware detections and mitigation and privacy issues in data mining are directly related to WP5 and WP6. It also highlights different data-related issues and threats within the Cyber-Trust project; these include identifying aspects that need to be taken care of when deploying Cyber-Trust competent and agents. For example, it explores potential data mining issues and challenges as well as cyber threats posed by malware that exist in some of the D2.3 Cyber-Trust use case scenarios. It also, intersect with the areas covered by work package WP5 of Cyber-Trust project, which include cyber-threat intelligence (CTI) gathering and sharing techniques, trust establishment and risk assessment, as well as game-theoretic security in which that the gathered and shared data should be privacy-preserved.

1.3 Structure of the document

This document is organized into four main sections, including the current introduction (section **Error! Reference source not found.**) and references, in order to achieve the abovementioned aim. More precisely, the rest of the document is structured as follows:

- Section 2 demonstrates IoT devices profiling methods. This section shows SDA operating on Linux-based distribution such as power consumption, performance profiling, and memory usage. This part follows by SDA App and with cloud service. The next part of this section is about IoT Connections in terms of short-range wireless such as Bluetooth mesh networking and Bluetooth low energy, ZigBee, Z-Wave, Wireless Fidelity (Wi-Fi), IPv6 Low-power wireless Personal Area Network (6LowPAN), Thread, and Light Fidelity (Li-Fi). This part is followed by demonstration Cellular Technologies such as Extended Coverage-GSM-IoT (EC-GSM-IoT), Narrow-Band IoT (NB-IoT), LTE-Machine Type Communication (LTE-MTC) and enhanced Machine Type Communication (eMTC). The next part of this section is to show Long-range Wireless Technologies, for instance, LoRa (LoRa PHY and

LoRaWAN), and Sigfox, Weightless. The last part is about Log files and Network profiling, such as protocols, ports, volume Traffic & Muddy Filtering. This section finally concludes.

- Section 3 presents the state-of-the-art of malware detection and mitigation. This section explains the classification of malware such as viruses, worms, Trojans, backdoor and Ransomware. Then, Malware Analysis Techniques mainly, basic static analysis, advance static analysis, and basic dynamic analysis such as the virtual box, process monitor, process explorer, ApateDNS, Wireshark, Sandboxes, and finally advanced dynamic analysis. The final part of this section is about evasion of malware and anti-evasion approaches.
- The last section presents the quest for privacy in the IoT which introduces the need for privacy-preserving data mining, heuristic-based techniques and tools, cryptography-Based Techniques and tools, and homomorphic encryption Techniques, such as homomorphic encryption, somewhat homomorphic encryption, fully homomorphic encryption, and limitations and generations.

2. IoT Devices profiling methods

2.1 Introduction

Cyber-Trust project aims to combat potential threats posed by the adoption of IoT by building a proactive cyber-threat intelligence gathering and sharing platform. As the services being offered via IoT platforms are becoming highly pervasive, ubiquitous, and distributed, any concerns about our society's security are amplified due to the appearance of new forms of sophisticated threats and cyber-attacks. However, IoT devices are essentially resource-constrained in terms of computation, battery power, intermittent connectivity, and network protocols. In which, achieving the Cyber-Trust ultimate goal requires investigating and developing optimal solutions compatible with such limitations and constraints that exist in IoT. Typically, IoT devices operate and utilise different computing services and protocols; these include cloud platforms, network protocols, customised operating systems and wireless connections technologies. Thereby, profiling and monitoring such devices' behaviour and security require incorporating various layers and approaches to attribute and profile malicious activities and threats through the developing of the device and network profiling services that the Cyber-Trust ecosystem aims to facilitate. The rest of this chapter explores and discusses existing IoT devices profiling methods including SDA and cloud services, IoT connections and network profiling.

2.2 Device Profiling

The estimated usage share of operating systems in personal computing and smartphones area is considered separate from the area of desktop computing. In the personal computing platform area of smartphones and watches two systems dominate: Google's Android with around 86% of the market share and Apple's iOS with around 14% [113], while in the area of desktop and laptop computers, Microsoft Windows is generally above 75% in most markets, Apple's macOS at around 13%, the remaining share is spread between Google's ChromeOS and Linux [41]. All these figures vary somewhat in different markets and depending on how they are gathered.

The Smart Device Agents (SDA) and Smart Gateway Agents (SGA) are the two Cyber-Trust components responsible for the acquisition of information from the end user IoT devices and gateways respectively and represent the links with the Cyber-Trust core components hosted on the service provider layer. Monitoring of the end user's gateway is by default inactive as it enables active monitoring for all connected devices and the need to transfer exchanged traffic to the Cyber-Trust backend for Deep Packet Inspection (DPI); the user may enable/disable this option through their profile at any time after they have clearly consented. In contrast to SGA, SDA operates in a more restrictive manner as its purpose is to receive information regarding new vulnerabilities and modes of operation from the Profiling Service and to communicate back in the occurrence of a suspicious event.

As a form of data minimisation and a measure of gathering only data that serve a legitimate purpose, the SDA exhibits intelligence and performs real-time monitoring. Different flavours of SDA will be implemented within the framework of Cyber-Trust as a range of smart devices need to be accommodated with mainly two modes of operation: one being for continuous/ real-time operation and the latter for ad-hoc operation when the circumstances call for it.

The SDA is responsible primarily for the monitoring of device's usage, critical files, security status (patching status, firmware integrity, vulnerability risk) as well as suspicious network transactions, and secondly for the application of mitigation policies and remediation actions after the detection of an attack or threat that could endanger the integrity and operation of the monitored device. Due to its intended operation, the SDA is designed to check whether the hosting device performs as intended by its manufacturer, ensures that critical OS files are uncompromised and that only secure means of communication are used. Data regularly synched with the Profiling Service involve information regarding runtime processes and used hardware resources.

Only in the case of identified suspicious traffic and activity, network packages are signed by SDA and communicated with the CT Cyber Defense service for further investigation.

Data from the SDA and SGA are communicated to the Profiling Service (PS), responsible for the storage and management of Cyber-Trust generated and acquired data. In particular, two separate access control layers are supported: one for defining architectural policies and one for controlling runtime operations for matching use preferences.

According to market research [36], the IoT device market is expected to further grow, with billions of devices connected to the internet in the near future. As such, the IoT device market represents a huge opportunity for device manufacturers. IoT devices are typically shipped in very high volumes, due to this, most IoT devices follow a simple design to prevent any defects from occurring due to hardware malfunction. A typical IoT device structure consists of a microcontroller (MCU), a Power Management Integrated Circuit (PMIC), a Radio Frequency Integrated Circuit (RFIC), various metering or detection sensors and small batteries such as coin batteries to perform metering or monitoring tasks and communicate with edge or cloud services. Since IoT devices are typically used without power lines, the battery can seldom be re-charged or replaced. In order to minimize current consumption, IoT devices therefore typically only maintain active mode for brief periods of time, mostly operating in a sleep mode.

Three different SDA implementations will be developed within Cyber-Trust to accommodate the following classes of IoT devices:

- Devices running a Linux-based OS distribution
- Apps to run on smart TVs and smartphones
- Devices implemented to use IoT cloud Services

Generally, profiling refers to recording and analyzing data to characterize personal behaviour to assess or conclude their personal interests in a specific domain or for differentiation objectives [102]. Off-the-shelf data mining tools can depict a full image of the customer requirements and easily offer a thorough customer profile. Following the rule “know your customer” [130, 4], in e-commerce, online profiling is a key tool for companies to better comprehend their customer wishes. Profiling data is progressively utilised for target advertisements, Web sites personalization, and service matching. However, profiling leads to privacy damage when employed to learn a lot about a user such as political and religious views, sexual orientation, and/or medical conditions [133, 53, 54], priceless information that can be collected, shared, and sold without even a customer’s consent [8, 37, 14, 78].

The growing of Internet-connected systems and the evolution of data mining algorithms and tools considerably participated in the emergence of big data [150]. From IoT and big data perspectives, the argument is that limiting access to private/personal data negatively influences the precision of the data mining exercise. In addition to this conflict of interest between privacy and profiling, it has been noticed that identification and tracking threats further aggravate the potentials for profiling and increase the risks of privacy leakage by data hunting black markets.

There are three potential techniques of monitoring and profiling that provide grounds for differentiation in IoT systems:

1. Data collection that leads to conclusions about the user, for instance, Internet browsing behaviour.
2. Profiling at large through linking IoT datasets (sometimes called ‘sensor fusion’).
3. Profiling that occurs when data is shared with third parties that combine data with other datasets such as employers and insurers.

Users can have access to an unprecedented number of personalized services, all of which would offer considerable data, and the environment itself would be able to obtain information about users automatically. Random, invasive profiling and inferential analytics can result from data sharing, in particular when several IoT devices offer data that is connected to a single user identity [101].

Identification technologies permit precisely this type of linkage. By linking multiple devices and the data they produce to a single user identity, the usage of a device or service can be personalized, based upon previous behaviours and preferences, and inferences drawn from these data [37]. Privacy risks of linkage between datasets become particularly critical when central authentication systems (e.g. SSO) or identity stores have access to data that authenticated devices generate. While possibly providing better user experience, linkage and personalization across several IoT devices and services nevertheless pose risks to user privacy. Data controllers can draw inferences about the user unrelated to the planned operation of the devices and services [78]. Device identification can be employed to link together a user's behaviours, even if each of the datasets is individually handled responsibly and properly de-identified. Algorithms can be utilised to update user profiles continuously to predict their behaviour and match their preferences [169].

Part of the challenge of controlling profiling is the uncertain value of data that sensors create. This has been described as 'sensor fusion' [154]. In addition to the stream of data collected, the possible inferences drawn can be broader if combined with other data categories from the IoT. For instance, Fitbit opens insight into the user's health status such as heart rate, as well as into user's movement such as geolocation data and steps taken per day [110]. Linking these two data streams could lead to further privacy offensive inferences [163]. Similarly, it has been argued that existing smartphone sensors can be used to infer a user's mood; stress levels; personality type; bipolar disorder; demographics (e.g., gender, marital status, job status, age); smoking habits; overall well-being; progression of Parkinson's disease; sleep patterns; happiness; levels of exercise; and types of physical activity or movement [150, 169, 163, 110].

Eskens have made a similar point about the 'Nest' brand thermostat, which "collects data such as current temperature, humidity, ambient light, and whether something in the room is moving" [162]. Depending on this data, which is gathered to adjust the temperature automatically, inferences can be made about the presence and specific location of occupants in a home, their current state such as asleep or awake, and other aspects of home activity [162]. These examples illustrate that smart devices need to gather data and make inferences (for instance, is the person at home? or does the temperature need to be adjusted?) in order to work appropriately [161], but this can simultaneously and inadvertently lead to interventions into the privacy of the customers.

While some inferences and profiling drawn from IoT can be benign – for instance when data is utilised to provide a more personalized user experience – they can also cause unfair differentiation such as economic or gender-based [163]. The likely for discrimination holds true even when using non-sensitive data categories, from which sensitive information can still be concluded [12]. Third parties with access to IoT data connected to an identified target can use this data for aims with which the user would not approve if asked. For instance, Fitbit data could be relevant to prospective employers, who could make inferences about impulsivity and the inability to delay gratification-both of which might be inferred from one's exercise habits-correlate with alcohol and drug abuse, disordered eating behaviour, cigarette smoking, higher credit-card debt, and lower credit scores. Lack of sleep-which a Fitbit tracks-has been linked to poor psychological well-being, health problems, poor cognitive performance, and negative emotions such as anger, depression, sadness, and fear [163].

Employers are not the only third parties possibly interested in this data. It has been clarified that data controllers progressively use the IoT for "monitoring people's online behaviour and using the information collected to show people individually targeted advertisements" [165]. The researchers forebode that a lack of awareness of such methods can be discriminating to customers; therefore, an increased level of transparency is needed. It has similarly been maintained that opaque profiling and automated decision-making in advertisements can represent a threat to diversity. Furthermore, even the neutral data (e.g. postcodes) can lead to inference and discrimination based on ethnicity, gender, or sexual preference especially when datasets are linked [160].

Weaknesses of anonymisation to prevent profiling and resulting discrimination lead to further problems. According to Gudgel, "there is special concern that if data is not anonymized then it could potentially be used to track specific individuals, linked to information in other databases, and possibly used to predict future behaviour" [80]. Chasing data of the type that many IoT devices create is notorious for opening to re-

identification and reverse engineering of identity. Following the assumption that data cannot always be anonymised without destroying its analytical value, non-technical ways may be necessary to inhibit profiling and discrimination in the IoT [119]. One possible solution is to treat all IoT created data that refers to a user as personal data under data protection law, as it will continuously be potential in principle to link the data back to a user. This method would ensure that the user would be able to use his/her rights granted under data protection law overall information that IoT devices generate and control. This concept would not avoid profiling as a result; but rather, lengthen the scope of existing user rights against privacy risks to cover all data, including inferences and profiles.

These problems will be expanded and magnified by the proliferation of machine learning in the IoT [163]. Machine learning will cause even less expected inferences, while the complexity and opaqueness of machine learning algorithms can unintentionally hide discriminatory treatment from customers [160]. Systems operating as ‘black boxes’, for which the inputs, internal logic, and outputs may be unavailable or incomprehensible to specific users do not facilitate systematic observation, identification of harmful effects, or investigation of their causes [161, 12]. Machine learning can inadvertently and unknowingly reinforce existing biases and prejudices as a result [80].

European legislators have specified the hazards of profiling and discrimination, although often lacking comprehensive recommendations [119]. European regulators have highlighted that the major concerns in declarations on profiling, acknowledging that profiling offers grounds for discrimination, particularly when datasets are joint. The European Commission has called for the creation of a set of guiding standards to manage IoT regulation, urging that always being linked to the things around us can cause more observation or more profiling by public authorities and private entities. Correspondingly, the European Data Protection Supervisor has raised concerns that RFID tags employed in IoT systems might cause profiling by linking customers to devices and usage records [66].

Similar concerns are reproduced in the General Data Protection Regulation (GDPR), particularly in Article 21 (Right to object) and Article 22 (Automated individual decision-making, including profiling). Article 21 presents the right of data subjects to object to data processing, including profiling, at any time. If the aim of data processing is direct marketing, the data subject will have an absolute right to object. In all other cases, data processing must end unless the data controller can prove compelling legitimate interests that override the interests of the data subjects. Regrettably, the framework does not define compelling interests of data controllers [111], leaving both data controllers and data subjects in an uncertain state. On top of this uncertainty, the technical feasibility of stopping data collection is also challenging. How data controllers can handle objections beyond stopping all service provision remains unclear. Consequently, users worried about their privacy or IoT-facilitated profiling might be left with a binary ‘take it or leave it’ choice.

Article 22 presents further protections against automated decision-making, including profiling, but just when data processing is only automated and has legal or similar significant influences. The scope of applicability is thus likely to be very limited, at least while these terms (‘solely automated’, ‘legal or similarly significant effects’) stay undefined in practice [160]. In cases where such decision-making and profiling are essential for entering or accomplishing a contract between data subject and the data controller, or grounded on explicit consent (Article 22 (2) (a) and (c)), data subjects are granted rights to acquire human involvement on the part of the controller, to express a perspective and to contest the decision (Article 22 (3)). If automated decision-making, including profiling, has major impacts on a data subject, individuals will possess a legal remedy if they upset with the outcome. Lastly, at first sight, Article 11 in the GDPR seems to be beneficial. This provision echoes the idea of only identifying data subjects for as long as necessary. However, as stated above, discrimination is also possible through extraneous additional, non-personal or anonymous data. In those cases, data protection law either does not apply or offers insufficient protection. With a broader and well-defined scope of applicability, these rights would provide a very promising method for data subjects to maintain some control over how the data is employed to personalize services and future opportunities.

2.3 SDA with Cloud Services

The computational power required to implement such IoT devices is quite small by modern standards. Moreover, so, they have been developed with simple processor technologies — mostly using ARM's Cortex M architecture and use an equally simple Operating System and software stack [171].

There are various types of Operating Systems, and pretty much every single computer, embedded or not, will have an operating system controlling it. IoT devices tend to use a type called RTOS, which officially is short for Real-Time Operating System (aka Not-a-Full-Featured Operating System). The main draw for using an RTOS is its simplicity and its modest requirements of resources for itself, as the selection of features required from the operating system is done at the time the image is being built, so as the consumer only pays for what the device users in terms of computational resources. Using such a lightweight operating system allows designers to design smaller, cheaper, and less power-hungry embedded computers for their IoT devices.

Such devices do not allow or may tolerate third-party services to be accommodated onboard the IoT device itself. Due to this, monitoring of such devices will be performed based on the exposed API of the IoT cloud network onto which it is built for communication with backend cloud services. There are at least 49 IoT cloud platforms exist in today's global market to meet the requirements of different user and application groups such as enterprises, government, farmer, healthcare, communication, transportation, and manufacturing [121]. Nevertheless, lack of overall knowledge about these IoT cloud platforms restricts researchers and enthusiasts to choose a particular cloud when they are in phase with the development of any product or solution utilising IoT enabled technologies. Several articles [122, 123] are found that develop and apply IoT solutions based on the existing clouds that are a matter of study in this paper. Strong need for integration of cloud and IoT is mentioned in [153] where an agent-oriented and cloud-assisted paradigm is envisaged based on a novel reference architecture [121].

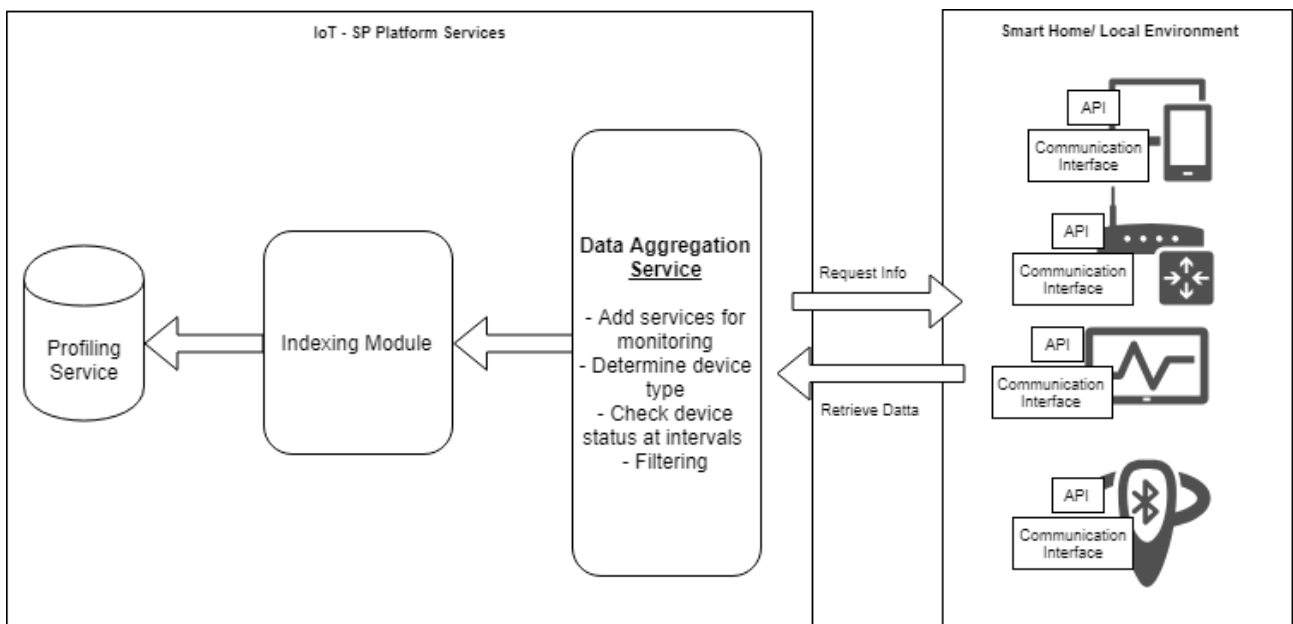


Figure 2.1: SDA Monitoring through Cloud Networks

Considering the need for the accommodation of as many platforms as possible, Cyber-Trust will implement a generalized solution for the integration of web-services provided by various IoT cloud network providers as shown in Figure 2.1. Another alternative to this approach is shown Figure 2.2, where an embedded system installed in the local environment of the end-user will be responsible for regularly gathering as much information as possible through the supported and already established communication channels. An example IoT cloud network is Tuya for which an easy-to-use open source API for devices that use Tuya's cloud services exist [172].

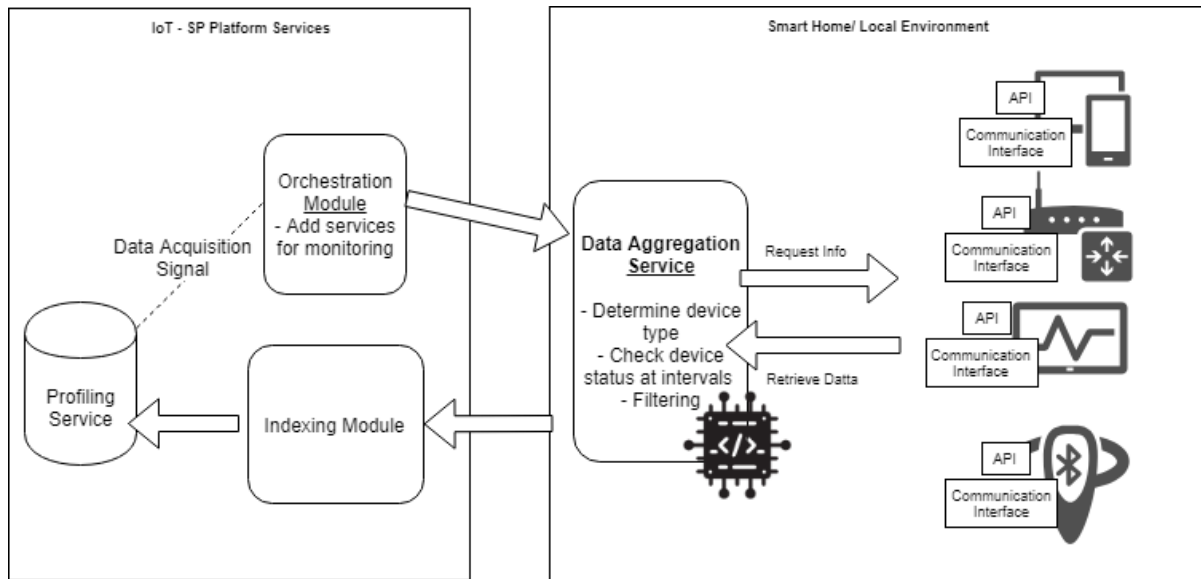


Figure 2.2: SDA Hybrid Monitoring through Cloud Networks

2.4 SDA operating on Linux-based distribution

For more powerful IoT devices equipped with a Linux- based distribution that allows access to the root OS, a different SDA flavour will be implemented in Cyber-Trust. A number of IoT devices bear lightweight Linux based distributions, and these devices are probably the ones providing most liberty in the monitoring and detection of various events. Such devices include embedded systems such as raspberry PIs, TV boxes etc.

For this class of IoT devices, the end user will be provided with an executable file to be installed on the end device of interest. Once the SDA agent is installed on the end device, the agent is paired with the profiling service and the SDA may be configured remotely (see Figure 2.3).

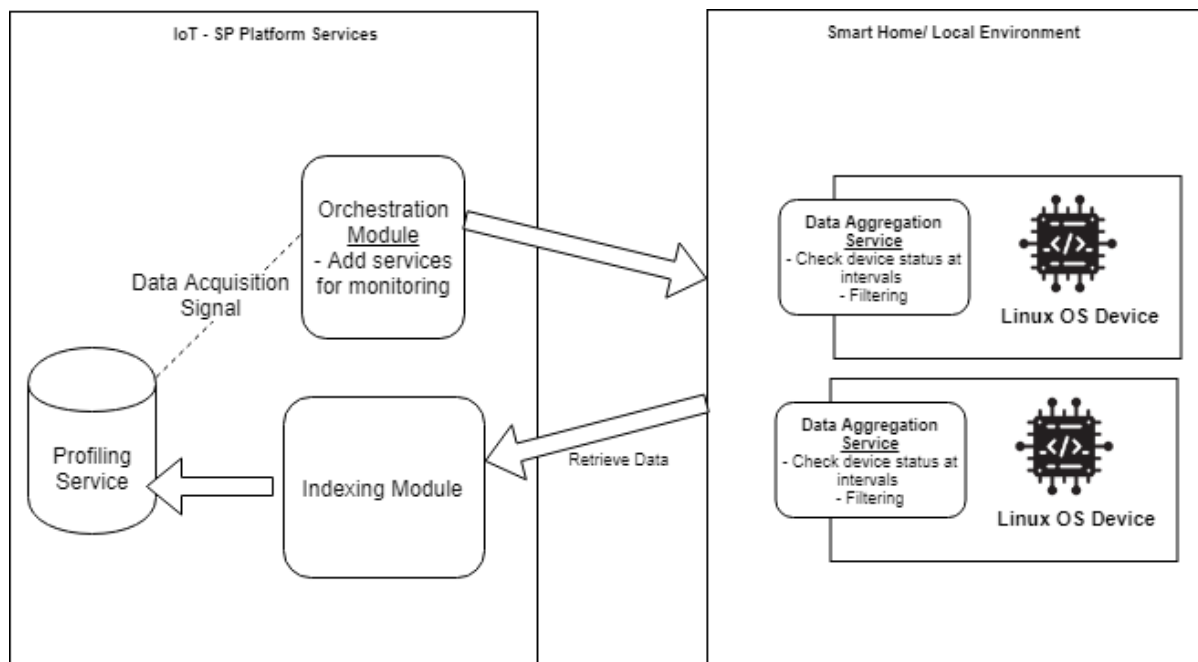


Figure 2.3: SDA operating on Linux Based Device

2.5 SDA App

Mobile app development since a few years ago followed a traditional development workflow where apps were developed in a programming language native to the device and operating system. This approach entitled that in case a mobile app needed to support multiple platforms (i.e. Android, iOS), the development process needed to be repeated for each platform separately. Following this approach, no code could be shared between app versions and as a result, the development time was significantly increased. Due to this and the need for the majority of apps to support multiple platforms, the concepts of cross-platform and hybrid development are introduced.

Cross-platform mobile apps are developed using an intermediate language, such as JavaScript, that is not native to the device's operating system. This means that some, or all, of this code can be shared across target platforms – for instance, across both iOS and Android. Cross-platform apps are different to HTML5 hybrid apps as hybrid apps usually incorporate a mix of native app and mobile app concepts; making them more powerful and flexible compared to apps developed with a cross platforms framework such as Xamarin [22], or PhoneGap [132]. HTML5 hybrid mobile apps are cross-platform apps but render the user interface using an embedded web browser, leveraging HTML, CSS and JavaScript. As a result, building the core part of the app using web technologies allows for faster development and greater flexibility, as the core of an app may be compiled for different platforms (i.e. Android, iOS) and device types (i.e. smartphones, tablets, smart TVs).

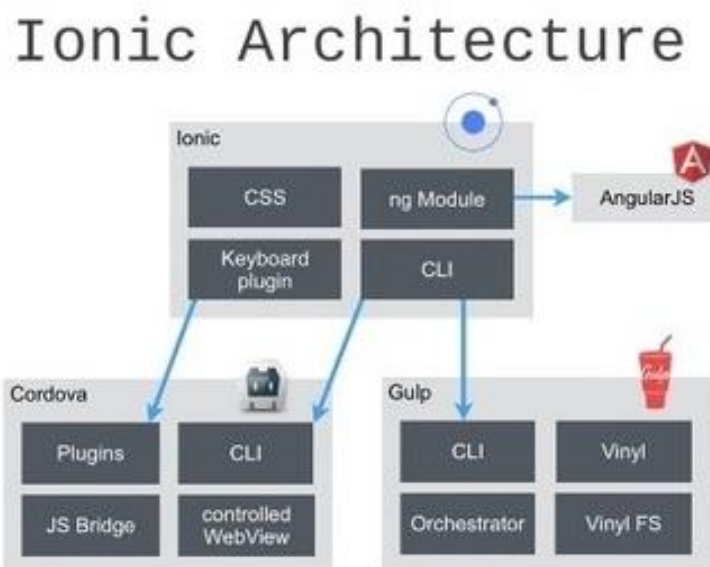


Figure 2.4: Hybrid Mobile app development

Considering the above-mentioned methodologies and the fact that the SDA for Cyber-Trust needs to be as flexible and versatile as possible, the Ionic hybrid framework will be utilised during development. Ionic is a complete open-source SDK for hybrid mobile app development created in 2013. The original version was released in 2013 and built on top of AngularJS and Apache Cordova, while the more recent releases, known as Ionic 3 or simply "Ionic", are built on Angular [23] (see Figure 2.4).

For device monitoring, ionic provides a number of plugins that allow the monitoring of runtime processes, CPU and memory usage as well as a number of libraries for the monitoring of various other aspects of the hosting device.

- **Monitoring of runtime processes:** Ionic DevApp extends the capabilities of Ionic Framework, making it easy to test apps directly on devices. DevApp offers a real-time view of changes as they are being made, with a rich library of pre-installed native plugins to test native features of the app.

- **CPU usage:** The “chrome.system.cpu” plugin provides the ability to query basic CPU information of the system [51]. The exposed API of this plugin allows getting information regarding the architecture, type, model and features of the processing unit as well as runtime properties of each core such as cumulative usage info for this logical processor (user, kernel, idle and total). For some devices, the plugin may also provide temperature information.
- **Memory Usage:** depending on the hosting platform, a number of libraries are available for the monitoring of an apps memory usage (i.e. safari dev tools and chrome dev tools), however, the “chrome.system.memory” cordova plugin [32] provides the ability to get physical memory information regarding the memory capacity of the device and the available memory capacity.
- **Monitor network traffic:** the **cordova-plugin-network-information** plugin may be used to gather information regarding the device's cellular and WiFi connection, and whether the device has an internet connection. Additionally, through this plugin, the SDA may listen to offline and online events check the connection type, get triggered when the device goes offline etc.

In contrast to smartphones, there is no one standard ensuring compatibility with Smart TVs as these devices are hugely affected by their different specs, resolutions, processing power and OS versions making most Smart TV apps incompatible with most platforms. However, the choice of the Ionic Framework seems promising in this scenario as it is the most widely used technology used for Smart TV app development is the combination of HTML/ CSS and JavaScript.

2.6 Services

Services [52] are software possess protocols and technologies that enable devices to extend and enhance their functionalities. For example, such functionalities are a) configuration of information infrastructure, the protection of device towards to malicious attacks, secure transfer files, allow computers to download files and make them available to other users on the network etc. Ideally, users would like to execute every Service for dissemination, mitigation, encryption etc. in every device and Operational System (OS), but by the time this is not feasible.

2.6.1 Computer Services

Simple Network Management Protocol (SNMP) is a protocol for:

- remote devices' configuration,
- monitor network performance,
- audit network usage,
- discover of network faults,
- detect inappropriate access,
- gathering and organising the information

SNMP also possess management applications and services, such as SNMP Management API and WinSNMP API. The former has a set of functions that can be used to the rapid development of basic SNMP management systems. Moreover, the latter supports a set of functions for encoding, decoding, sending and receiving SNMP messages. Encoding and decoding are applicable in data communications, networking, and storage especially radio communications systems.

Microsoft Server Message Block (SMB) [53] is an application that is installed by default in Microsoft Windows Server and implemented by the network file sharing SMB protocol. The third version of SMB (SMBv3) protocol has the SMB Encryption feature, but it is not configured by default.

Google Cloud DNS [35] composed by a scalable, reliable and managed authoritative Domain Name System (DNS) service running on the same infrastructure as Google. Cloud DNS enables the translation requests to

domain names into IP addresses. Also, it is a flexible and programmable tool, it could easily publish and manage millions of DNS zones and records using our simple user interface, command-line interface or API.

App Net Manager is a secure web-based portal that is used by users in order to create, configure, modify, delete, and monitor the components of the network. It is also, available through the Compute Classic web console [54].

2.6.2 Router Services

Router's Services possess software and hardware solutions that allow devices to have capabilities regarding device and network activities. Such software capabilities are regarding to intrusions prevention, content security, monitoring, network data transferring, mitigation actions. Moreover, router's Services enables devices get visibility into and control over activity across your network. Delivering network information enhance threat defence of the device and of the entire network as well.

Services such as IPSec, SSL and VPN are protocols which safeguard information from cyber-attacks (e.g. malware etc.). More analytic, IPSec and SSL protocols are both designed to secure data in transit through encryption. Also, VPN encryption prevents third parties from reading user's data as it passes through the internet.

Trust Management Systems (TMS) applications that enables users to utilize an integrated, centralized and easy security management. Some of the domain systems that manage are firewalls, application control, intrusion prevention, URL filtering and protection towards malicious attacks.

Simple Network Management Protocol (SNMP) applications, such as SNMP Management API and WinSNMP API applied to routers for monitor the actions of devices, such as, the volume and the frequency of data etc. Most of the times, routers possess Network Address Translation (NAT) protocols, which allows all devices on a subnetwork, such all the devices in a store, to share the same public IP address.

Although emphasizing in software solutions there are mature hardware market solutions. Hardware solutions either referring to hardware solutions applied to routers or are hardware itself, are sophisticated solutions in order to beat complex cyber-attacks, such as, Cisco Series for firewall solutions, HP Networking Routers etc.

2.6.3 Camera Services

An IP camera (Internet Protocol camera), is a device that has access to network and have the capability to transfer data via Internet. In order to interconnect IP camera to cloud for storing data (e.g. photos and videos), network protocols are provided. Such protocols are TCP/IP and UDP, FTP.

User Datagram Protocol (UDP) is a connectionless protocol and provides rapid transmission of data. For this reason, sometimes keeps real-time data ignoring data confirmation and packet loss.

Transmission Control Protocol (TCP) enables data to be transmitted with integrity confirmation.

FTP server is another mechanism for image transferring towards to internet, it is based on File transfer protocol (FTP).

Routing Information Protocol (RIP) based on the UDP, is used in order to implement mechanisms for prevent incorrect routing information from being propagated.

Cloud DNS is an application based on Domain Name System (DNS) and UDP protocols. Cloud DNS enables the translation requests to domain names into IP addresses (e.g. Google Cloud DNS etc.) [56].

2.6.4 Smartphone Services & Tablet Services

Mobile Device Managers Plus (MDMP), is a suggested solution for the smartphone's Device Management. Generally, smartphones, tablets do not support Services for Device Management. For this reason, a suggested solution is the upload of the Mobile Device Manager Plus. MDMP enables system administrators

to manage mobile Android devices running version 4.0 and above. This Android mobile device management (MDM) software empowers administrators to monitor, manage, audit, and secure corporate data on those devices. It also offers advanced controls and enhanced Android MDM capabilities.

Android.Opfake provided by Symantec is a mature application that detects Trojan horses on the Android platform and sends SMS texts to premium-rate numbers [56].

Java.Opfake provided by Symantec is a mature application that detects Trojan horses on mobile devices and sends SMS texts to premium-rate numbers [57].

WatchHound Cell Phone Security Monitor continuously scans any mobile phone prohibited area for wireless activity essentially creating wireless-free zones without the need for jamming. All incoming and outgoing cellular calls are detected (in active or standby mode) and time stamped for later analysis.

2.6.5 Gateway Services

Gateway is a node in a computer network, and it is responsible for data sharing through networks. For this reason, security mechanisms are provided to gateway in order to enhance data safety and safeguard sensitive information of the device. Most of the times gateways behave as routers, since they manage packets of data and control information paths. Most of the router services are applied to gateway services. Gateways services are responsible for data sharing, accessing in different networks, download files from internet etc

Exterior Gateway Protocol (EGP) is used to exchange routing information between autonomous systems. It is one of the main protocols for data communication across the internet.

Border Gateway Protocol (BGP) is a standardized exterior gateway protocol. The BGP service provides routing decisions in the internet based on network policies, rules that implemented by network administrator. In this way they mitigate network threats and malicious attackers.

2.6.6 Categories Services

The classification of Services into categories is not a simplistic procedure, since services are constructed, for different domain-utilities, for various frameworks and various levels of processes (high, moderate, low). The types of Services that are described are characterized by discrete and low-level processes. The subcategories of Services that identified are a) File transfer services, b) Encryption/Decryption services, c) Monitoring services, d) Data format transformation services, e) Mitigation Threats Services.

Table 2.1: Correlations between Services and Service Categories

	File transfer services	Encryption/Decryption services	Monitoring services	Data format transformation services	Mitigation Threats Services
SNMP Management API			x		
WinSNMP API			x	x	
SMB	x	x			
Google Cloud DNS			x	x	
App Net Manager			x		
IPsec &SSL		x	x		x
VPN		x			
TMS			x		x
UDP	x				
TCP/IP	x				
FTP server	x				
RIP	x		x		
MDMP			x		x
Android.OpFake					x

Java.OpFake					x
WatchHound			x		x
EGP	x			x	
BGP			x	x	x

As indicated in Table 2.1: the majority of the Services that are described have capabilities in order to mitigate cyber threats and malicious attacks towards devices as well as, they also have capabilities in order to monitor the devices' conditions (e.g. traffic, abnormalities etc.).

2.7 IoT Connections

2.7.1 Short-range wireless

2.7.1.1 Bluetooth mesh networking and Bluetooth low energy

Bluetooth mesh networking is a protocol based upon Bluetooth Low Energy that allows for many-to-many communication over Bluetooth radio. Bluetooth Low Energy (Bluetooth LE, or BLE, formerly marketed as Bluetooth Smart) is based on mesh networking principles. The mesh network operation is designed to:

- enable messages to be sent from one element to one or more elements;
- allow messages to be relayed via other nodes to extend the range of communication;
- secure messages against known security attacks, including eavesdropping attacks, man-in-the-middle attacks, replay attacks, trash-can attacks, brute-force key attacks, and possible additional security attacks not documented here;
- work on existing devices in the market today;
- deliver messages in a timely manner;
- continue to work when one or more devices are moved or stop operating; and
- have built-in forward compatibility to support future versions of the Mesh Profile specification

BLE technology operates in the same spectrum range (2.400–2.4835 GHz ISM band) as classic Bluetooth technology but uses a different set of channels. It can serve IoT devices in cells with a radius over 100m and supports data rates in the range 1 – 2 Mbps.

2.7.1.2 ZigBee

ZigBee, like Bluetooth, has a large installed base of operation, although perhaps traditionally more in industrial settings. ZigBee PRO and ZigBee Remote Control (RF4CE), among other available ZigBee profiles, are based on the IEEE802.15.4 protocol, which is an industry-standard wireless networking technology operating at 2.4GHz targeting applications that require relatively infrequent data exchanges at low data-rates over a restricted area and within a 100m range such as in a home or building.

ZigBee/RF4CE has some significant advantages in complex systems offering low-power operation high security, robustness and high scalability with high node counts and is well positioned to take advantage of wireless control and sensor networks in M2M and IoT applications.

The latest version of ZigBee is the recently launched 3.0, which is essentially the unification of the various ZigBee wireless standards into a single standard.

2.7.1.3 Z-wave

Z-Wave is a low-power RF communications technology that is primarily designed for home automation for products such as lamp controllers and sensors among many others. Optimized for reliable and low-latency communication of small data packets with data rates up to 100kbps, it operates in the sub-1GHz band and is impervious to interference from Wi-Fi and other wireless technologies in the 2.4-GHz range such as Bluetooth

or ZigBee. It supports full mesh networks without the need for a coordinator node and is very scalable, enabling control of up to 232 devices. Z-Wave uses a simpler protocol than some others, which can enable faster and simpler development, but the only maker of chips is Sigma Designs compared to multiple sources for other wireless technologies such as ZigBee and others.

2.7.1.4 *Wireless (Wi-Fi)*

Wi-Fi connectivity is often an obvious choice for many developers, especially given the pervasiveness of Wi-Fi within the home environment within LANs. It requires little further explanation except to state the obvious that clearly there is a wide existing infrastructure as well as offering fast data transfer and the ability to handle high quantities of data.

Currently, the most common Wi-Fi standard used in homes and many businesses is 802.11n, which offers serious throughput in the range of hundreds of Mbps, which is fine for file transfers but maybe too power-consuming for many IoT applications.

2.7.1.5 *IPv6 Low-power wireless Personal Area Network (6LowPAN)*

A key IP (Internet Protocol)-based technology is 6LowPAN. Rather than being an IoT application protocols technology like Bluetooth or ZigBee, 6LowPAN is a network protocol that defines encapsulation and header compression mechanisms.

The standard has the freedom of frequency band and physical layer and can also be used across multiple communications platforms, including Ethernet, Wi-Fi, 802.15.4 and sub-1GHz ISM. A key attribute is the Internet Protocol version 6 (IPv6) stacks, which has been a very important introduction in recent years to enable the IoT. IPv6 is the successor to IPv4 and offers approximately 5×10^{28} addresses for every person in the world, enabling any embedded object or device in the world to have its own unique IP address and connect to the Internet. Specially designed for home or building automation, IPv6 provides a basic transport mechanism to produce complex control systems and to communicate with devices cost-effectively via a low-power wireless network.

2.7.1.6 *Thread*

A very new IP-based IPv6 networking protocol aimed at the home automation environment is Thread. Based on 6LowPAN, and also like it, it is not an IoT applications protocol like Bluetooth or ZigBee. However, from an application point of view, it is primarily designed as a complement to Wi-Fi as it recognises that while Wi-Fi is good for many consumer devices that it has limitations for use in a home automation setup.

Made available in mid-2014, Thread it is a royalty-free protocol based on various standards including IEEE802.15.4 (as the wireless air-interface protocol), IPv6 and 6LoWPAN, and offers a resilient IP-based solution for the IoT. Designed to work on existing IEEE802.15.4 wireless silicon from chip vendors such as Freescale and Silicon Labs, Thread supports a mesh network using IEEE802.15.4 radio transceivers and is capable of handling up to 250 nodes with high levels of authentication and encryption. A relatively simple software upgrade should allow users to run a thread on existing IEEE802.15.4-enabled devices.

2.7.1.7 *Light Fidelity (Li-Fi)*

Li-Fi is a wireless communication technology where the devices are using light to exchange data. Li-Fi is a derivative of optical wireless communications technology using light as a medium to deliver networked, mobile, high-speed communication similarly to Wi-Fi. As of nowadays, Li-Fi remains a niche market, primarily used for technology evaluation in the IoT sphere.

2.7.2 Cellular Technologies

2.7.2.1 *Extended Coverage-GSM-IoT (EC-GSM-IoT)*

EC-GSM-IoT enables new capabilities of existing second generation (legacy) cellular networks for Low Power Wide Area (LPWA) IoT applications. EC-GSM-IoT can be activated through new software deployed over the

GSM serving areas. The benefit is the use of already deployed infrastructure (GSM network nodes) to offer extensive coverage and serve IoT devices.

2.7.2.2 *Narrow-Band IoT (NB-IoT)*

NB-IoT is a Low Power Wide Area Network (LPWAN) radio technology standard developed to enable a wide range of cellular devices and services. The NB-IoT specification is described in 3GPP Release 13 (LTE Advanced Pro). NB-IoT uses a subset of the LTE standard and as the name implies to use a fraction of LTE bandwidth (200 kHz). The Narrow-Band IoT focuses specifically on indoor coverage, serves low-cost end devices, provides a long battery life (to connected devices) while it can serve a large population of connected devices.

2.7.2.3 *LTE-Machine Type Communication (LTE-MTC) and enhanced Machine Type Communication (eMTC)*

The following standards-based family of technologies supports several LTE technology categories, such as Cat-1 and CatM1, both suitable for the IoT ecosystem.

- enhanced Machine Type Communication (eMTC)
- LTE-Machine Type Communication (LTE-MTC)

LTE-MTC is a type of Low Power Wide Area Network (LPWAN) and includes eMTC (enhanced Machine Type Communication) radio technology standard developed by 3GPP to enable a wide range of cellular devices and services (specifically, for Machine-to-machine and Internet of Things applications).

The specification for eMTC (LTE CatM1) is defined in 3GPP Release 13 (also known as LTE Advanced Pro). The advantage of LTE-M over NB-IoT is its comparably higher supported data rate, mobility, and voice over the network, despite it requires more bandwidth, which comes at an increased cost.

2.7.3 Long range Wireless Technologies

2.7.3.1 *LoRa (LoRa PHY and LoRaWAN)*

LoRa stands for Long Range and it is a digital wireless data communication technology which uses license-free sub-gigahertz radio frequency bands like 169 MHz, 433 MHz, 868 MHz (Europe) and 915 MHz (North America). LoRa enables very-long-range transmissions (above 10 km in rural areas) with low battery.

LoRaWAN defines the communication protocol and system architecture for the network, while the LoRa physical layer enables the long-range communication link. LoRaWAN is also responsible for managing the communication frequencies, data rate, and power for all devices. LoRa and LoRaWAN permit inexpensive, long-range connectivity for the Internet of Things (IoT) devices in rural, remote and offshore industries. Typical uses of LoRa products can be found in the following industries: mining, natural resource management, renewable energy, transcontinental logistics, and supply chain management. Fleet Space Technologies¹ uses LoRaWAN to provide massive connectivity to IoT sensors and devices in rural, remote and offshore areas.

2.7.3.2 *Sigfox*

An alternative long-range technology is Sigfox, which in terms of range comes between Wi-Fi and cellular. It uses the ISM bands, which are free to use without the need to acquire licenses, to transmit data over a very narrow spectrum to and from connected objects. The idea for Sigfox is that for many M2M applications that run on a small battery and only require low levels of data transfer, then Wi-Fi's range is too short while cellular is too expensive and also consumes too much power. Sigfox uses a technology called Ultra Narrow Band (UNB) and is only designed to handle low data-transfer speeds of 0,01 to 1 kbps. It consumes only 50 microwatts compared to 5000 microwatts for cellular communication or can deliver a typical stand-by time 20 years with a 2.5Ah battery while it is only 0.2 years for cellular.

¹ <https://www.fleet.space/>

Already deployed in tens of thousands of connected objects, the network is currently being rolled out in major cities across Europe. The network offers a robust, power-efficient and scalable network that can communicate with millions of battery-operated devices across areas of several square kilometres, making it suitable for various M2M applications that are expected to include smart meters, patient monitors, security devices, street lighting and environmental sensors.

2.7.3.3 *Weightless*

Weightless is a proposed proprietary open wireless technology standard for exchanging data between a base station and thousands of end user devices around it (using wavelength radio transmissions in unoccupied TV transmission channels) with high levels of security. As of 2018 weightless devices are operating in license-exempt sub-GHz frequency bands (e.g. 138 MHz, 433 MHz, 470 MHz, 780 MHz, 868MHz, 915 MHz, 923 MHz).

The defining characteristics of Weightless are the following: 100% bidirectional, fully acknowledged communication for reliability; optimized for a large number of low-complexity end devices with asynchronous uplink-dominated communication with short payload sizes (typically < 48 bytes); optimized for ultra-low-power consumption (at the expense of latency and throughput compared to cellular technologies). The weightless standard data rates vary from 0.625kbps to 100kbps. Typical End Device transmit power of 14dBm (up to 30dBm) while the Base Station transmit power of 27dBm (up to 30dBm).

A typical Weightless network is composed of the following elements:

1. End Devices (ED): the user nodes in the network which are of low-complexity, the low-cost.
2. Base Stations (BS): the central node in each cell, with which all EDs communicate via a star topology.
3. Base Station Network (BSN): interconnects all Base Stations of a single network to manage the radio resource allocation and scheduling across the network, and handle authentication, roaming and scheduling.

2.8 Log files

Log files also referred to as logs, are records of events concerning the execution of an application, the state of a system or device, and the actions of a network and its users [34]. They can be combined to construct a complete picture of a specific security event, but system administrators and automated systems should be cautious about the integrity and trustworthiness of the logged information.

The logging process may present issues with: log management and security, log content inconsistencies (e.g. timestamp inconsistencies) making the combination of logs from different sources difficult, and log file format inconsistencies.

The log management process according to NIST [84] consists of three phases:

1. **Log generation** includes management of monitoring nodes and the collection of log files from network devices.
2. **Log analysis and storage** includes the transmission, storage and analysis of the log files collected by the first phase. Log analysis includes the log file parsing process, log normalization, event filtering, event correlation and event aggregation. Log storage includes the log rotation process along with log archival and integrity checking.
3. **Log monitoring** includes the generation of alerts and reports about significant events detected by the second phase.

According to NIST [84], three log file types are of interest in the context of computer security:

1. **Security software logs**, containing security-related information; they are usually generated by security-focused applications, such as firewalls, intrusion detection or prevention systems, authentication servers etc.

2. **Operating system logs**, containing general usage information about the state and actions performed by the operating system and its users.
3. **Application logs**, containing general usage information about the execution of a specific application.

More specifically, security software logs are generated by:

- **Antimalware software**: recording malware detections, malware removals, file quarantines, signature/software updates and information about malware scans.
- **Intrusion detection and prevention systems (IDS and IPS respectively)**: recording detected attacks, suspicious behaviour and mitigation actions performed.
- **Vulnerability management software (e.g. vulnerability scanners)**: recording the patch installation history and the vulnerability status of each managed host.
- **Remote access software** (e.g. VPNs) and *authentication servers*: recording user access logs along with any successful and failed authentication attempts. Note that operating systems generate similar log files/log lines, recording both local and remote authentication attempts and their results.
- **Web proxies**: recording all URLs accessed by its users and (in some cases) cached files.
- **Routers and firewalls**: recording permitted/blocked traffic and connection attempts along with general information about the handled traffic.

Operating system logs contain two types of information:

- **System events**: recording failed events and some significant successful events; events can also be recorded as set by the system administrator.
- **Audit records**: recording successful and failed authentication attempts, account and security policy changes and use of privileges by its users.

Application logs contain:

- **Client requests and server responses**: recording event sequences and their outcomes from which the usage of an application can be monitored.
 - **Account information**: recording successful and failed authentication attempts, account and security policy changes and use of privileges by its users.
- Usage information*: recording the number and size of actions performed by the application, allowing anomaly detection to be performed.
- **Application events**: recording general application events, such as application shutdown, failure events and configuration changes.

The detail of information put in log files may vary, depending on the application and the configuration by the system administrator. Logging may be minimal, containing only the basic information about the logged event (timestamp, type of event, involved user), whereas on the other extreme detailed logging can include all parameters passed to the request, the full result body returned and possibly intermediate activities performed for serving the request. Detailed logging eases the analysis of malfunctions and security events. However it poses two major concerns: (a) it constitutes a threat for the user privacy, since personal data of the user are recorded in permanent storage and can be accessed/processed by system administrators and security officers authorized to view the log files and (b) in case of a breach, intruders that gain access to the log file will have at their disposal all recorded user personal data (sometimes including passwords) as well as detailed information about the system operation.

Regarding the storage of log files, two options can be considered:

1. **Store the log files on the device's filesystem.** This is possible for devices having adequate (considering the amount of log data to be maintained envisioned) storage space. This method is highly efficient, since persistent storage access costs are typically small, and can allow (storage space permitting) the maintenance of extensive log data. System administrators and security staff will be able to access the log files by remotely logging onto the device (e.g. via ssh) or by offloading the log files onto other servers -using, e.g. FTP/SFTP) and processing the files there. Four major considerations, however, are associated with this approach: firstly, an intruder that has successfully launched an attack against the device may have in many cases acquired adequate privileges to tamper with the log file and destroy or counterfeit evidence that will have been stored therein. Secondly, the log files are dispersed across devices and thus the potential to correlate information among log files to identify distributed attacks (e.g. network service enumeration or attempting a particular exploit across all web servers of the network) is substantially limited. Moreover, thirdly, IoT devices, being limited in many cases in terms of memory and processing resources [34], [89], may not have the potential to implement efficient alerting procedures in the case that a particular event is recorded in the log file. Moreover, fourthly, if a device malfunctions or is retired, log files on its persistent store are rendered unavailable.
2. **Store the log files in a cloud service.** RFC 3164 [93] defines the legacy protocol for communication between a device that generates log files and a *collector* device which is responsible for log storage, to transmit and receiving, correspondingly, log information. RFC 5424 [60] is the new and updated version. Both RFCs allow for using either the UDP or the TCP transport-layer protocol, with the use of UDP being predominant in the context of LANs, for efficiency purposes; additionally, both RFCs allow for specification of intermediate relays, which intervene between the log source and the log collector, arranging for log messages to be transferred on a hop-to-hop basis. Nodes also have the capability to act as both relays and collectors: a typical usage scenario would be for a node to store all messagees and forward important messages to an upstream collector, where they can be assigned secure timestamps or digitally signed to be able to be used as evidence. Finally, RFC 5424 accommodates both structured and non-structured log information, with structured information easing the task of log file analysis and aggregation. The existence or not of structure within the log messages is primarily dependent on the application, not the log server, however special filtering mechanisms can be employed either by collectors or relays to inject structure into otherwise unstructured messages, provided that some methods are available for recognizing individual elements (e.g. applications, timestamps, users) in the unstructured message.

Storing the log files on a cloud server has a number of advantages: firstly, it deprives intruders of the potential to tamper with log files, after a successful breach: while in such an event, an intruder can inhibit transmission of further log data, traces of his/her activity up to the point of the breach success will have been transmitted to the cloud server and will have been securely stored there; furthermore, it ensures availability of log data regardless of the device state (e.g. malfunctioning or retired). Secondly, it allows the exploitation of the ample storage capacities provided by server machines; thirdly, it facilitates the usage of sophisticated log analysers which can identify events of high importance and also correlate events; fourthly, cloud servers can arrange for securely timestamping, securely hashing, digitally signing, or otherwise arranging for the attestation of the validity of log timestamps and content, so that they can be used as forensic evidence; and finally, elaborate alerting mechanisms can be employed.

The cloud storage option, on the other hand, has three potential drawbacks: firstly, it entails a communication cost, which can range from moderate to high, depending on the amount of log data transmitted from the source machines to the collectors. Provided that adequate configuration options are implemented on the IoT device's logging subsystem, this issue can be tackled by sending to the cloud server the indispensable/most important log entries and keeping on the device storage the detailed entries. Then, if the cloud server detects traces of misuse, it can launch a log transfer process to obtain the detailed logs from the device.

Secondly, log data transmitted may be eavesdropped or tampered with at transmission time. As a defence measure against transmission-time eavesdropping and tampering, modern log system implementations such as *rsyslog* implement TLS encryption for log data transfer [144]; it has to be noted though that log encryption may not be implemented by all log source IoT devices, since it is not a part of neither RFC 3164 nor RFC 5424. It has to be noted that in some cases, transmitted log data may convey sensitive personal information, e.g. in the case that a smart watch monitoring the heart rate issues an alert; in these cases, the encryption during transport is indispensable.

The third issue is associated with user privacy: in the cloud storage scheme log lines may be (depending on the specification of the collector device) stored on remote servers, transcending network boundaries and/or falling under different access rights and jurisdictions. Therefore, the log maintenance configurations chosen should be tailored to preserve user privacy, while also enabling log collectors to perform causal analysis and event correlation. Selective transmission of events, execution of anonymization procedures before log data transmission or encryption techniques for sensitive log data are three indicative techniques that can be used to that effect. These procedures should be carried out while the log data is still in the data owner's control and jurisdiction, e.g. within a smart home, the relevant procedures could be performed in the Smart Gateway, which could be configured as a relay and/or a collector. Opting for private cloud services for storing log data alleviates privacy issues to a great extent.

Overall, the advantages of storing log files on a cloud server considerably outweigh the corresponding disadvantages. Hence this option is recommended by both industry/practitioners [129], [131], [40] and academia [182], [175] alike.

2.9 Network profiling

A network is simply a collection of computers or other hardware devices, i.e. IoT sensors and actuators that are connected, either physically or logically, using special hardware and software to allow them to exchange information and cooperate. Section 2.9.4 has already covered the communication protocols that dominate the IoT arena, and this section aims to elaborate on the networking element of computational activity profiling and its role in monitoring and protecting devices and the wider network. In the IoT every physical object becomes locatable, addressable and reachable in the virtual world. Privacy concerns for IoT devices are valid, especially as social media is increasingly incorporated into IoT services, i.e. Google Home and Facebook [90], and network profiling has a role to play in privacy preservation in IoT network profiling [17].

As more and more objects in the physical world are expected to connect to the Internet, the IoT is supposed to contain millions or billions of objects which will communicate with each other and with other entities (e.g., human beings). These objects not only include computers and laptops which already exist in traditional networks, but also physical devices such as home appliances, vehicles, etc. The heterogeneity of devices and technologies that are used for providing IoT services has had a great impact on the interoperability and management of IoT devices. Many devices have constrained resources and limited computational capabilities and are deployed in an open environment (e.g., street lights), which makes them prone to being controlled or destroyed by malicious attacks. With its inherent complexity and heterogeneous structure, the IoT is facing numerous threats and attacks which will negatively affect its normal functionality. Thus, protecting the security of the IoT is a difficult yet important task. Within the approach of Cyber-Trust, this protection has been divided into on-device protection via a device agent and network protection based on an enhanced gateway and associated network profiling. A high-level view of the complexity of the IoT network environment is shown below in Figure 2.5.

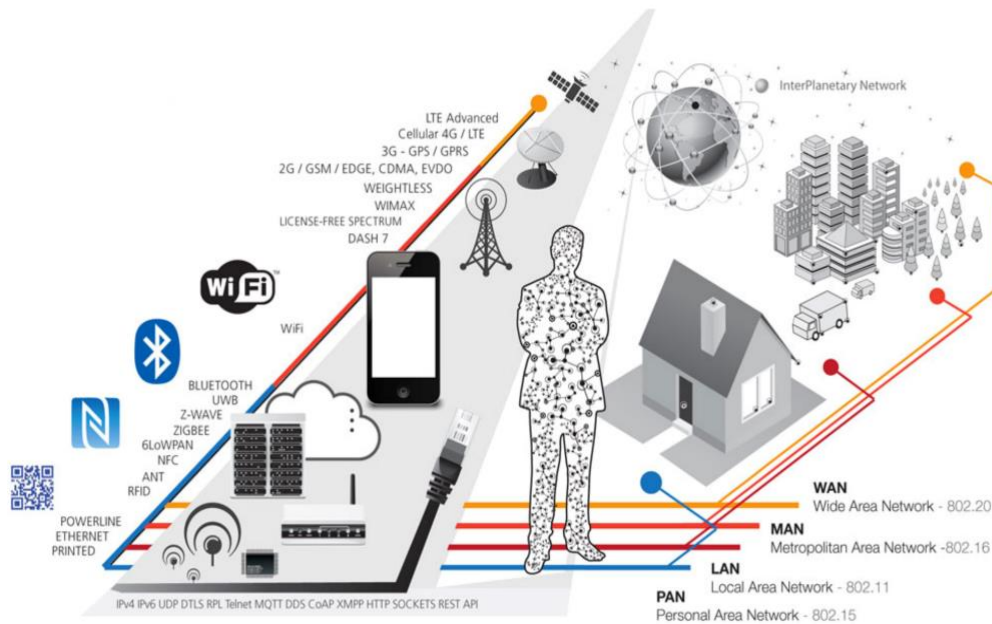


Figure 2.5: The IoT Network Environment [10]

Network systems need to be able to detect malicious activity and classify it to enable corrective actions to be taken. Accuracy and speed are critical factors in enabling effective threat classification and quarantining in order to prevent the propagation of threat vectors such as malware beyond an infected device into the local or wider network. For example, a botnet upon infecting a smart device, such as a home surveillance system, by compromising the software binary with an unknown (until now) malicious bot was installed [10] will listen for commands through HTTP and HTTPS and can execute three different types of attacks. These attacks are:

- DDoS;
- Eavesdropping;
- Spamming.

The bot is trying to replicate itself over the network using telnet/FTP/SSH default logins; however, it can also update itself from a C & C server with exploits that can attack more devices with firmware vulnerabilities. The details of such an attack are detailed in Cyber-Trust Document D2.3: Use Case Scenarios, however the relevant elements here are that the botnet's behaviour will create a unique network traffic and port usage signature that can be detected against the background of normal traffic and port usage provided the means by which detection occurs has the computational power and algorithms to conduct fast, accurate traffic flow analysis. Programming elements such as multi-threading and hardware elements such as multicore CPUs (or even specialist FPGAs) are all necessary devices by which to exploit existing capabilities to enhance the gateway and conduct localised traffic analysis, which can be further enhanced by GPUs [174], a concept which will be explored in the Cyber-Trust network profiling capability within the cloud.

2.9.1 Network Fundamentals

By way of introduction, it is necessary to highlight the details of the TCP/IP protocol stack, as we need to understand what it is we are profiling. The first sentence of this section provided a seemingly simplistic sentence to describe a network. Although simplistic it is accurate, and the enabler for this seeming simplicity is the TCP/IP protocol suite. Figure 2.7 below provides a simple overview of the protocol, and to enable less technical readers to understand its functionality a comparison with the introductory OSI Reference Model is provided.

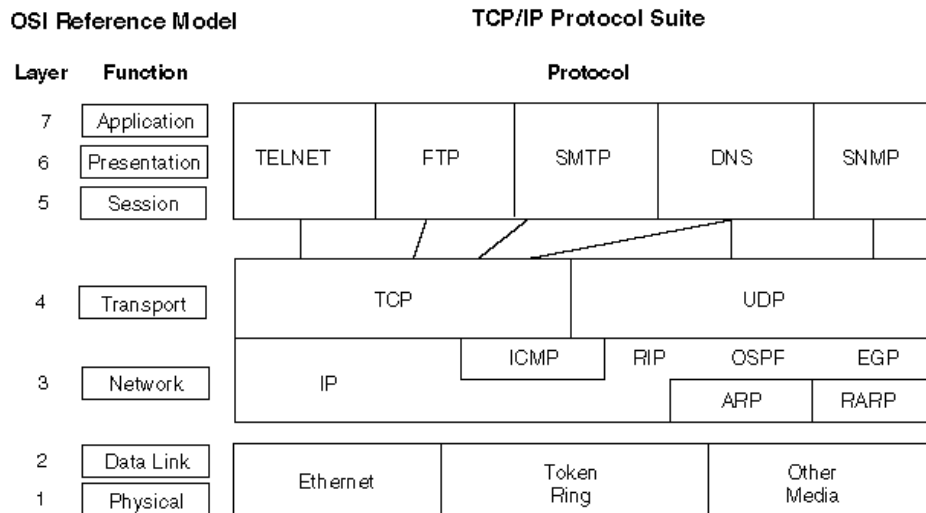


Figure 2.6: The TCP/IP Protocol Suite [174]

As is shown in Figure 2.6 above, network protocols are developed in layers, with each layer assigned “responsibility” for an aspect of the communication task. Ergo a protocol suite such as TCP/IP is simply a combination of layers. The TCP/IP layers are as follows, with informal TCP/IP IoT-capable elements in Blue, moving from the bottom up (with protocol details being discussed in more depth throughout the rest subsection of Section 4.9):

- **Link Layer (Network Driver and Interface (NIC) card):** This layer handles the hardware requirements and details for interfacing with the cable or wireless network.
- **Network (IP, ICMP, IGMP, 6LowWPAN):** This layer handles the movement of packets around the network. Routing packets take place at this layer, as an example. Internet Protocol (IP), Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP) are the network layer in the TCP/IP stack, but for an IoT network, it is necessary if not proper to place the 6LowWPAN (IPv6 over Low-Power Wireless Personal Area Networks) at this layer.
- **Transport (TCP, UDP):** The transport layer provides the data flow between hosts for consumption by the application layer. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are radically different protocols within the TCP/IP protocol suite. UDP is the primary IoT protocol at this layer. TCP is a reliable, addressable protocol providing bi-directional byte-stream communication, with reliability assured by the transport layer (ignored by the application layer) while UDP is an unreliable unaddressed broadcast providing multiplexing, with reliability assured by the application layer. UDP delivery is as that of IP, asynchronous and unreliable. The Real-Time Transport Protocol (RTP) sits within UDP for delivering VoIP, media streaming and video-teleconferencing. Both TCP and UDP produce a set of {1....65535} ports.
- **Application (Telnet, FTP, SMTP, SNMP, CoAP):** The application layer handles the details of a particular application. Some examples are File Transfer Protocol (FTP), Telnet, Simple Mail Transfer Protocol (SMTP) and more recently the Constrained Application Protocol (CoAP) designed to enable IoT and Machine-to-Machine devices within low power lossy networks to communicate via RESTful services.

2.9.2 Profiling the Network

Network Analysis and Profiling can be fixed on six key capability areas, within open source and commercial software operate to provide network operators with the tools necessary to understand, control and management the networks under their control, i.e. profile the network. The six principle areas are, including examples of applications are summarised in Table 2.2.

Table 2.2: The six principle areas are, including examples of applications.

Areas		Examples of applications
1	Network Spoofing and Redirection	A. DNSMasq; B. Ettercap.
2	Executable Reverse Engineering	A. Java Decompiler; B. .NET Reflector; C. IDA Pro; D. Hopper; E. ILSpy.
3	Web App Testing	A. Mitmproxy; B. Zed Attack Proxy; C. Burp Suite.
4	Active Network Capture and Analysis	A. Canape; B. Canape Core; C. Mallory.
5	Passive Network Protocol Capture and Analysis	A. Wireshark; B. SiLK C. LibPCAP; D. TCPDump; E. MS Message Analyser
6	Fuzzing, Packet Execution, Vulnerability Exploitation Frameworks	A. American Fuzzy Lop (AFL); B. Kali Linux; C. Metasploit; D. Scapy; E. Sully

For this document and this section in particular, the focus will be on the passive network protocol tools, as these align to the Cyber-Trust capabilities being discussed within this document and being implemented within the project. The tool used for the passive capture and analysis of network traffic flow will be the System-for-internet-Level-Knowledge (SiLK) toolset [67]. While proprietary toolsets such as NetFlow (Cisco) and ntopng (ntop) offer cut-down solutions to network monitoring, full functionality is retained for their respective commercial offerings, i.e. NetFlow via Cisco routers. SiLK offers full functionality as an open-source capability and delivers a powerful network analyst toolkit [85] centered on network flow. An example workflow is shown below in Figure 2.7: Example SiLK workflow.

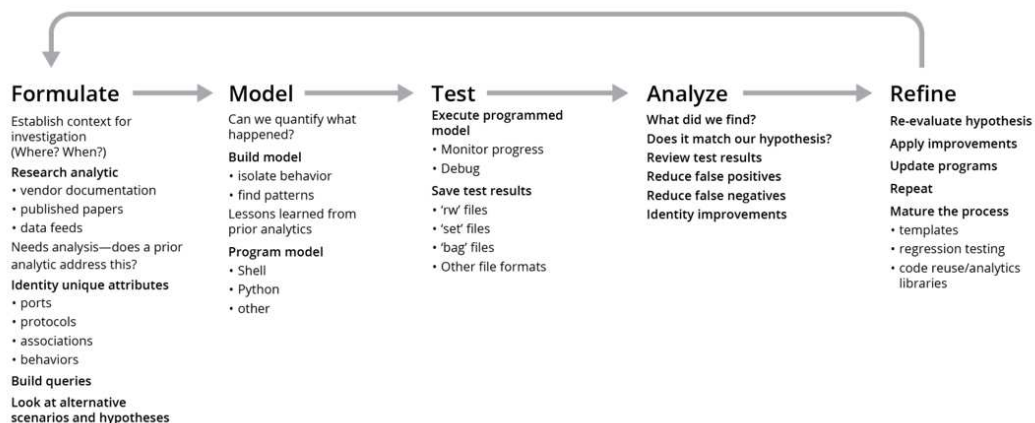


Figure 2.7: Example of SiLK Workflow [85]

By focusing on flow analysis as the primary method of detecting, profiling and protecting the Cyber-Trust network, we can maintain a computationally lightweight presence on the gateway while retaining separate packet capture and deep packet inspection (DPI) capabilities. By further extension, and linking to the SDN approach mentioned previously, network profiling can take a flow approach to trusted devices, and a deeper packet analytical approach to less trusted devices or devices demonstrating anomalous network behaviour via integration of trust metrics.

2.9.3 Passive Network Protocol Capture and Analysis

A network profile is an inventory of all the assets on a network and their associated purpose. It is a log of all network activity, rather than a recording of all packets, with packet metadata stored but not the contents, which allows flowing analysis to enhance privacy for network users. The metadata consists of the following types of information (not exhaustive):

- Source address, destination address
- Source port, destination port (Internet Control Message Protocol [ICMP] type/code)
- IP [transport] protocol
- bytes, packets in the flow
- Accumulated TCP flags (all packets, first packet)
- Start time, duration (milliseconds)
- End time (derived)
- Sensor identity
- Flow termination conditions
- Application-layer protocol

As the profile changes over time, especially within a dynamic environment such as IoT networks, network operators and defenders can monitor for emerging concerns, i.e. within the context of Cyber-Trust, this could be a new device with unpatched firmware. This, in turn, can lead to policy changes and reallocation of network resources. The network profiling process, as followed by the Cyber-Trust project, the general steps for network profiling are as follows:

- Gather available network information;
- Select an initial data set
- Identify the active address space,
- Catalogue common services;
- Catalogue other services;
- Catalogue leftover assets;
- Report on findings.

These steps are a cyclical process in order to ensure changing network activity is captured and catalogued. To enable this, Cyber-Trust will employ sensors to capture network services or packets for anomalous/untrusted devices as discussed previously) shown below in Figure 2.8: Sensor Functionality.

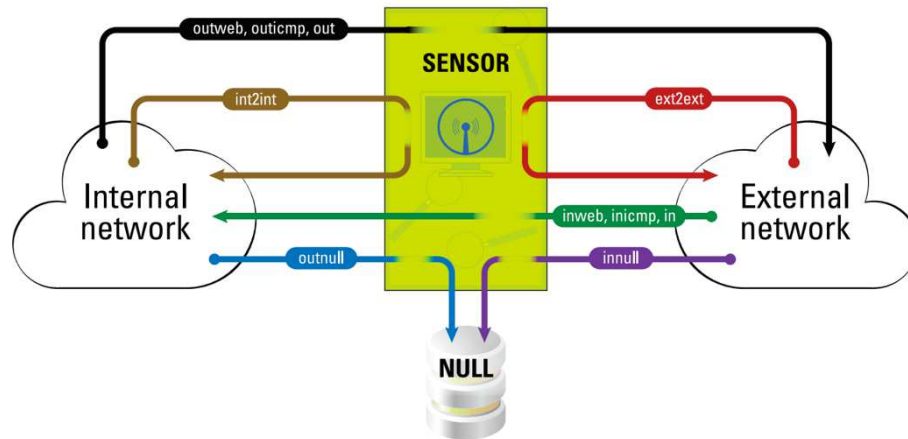


Figure 2.8. Sensor Functionality [85]

For the Cyber-Trust network monitoring capability, a two-step approach is required in order to encompass the limitations of local, consumer-grade networking items when conducting activities as described above, such as flow monitoring. As shown in Figure 2.9: High-Level Cyber-Trust Network Monitoring Approach, the residential gateway will be enhanced with an ability that will, amongst other activities such as Intrusion Prevention, provide a MUD-driven network monitoring capability.

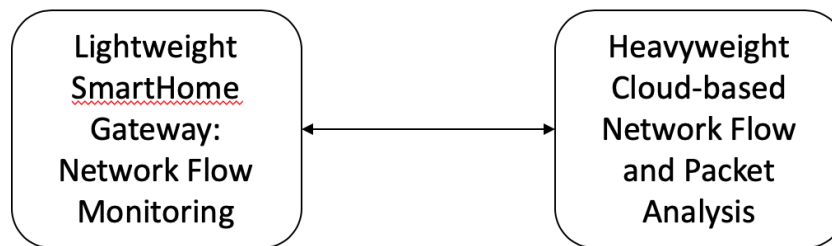


Figure 2.9: High-Level Cyber-Trust Network Monitoring Approach

As discussed, port, payload and behaviour-based classification of network traffic are all considered to be part of the Cyber-Trust solution. However, with a hard constraint of a residential gateway and its sparse resources, even when enhanced with a Raspberry-Pi-like device (1.4GHz 64-bit quad-core processor), it is necessary to implement a two-stage approach to network operations. At the edge, the residential gateway can be treated as an inference-layer, running flow-based behavioural algorithms, including a machine learning-based anomaly-driven IPS, to enable timely, accurate network profiling and anomaly detection with small resource availability. This is backed by an offsite, or cloud-based, analytical capability that has the computational resources to conduct DPI, packet inspection and malware analysis and to provide post-analysis rule-set, signature and behavioural updates to the residential gateway.

2.9.4 Protocols

The primary network communication protocols are ARP, IPv4, IPv6, ICMP and ICMPv6. In addition to these specific IoT protocols have been developed to enable low-power, low-resource devices to communicate effectively. These are 6LoWPAN (which is IPv6 over Low-Power Wireless Personal Area Networks specified in RFCs 4944, 6282, 6775 and can be seen as complementary to IPv6 over Bluetooth in RFC 7668 as described in Section 3.7) and CoAP (Constrained Application Protocol specified in RFCs 7252 and 7959). All these protocols will be discussed in this section.

2.9.4.1 Internet Protocol (IP)

The Internet Protocol has two versions in use, v4 and v6, both of which will be discussed here. IPv4 is the older and more common of the two versions and assigns unique 32-bit addresses to identify devices connected to a network. Any device that wishes to connect to the internet will require the capability to have an IP address, which is provided by the IP protocol (working together with the distributed Domain Name System, DNS, to assign textual names to these addresses). As the number of networked devices grew and the limit on the number of addressable IPv4 devices being ~4.3 billion devices, the IPv6 specification was developed. IoT devices use IPv6, and the specifications made several improvements on IPv4 such as a larger address space using 128-bit addressing (which creates an addressable space of a trillion trillion trillion, or undecillion) which supports the rapid growth in network devices as seen with the development of the IoT concept. Figure 2.10 below highlights the differences between IPv4 and IPv6. The use of IPv6 for the IoT is more than just about addressable space, one of the workarounds for the IPv4 address problem was to NAT networks (create local networks that all shared a single public IP but without a public IP themselves) which isn't ideal for IoT devices which often need to be accessed from the internet.

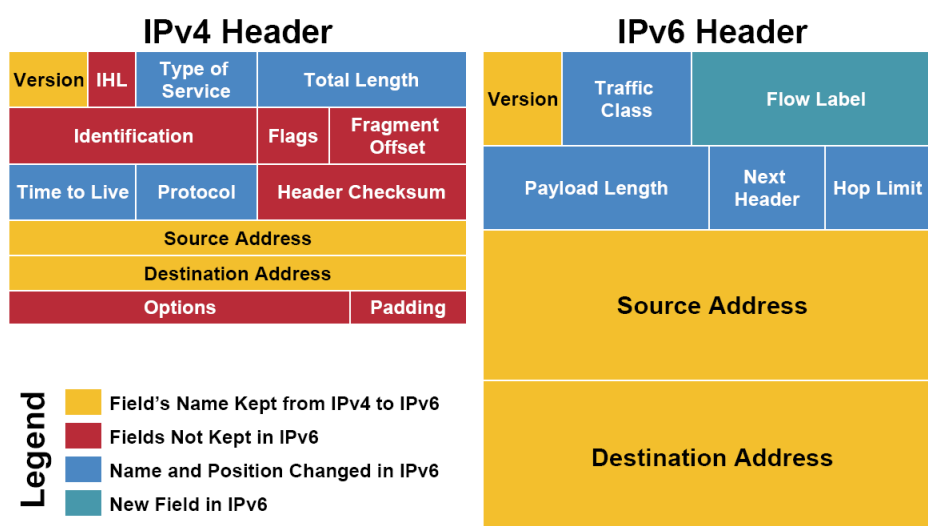


Figure 2.10: IPv4 and IPv6 Differences

As IPv6 is being discussed, it is appropriate to include 6LowWPAN, which is a compressed version of IPv6. Developers of IoT devices have focused on IPv6 and 6LowWPAN in order to support low-power devices, which when used together with the Constrained Application Protocol (CoAP) protocol enable devices such as embedded computers which have very low computational and memory resources, to be networked using the IPv6 protocol. Stateless Address Auto-Configuration (SLAAC) also provides an auto-configuration capability that means there is no need to configure IP addresses for end systems, even Dynamic Host Configuration Protocol (DHCP), ergo devices can communicate on detection allowing for a 'plug and play' approach that suits IoT devices. IPv6 also supports multicast more effectively and securely than IPv4, providing several multicast addresses for devices on the WAN, as well as any cast (unique to Ipv6) which provides for more robust allocation within IoT networks. The [iot6.eu](https://www.iot6.eu/) project is one initiative supporting the development of IoT6, the IPv6-aligned IoT.

2.9.4.2 Address Resolution Protocol (ARP)

Logical and physical addresses are used for communication on a network. Logical addresses allow for communication amongst multiple networks and indirectly connected devices, whereas physical addresses facilitate communication on a single network segment for devices directly connected via a switch. It would be too simplistic to assume IoT devices within a local network like a smart home fit within physical address constraints, as whilst they may have some perceived local network functionality, i.e. a thermostat

communicating with a phone app, the likelihood is that smart home and industrial IoT user software is more likely to consume data from a cloud-based service provider that is communicating with the device directly. Logical and physical addressing work together to enable data links to operate with the TCP/IP protocol suite using ARP to resolve an IP address to a MAC address (the address used in switch Content Addressable Memory (CAM) tables to map devices on its network). An ARP request broadcasts a requirement to connect to another device, about which only the IP address is known, and the device responds with an ARP response containing the required MAC address. In summary, ARP provides a mapping between IP addresses (which are only understood by the TCP/IP protocol suite) and a data link.

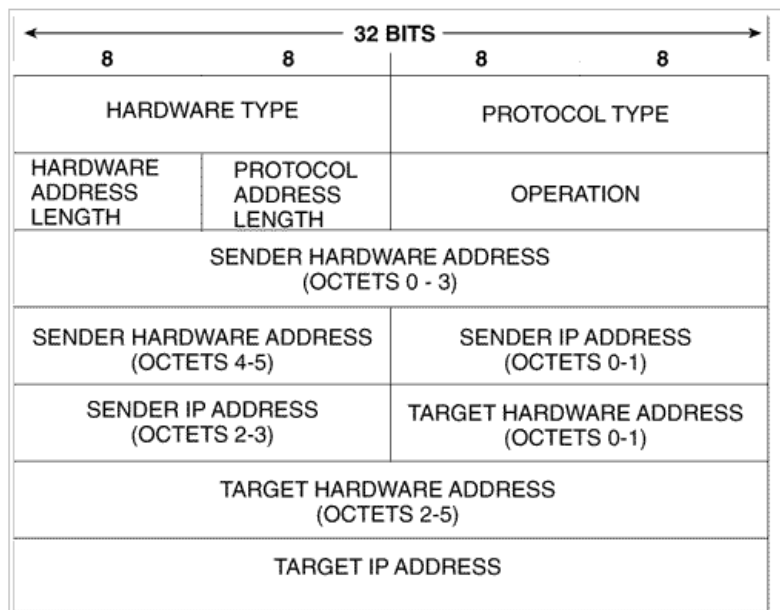


Figure 2.11: The ARP Packet Structure

2.9.4.3 Internet Control Message Protocol (ICMP)

ICMP is the Internet Control Message Protocol which is the utility protocol of TCP/IP responsible for providing information about device availability, TCP/IP network routes and forms a foundation protocol for enabling troubleshooting activities to occur. It relies on IP, and performs activities such as ping (which tests for connectivity to a device) and traceroute (which identifies the path from one device to another).

2.9.5 Ports

Ports serve as the interface between a computing device and other computing devices. Transport layer protocols, such as TCP and UDP, transfer data using protocol data units (PDUs). For TCP, the PDU is a segment, and for UDP a datagram. Both protocols use a header field for recording the source and destination port number. A port number is a 16-bit unsigned integer, ranging from 0 to 65535. For TCP, port number 0 is reserved and cannot be used, while for UDP, the source port is optional and a value of zero means no port. A process associates its input or output channels via an Internet socket, which is a type of file descriptor, with a transport protocol, an IP address, and a port number. This is known as binding and enables the process to send and receive data via the network.

The operating system's networking software has the task of transmitting outgoing data from all application ports onto the network, and forwarding arriving network packets to processes by matching the packet's IP address and port number.

Common software activities and hardware assign common ports examples of which can be seen below in Figure 2.13.

7 Echo	554 RTSP	2745 Bagle.H	6891-6901 Windows Live
19 Chargen	546-547 DHCPv6	2967 Symantec AV	6970 Quicktime
20-21 FTP	560 rmonitor	3050 Interbase DB	7212 GhostSurf
22 SSH/SCP	563 NNTP over SSL	3074 XBOX Live	7648-7649 CU-SeeMe
23 Telnet	587 SMTP	3124 HTTP Proxy	8000 Internet Radio
25 SMTP	591 FileMaker	3127 MyDoom	8080 HTTP Proxy
42 WINS Replication	593 Microsoft DCOM	3128 HTTP Proxy	8086-8087 Kaspersky AV
43 WHOIS	631 Internet Printing	3222 GLBP	8118 Privoxy
49 TACACS	636 LDAP over SSL	3260 iSCSI Target	8200 VMware Server
53 DNS	639 MSDP (PIM)	3306 MySQL	8500 Adobe ColdFusion
67-68 DHCP/BOOTP	646 LDP (MPLS)	3389 Terminal Server	8767 TeamSpeak
69 TFTP	691 MS Exchange	3689 iTunes	8866 Bagle.B
70 Gopher	860 iSCSI	3690 Subversion	9100 HP JetDirect
79 Finger	873 rsync	3724 World of Warcraft	9101-9103 Bacula
80 HTTP	902 VMware Server	3784-3785 Ventrilo	9119 Mxit
88 Kerberos	989-990 FTP over SSL	4333 mSQL	9800 WebDAV
102 MS Exchange	993 IMAP4 over SSL	4444 Blaster	9898 Dabber
110 POP3	995 POP3 over SSL	4664 Google Desktop	9988 Rbot/Spybot
113 Ident	1025 Microsoft RPC	4672 eMule	9999 Urchin
119 NNTP (Usenet)	1026-1029 Windows Messenger	4899 Radmin	10000 Webmin
123 NTP	1080 SOCKS Proxy	5000 UPnP	10000 BackupExec
135 Microsoft RPC	1080 MyDoom	5001 Slingbox	10113-10116 NetIQ
137-139 NetBIOS	1194 OpenVPN	5001 iperf	11371 OpenPGP
143 IMAP4	1214 Kazaa	5004-5005 RTP	12035-12036 Second Life
161-162 SNMP	1241 Nessus	5050 Yahoo! Messenger	12345 NetBus
177 XDMCP	1311 Dell OpenManage	5060 SIP	13720-13721 NetBackup
179 BGP	1337 WASTE	5190 AIM/ICQ	14567 Battlefield
201 AppleTalk	1433-1434 Microsoft SQL	5222-5223 XMPP/Jabber	15118 Dipnet/Oddbob
264 BGMP	1512 WINS	5432 PostgreSQL	19226 AdminSecure
318 TSP	1589 Cisco VQP	5500 VNC Server	19638 Ensim
381-383 HP Openview	1701 L2TP	5554 Sasser	20000 Usermin
389 LDAP	1723 MS PPTP	5631-5632 pcAnywhere	24800 Synergy
411-412 Direct Connect	1725 Steam	5800 VNC over HTTP	25999 Xfire
443 HTTP over SSL	1741 CiscoWorks 2000	5900+ VNC Server	27015 Half-Life
445 Microsoft DS	1755 MS Media Server	6000-6001 X11	27374 Sub7
464 Kerberos	1812-1813 RADIUS	6112 Battle.net	28960 Call of Duty
465 SMTP over SSL	1863 MSN	6129 DameWare	31337 Back Orifice
497 Retrospect	1985 Cisco HSRP	6257 WinMX	33434+ traceroute
500 ISAKMP	2000 Cisco SCCP	6346-6347 Gnutella	
512 rexec	2002 Cisco ACS	6500 GameSpy Arcade	
513 rlogin	2049 NFS	6566 SANE	
514 syslog	2082-2083 cPanel	6588 AnalogX	
515 LPD/LPR	2100 Oracle XDB	6665-6669 IRC	
520 RIP	2222 DirectAdmin	6679/6697 IRC over SSL	
521 RIPng (IPv6)	2302 Halo	6699 Napster	
540 UUCP	2483-2484 Oracle DB	6881-6999 BitTorrent	

Figure 2.12: Common TCP/UDP Ports [63]

As the above Figure 2.12: Common TCP/UDP Ports shows, within any profile analysis of network traffic there is a multitude of ports in use at any one time for everyday tasks and device-orientated tasks. Unfortunately, these do not mean that these standard ports can be ignored, as malware can utilise a device's assigned port for malicious activity, i.e. Mirai which is why the ability to detect anomalous network behaviour is critical.

2.9.6 Volume Traffic & Muddy Filtering

As the preceding sections have shown, networks are areas of intensive traffic and analysis of them is no easy task, indeed volume traffic could also be labelled big data in terms of its volume, complexity and variety, especially within the IoT environment (such as a smart home) where a plethora of devices are networked and communicating, often without any visibility to the user, i.e. smart fridges, lights and thermostats given the common lack of a UI within the IoT device taxonomy. This complexity has the potential to increase exponentially as more devices come online, and an example profile of a Smart Home (which is the focus of the Cyber-Trust project at this stage) is shown in Figure 2.13: An example Smart Home and its network edge below, where the network edge (the smart devices) are detailed and which can be interpreted at the network level as a closed group of NICs with software and hardware-driven port assignments.

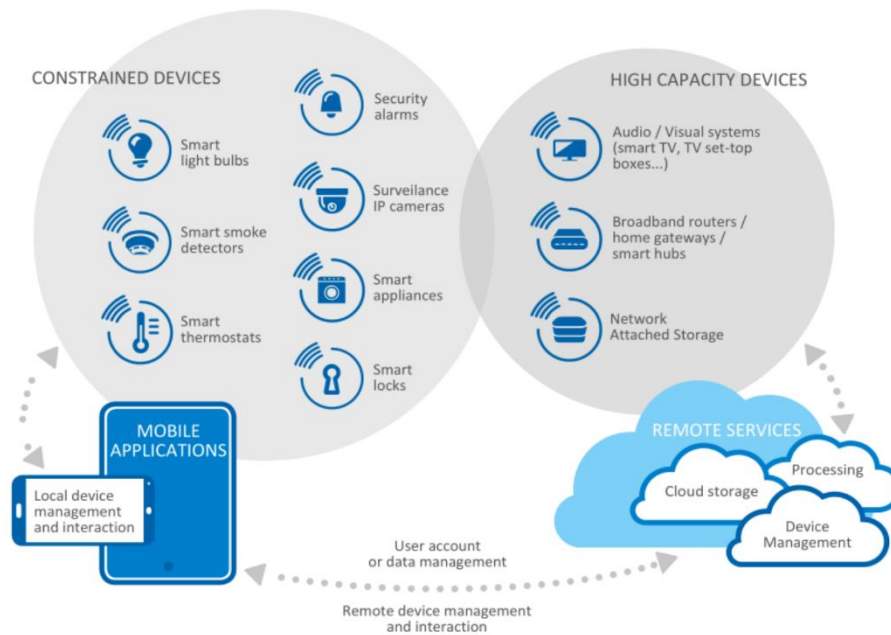


Figure 2.13: An example Smart Home and its network edge [18]

Filtering is the key to be able to tackle this volume and complexity, where traffic can be analysed based on elements such as destination IP, or port numbers or sliced according to device type. Software-Defined Networking, or SDN, will be part of the network solution to managing this volume and to support the network profiling though targeted packet management [158], [63][18], falls outside the scope of this report. However as discussed previously, computational overhead and accurate analytics are also key to effective profiling and anomaly detection. Here, the Cyber-Trust Gateway Service will use the Manufacturer Usage Description, or MUD, to deliver device-focused network profiling to support accurate feature-set development for use in statistical-learning based anomaly detection. To place MUD in the correct context within the network, Figure 2.14 below details its use within the Cyber-Trust project.

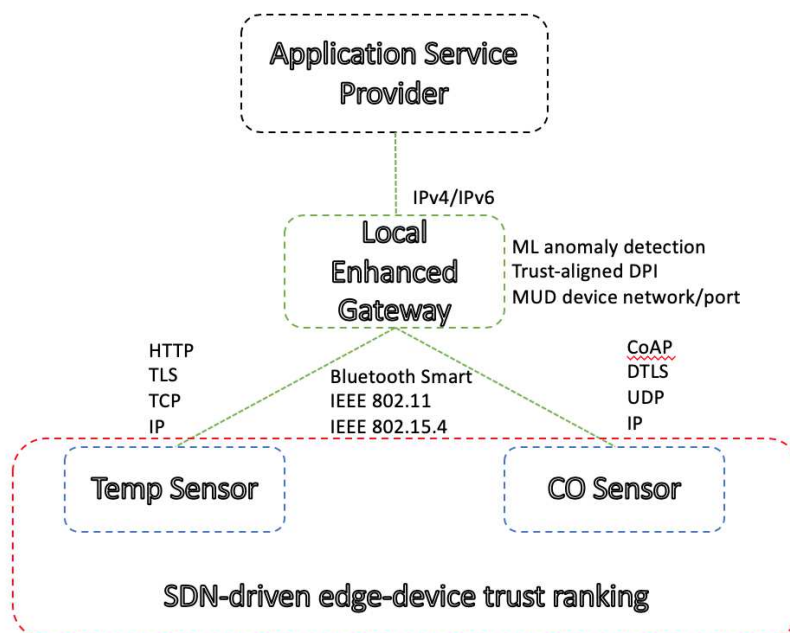


Figure 2.14: The Cyber-Trust MUD environment

As discussed previously, IoT devices are increasingly leveraged by cyber-attacks, either in actuality or by implication, as the number of devices proliferates and the security mechanisms for these devices remains ad-hoc or non-existent. The challenge with IoT devices is that security protocols for internet-connected devices were designs for computationally-capable edge devices with non-heterogonous architectures, i.e. a desktop computer or mainframe. The IoT introduces a more complex security paradigm insofar as the edge becomes a collection of heterogonous low-power low-resource devices utilising local and cloud-based resources in order to deliver a specific function (or a very limited set of functions). As such, profiling has both local network and gateway-to-internet behaviour to consider.

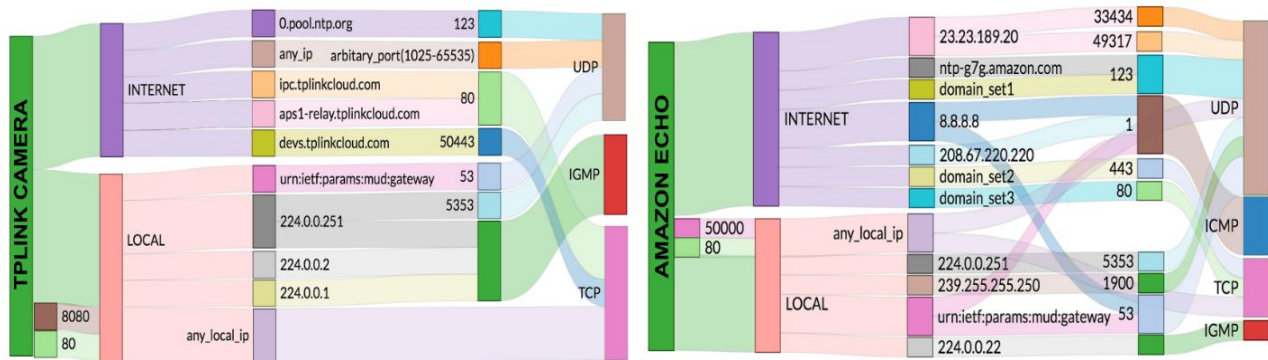


Figure 2.15: Results of MUD analysis on IoT devices [65]

The Internet Engineering Task Force (IETF) MUD specification is still under development, and so not implemented by manufacturers, a shortfall that has been overcome by researchers and whose open-source software will be used as part of the network profiling for Cyber-Trust. MUD enables devices to signal to the network what access and functionality they require in order to function, and provides an enforceable policy mechanism for enhancing the network security of IoT devices. At its core, MUD requires manufacturers to provide a behavioural profile of their device(s), i.e. an IP camera may need to use DNS and DHCP on the local network and communicate with a cloud-based controller and NTP servers. This, in turn, enables network controllers to auto-develop a device-specific access control list (ACL), placing restrictions on IoT devices and reducing the potential attack surface on the network while providing a baseline security model for the attached IoT devices themselves [65].

Consumer IoT devices expose services to local hosts and use services provided by remote cloud servers. As shown above in the Sankey diagram in Figure 2.16: Results of MUD analysis on IoT devices, [65] the TPLink Camera and Amazon Echo have network profiles that consist of local and internet connections. Traffic profiling using MUD via the MUDgee application[65] (which utilises IoT traffic flow according to rules set by [88]) shows that IoT devices, especially IP cameras, use the Session Traversal Utilities NAT (STUN) protocol to check a user can stream video from the camera over the internet, which means that profiling and network management has to allow all UDP traffic to and from internet servers as the STUN protocol often requires the client device to connect to different IP addresses or port numbers. This general rule means that UDP is a risk-vector for malicious infection or activities that profiling and anomaly detection need to be attuned to, given a general 'block UDP' rule would render common user IoT devices which stream video useless, which would make Cyber-Trust unworkable.

IP cameras communicate with many remote servers operating on the same port, which means that remote traffic to and from any IP address on that specific port number must be allowed [65]. This creates a challenging security situation in that an IP camera could communicate with a botnet C&C server using the same port as a benign service server. Without a rule to block common IP addresses, i.e. a blacklist, for IoT ports it means that it is unlikely that IoT devices such as IP cameras can have the network trust level that means the flow is the only means by which traffic to and from such devices is monitored. At this stage, as shown in Figure 2.14: High-Level Cyber-Trust Network Monitoring Approach, it is likely that such devices will be managed at a lower trust level and subjected to packet analysis and DPI.

MUD analysis provides how IoT devices and their network behaviour can be understood and thus managed from a security perspective. A device can exchange DNS queries/responses with the local gateway, communicate with a single domain name utilising a single port. Such behaviour can be classified as static and allows the device to be locked to a specific set of flow rules [65]. However, other devices can have a transparent local profile, such as the TP-Link IP camera shown in Figure 2.16: Results of MUD analysis on IoT devices [65], allowing for a clear set of access controls, but over the internet and utilising its STUN server it accesses an arbitrary range of IP addresses whose provenance is unknown, however, behavioural profiling can still be applied even if a lighter set of access controls is required [116].

Another step up in complexity is represented by the Amazon Echo, again shown in Figure 2.15: Results of MUD analysis on IoT devices [65], are devices with complex and dynamic functionality using custom recipes, which thwarts behavioural profiling and so arguably creates a case for continual packet analysis and further deep packet inspection based on thrust profiles for the device itself, i.e. firmware patching etc, as shown at Figure 2.14: High-Level Cyber-Trust Network Monitoring Approach. What Cyber-Trust cannot do is drive security from the provenance of the manufacturer, as Amazon is as much at risk of malicious activities as is Huawei, Belkin or TP-Link [116].

3. State of the art in malware detection and mitigation

3.1 Introduction

There are numerous recent examples of cyber-attacks exploiting lightweight IoT endpoint devices to perform distributed denial-of-service (DDoS) attacks of unprecedented scales, spy on people in their office/homes, or hijack communication links to deliver full control of remotely controlled objects to cyber-criminals. The availability of massive IoT botnets-for-hire is expected to lead to a significant increase of cyber-security incidents targeting at critical information infrastructures (CIIs), which provide vital functions that our societies depend upon, and should be considered as high risks. Their formation is facilitated by the security problems arising from embedded devices and legacy hardware, whose flawed design or poor configuration allows cyber-criminals to compromise them. This is easily achieved using known (even years old) vulnerabilities, since there are often no (efficient) means to patch those devices for preventing any further exploitation. However, the cases where previously unknown (called zero-day) vulnerabilities are employed to compromise connected systems have grown noticeably, as they are found on black markets that have evolved in the darknet. Cyber-Trust ambition is to address the emerging cyber-security challenges and prevent vulnerable IoT devices from being used as a vehicle to attack CIIs as well as other infrastructures. In order the above to achieved we need to see the landscape of attacks and review the state-of-the-art in order to come into conclusions that will be implemented in the WP6. Thus, the rest of this chapter provides a thorough analysis and reviews of the areas malware detection and mitigation.

3.2 Malware

There are numerous recent examples of cyber-attacks exploiting lightweight IoT endpoint devices to perform distributed denial-of-service (DDoS) attacks of unprecedented scales, spy on people in their office/homes, or hijack communication links to deliver full control of remotely controlled objects to cyber-criminals. The availability of massive IoT botnets-for-hire is expected to lead to a significant increase of cyber-security incidents targeting at critical information infrastructures (CIIs), which provide vital functions that our societies depend upon, and should be considered as global risks. Their formation is facilitated by the security problems arising from embedded devices and legacy hardware, whose flawed design or poor configuration allows cyber-criminals to compromise them. This is easily achieved using known (even years old) vulnerabilities, since there are often no (efficient) means to patch those devices for preventing any further exploitation. However, the cases where previously unknown (called zero-day) vulnerabilities are employed to compromise connected systems have grown noticeably, as they are found on black markets that have

Term malware is created by combining two words 'malicious' follows the Standard English definition which implies the actions characterized by malice, and the word 'software', which is a combination of programs to execute a specific task. Intrinsically malware is a malicious piece of software design by malware authors with the intention to destroy, destruct and damage the normal functionality of Information systems. It has become the most disastrous and pernicious cyberweapon like never before and with the passage of the time, there is a significant increase in the growth of malware and its complexity. Nowadays there is a drastic increase in the production of malware variants, according to McAfee 2018 reported that they had seen an exponential increase in the power shell malware which grows by 267% in 2017 [112], Panda labs quoted that 200,000 new malware samples were spotted per day in 2014 [127] and Internet security report of Symantec reports that 350 million malware variants were developed in 2016 [72].

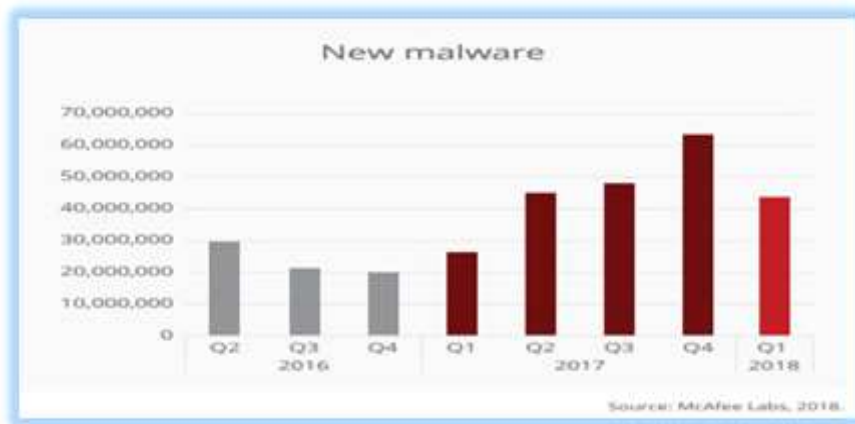


Figure 3.1: Malware trend (Source: [112])

The mounting malware proliferation is not just damaging the information system but also challenging the critical networked infrastructure of the states all over the world. Currently, it has become one of the most heinous threats posed to the state's security structure and if a threat enters in the system, the security of hundreds of computers is compromised [170]. According to an INFOSEC report, these attacks can bring catastrophic destruction to the whole population of a state. Previously, malware attacks such as Wannacry ransomware and PetiYa have disrupted the critical infrastructures of numerous states in 2017, so the current wave of malware necessitates rapid, creative and inclusive solutions. With the increasing popularity of the internet and rapid development of the information system, malware has become more complex and complicated and has the very first virus sample to complicated worms, and pernicious Trojans to notorious rootkits and nowadays to ransomware like CryptoLocker, WannaCry, NotPetya, etc.

Malware can be broadly classified into the following categories depending on their working and the mechanism of propagation.

3.3 Classification of Malware

3.3.1 Viruses

The best known and one of the oldest types of malware is a virus and word 'virus' itself is Latin which is used for "poison", and a term computer virus is derived from and analogous to a biological virus. As most of the viral infection is spread by a small shell containing genetic material when injects its nefarious contents into larger body cell of any human being it infects that specific area with it poison, in similar way computer Virus is a small program designed with harmful intent and possess the ability to poison/damage the computer, moreover possess the ability to replicate itself. In contrast to worms, viruses do not use network resources for their propagation and usually, it works by appending virus code to an executable file. Another difference between virus and worm is that a virus always requires some user interaction or intervention in order to spread itself, whereas in worm no user intervention is required, and it can spread automatically. A virus copies can penetrate into other machines through the network or by using different types of compromised media e.g. USB stick, floppy disks etc. one possible scenario to infect other users on the network is by infecting files hosted in a shared location on the network in order to compromise other systems. The virus usually comes with very specific and targeted binary executables files for e.g. portal executable which is the most common format for Windows, COM, exe files in MS-DOS, documents with compromised macros, malicious script files etc.

In some occupational scenario, the virus is carried by worms as additional payloads or in another case; it has been observed that virus itself can include the functionality of backdoor or Trojan which compromise the valuable information and data on the target machine. The virus can be of different types e.g. an original copy of the virus is being modified to make new variants to produce metamorphic virus. The typical virus works in

a way that it adds itself to executable code of some other software so that malicious or viral code can be executed before the code of its infected host.

There are three different ways that a virus can add itself to the host code.

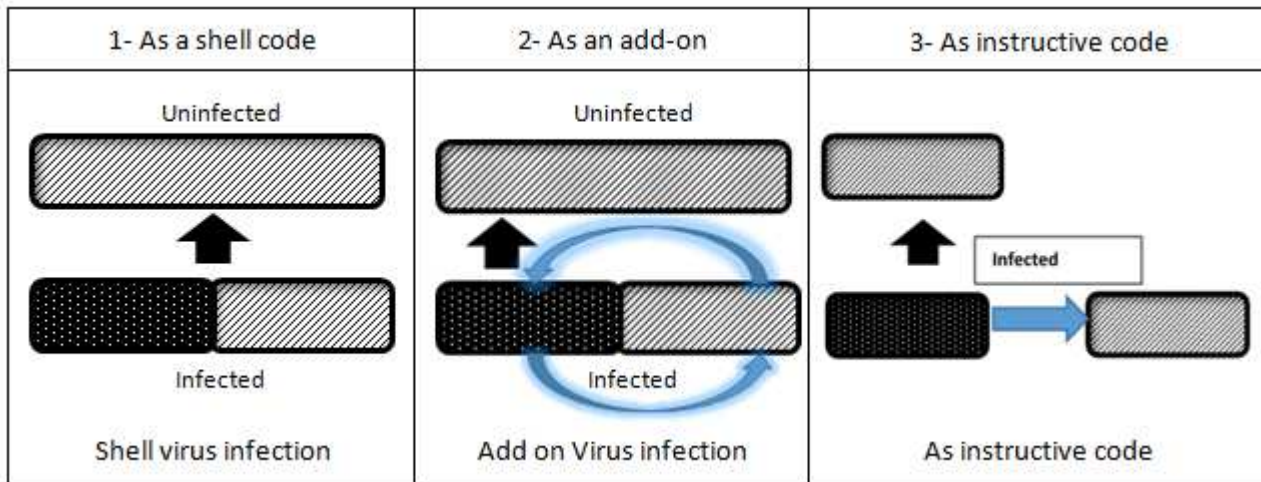


Figure 3.2: Ways that a virus can add itself to the host code

3.3.2 Worms

A worm is a type of malware that propagates independently via local area network or public internet by exploiting known and unknown vulnerabilities in software to damage a system [100]. It uses different networking system to replicate or propagate themselves for instance; some time by penetrating a remote system, sometime launch copies on target systems, in some cases through file-sharing(P2P), most commonly through networks by using email to compromise other system and may copy themselves by using file transfer protocols, or by IRC channels and WANs [100].

It is interesting to understand how worms penetrate the system. The most common path is the file form whereby worms are induced in file attachments and ICQ messages through P2P Networks. Besides these files attached worms, there are some file fewer worms as well. These file fewer worms penetrate in the system as network packets and immediately spread in the, RAM where the code execution takes place. There is a wide variety of method by which worms penetrate in the victim's machine and execute the code. Most of the times it is the email which seeks the attention of victim to open that specific malicious email and eventually leaves the system vulnerable or even, sometimes, compromises its security. It is significant to note that unlike viruses, worms spread automatically. Viruses do not spread automatically rather an intervention by the user is ultimately required for the virus to accomplish its task whereas worms do not require such action by the user. Because of this fundamental difference between these two, the system infection caused by the email attachment or MS word document would fall in the category of a virus instead of a worm classification.

In 2006, a graphic icon was used by Leap A. in the form of a JPG image. This was penetrated by using the iChat messenger client and latestpics.tgz was the name of the file used in this worm attack.

3.3.3 Trojans

The Trojan is the short name for the 'Trojan Horse', which has its links with the Greek mythology of Trojan War, in terms of its meaning. The story is based on a wooden horse, used for stealth, which Greek troops utilised for the city of Troy invasion. Relying on the same principle, in computing, a Trojan horse is a kind of malware which appears as something very useful and attractive to misrepresent the user for the sake of convincing the user to click on it and install it. Trojan seems to perform a completely different action which it actually performs i.e. malicious action which is not known to the victim. It aims to complete its task by

running the program secretly in a way that the user or administrator of that system is unable to shut it down or delete it. Most of the times, the Trojan horse's payload is a backdoor which is used to infect the system. Sometimes, a Trojan which is called as the dropper is used to inject the worm in the local networks of the system. The Trojans perform numerous functions which are not known to the user of the system and are performed without the consent of administrator. These tasks include but are not limited to collecting specific data and then transferring it to the cybercriminals, sending spam emails, destroying the system's data, using the system for criminal purposes and many other malicious functions [100].

In general, Trojan is a type of malware that misleads the user and invites him to run a specific malicious program. This payload may disrupt the system immediately and delete important data or may cause any other disastrous consequences. A clear example of Trojan is presented here. According to ESET, in 2004, the first most Mac Trojan was Amphimix that was perceived as an MP3 file with an mp3 icon, whereas, in actual, as a result of clicking on it, a message was displayed, to play an audio file 'wild laughter'.

3.3.4 Bots

The bot is another type of malware and the word used is coming from the short of Robot. Bot works on the automated principle; therefore, it usually has an active interaction with other networks. The collection of useful information is the fundamental task of bots while it also automatically communicates with other applications such as with any web interface or Instant Messaging. Bot software is useful for the operator in terms of providing access to the operator to control each system from far away or anywhere remotely. It also allows him to make a zombie army or botnet by controlling and gathering each system collectively in the form of a group. The benefit of forming such kind of zombies for attackers is that while launching attacks, they are able to conceal their original identities by using anonymous proxies with the help of utilising these Botnets or zombies [103].

It is significant to note that a Botnet is a group of a large number of compromised systems all over the internet service. There are several ways which the attackers can use a botnet. Some of these include flood type attacks, attacks from far away or remote-control attacks and broad-based attacks to affect their particular targets. Contemporarily, the bots used are, usually, the combination of the other threats. In general terms, they have the qualities of other malware types. For instance; they have the ability to conceal and are unable to detect like viruses, they are able to penetrate like worms, they may solely launch an attack just like other tools and in addition to this, and they also have a unified command and control system. Plus, they have also been using backdoors to provide them access to other networks. It should also be noted that bots penetrate in other networks while remaining concealed and unnoticed.

Attackers show an increasing interest on controlling our Smart Home devices, and recent examples like the Mirai botnet highlight this ambition. Mirai infects poorly protected internet devices by using telnet to find those that are still using their factory default username and password. The effectiveness of Mirai is due to its ability to infect tens of thousands of these insecure devices and co-ordinate them to mount a DDOS attack against a chosen victim. It has been reported that Mirai attacks exceeded 1 Tbps—the largest on public record.

3.3.5 Ransomware

Ransomware is another type of malware in which victim data is encrypted by a hacker with his own key and remains in the same state until ransomware is being paid by the victim, usually hackers to pay money using cryptocurrency to keep them anonymous [99]. An example is quoted here from the year 2016. BitTorrent client Transmission's official servers offered infected files. This kind of malware was noticed for the first time and was considered as 'the first functional ransomware for Mac'. Its process does not appear in the front rather it communicates with the KeRanger C&C servers remains in the background and seeks required data and information to properly start the encryption phase. Then, three days later, the documents' and files' encryption take place by using KeRanger process. Finally, KeRanger displays the demand to pay the ransom for decryption of their files and also stores necessary information and instructions for the system's administrator to pay the ransom.

3.3.6 Backdoors

Backdoors is also a type of malware which easily goes through the usual authentication processes. When any of the types mentioned above disrupt the system, then it becomes extremely convenient to install one or more backdoors to seek access to the system in the future as well. Another important point is that these backdoors are able to be installed before any other malicious software is installed in the system. It is usually assumed that companies that manufacture computers already install backdoors to assist their customers when an issue arises in the system, but this has not been verified yet [100]. A recent detected was found in a vulnerable firmware of almost all dbltek GSM-to-VoIP devices, a range of equipment mostly used by small to medium size businesses, it claims. Trustwave researchers claimed they had found hundreds of at-risk devices on the internet vulnerable to be infect with this backdoor [183].

3.3.7 Spyware and Adware

Any kind of software which is installed in the system when the system's administrator is unaware of the installation, it is known as spyware. This software gathers all the required data and information which is, afterwards, transferred to the attacker and, as a result, attacker utilises the collected data to seek credit card numbers, passwords and even to change or covert the settings of the system. Spyware is usually distributed using a Trojan horse which spreads with the file that system user unknowingly downloads, considering it another file. Therefore, spyware is installed in the system when the user clicks on the software with the intention to instal all it. Sometimes, spyware administrator uses legal way to interact with the victim by displaying a licence agreement which, most of the times, is not read or understood by the system user. An instance of spyware is LogKext. An open source LogKext comprises of user space, kernel extension and client. The purpose of kernel extension is to drive the keyboard hardware by using the call back functionality [99].

3.3.8 Rootkits

The program which is fundamentally designed for seeking the control of a computer system but without being noticed or allowed by the system's administrator or any other legal user. The major task of the rootkit is to take control of the operating system, therefore, seeking access to the hardware is not obligatory or a required objective. Usually, rootkits do not show their presence in the system and manage to conceal themselves. Most of the times, they are Trojans to seek control of the other systems. The most common method which is used for this objective is hiding the running processes to supervise any of the programs [100]. Security researchers from ESET came across a Unified Extensible Firmware Interface (UEFI) rootkit in the wild being used for cyberespionage. Named LoJax (detected by Trend Micro as BKDR_FALLOJAK.USOMON and Backdoor.Win32.FALLOJAK.AA) after the legitimate anti-theft software LoJack, the rootkit is reportedly packaged with other tools that modify the system's firmware to infect it with malware.

3.4 Malware Analysis Techniques

Malware analysis is a study to dissect malware in order to understand its behaviour; moreover, it also articulates how to study the different components and artefacts of malicious software. Malware is usually analyzed either through static or by dynamic analysis. Each of this technique has its own advantages and disadvantages which are being highlighted in the next section. Furthermore, static analysis can be divided into two stages, namely basic and advanced static analysis. Malware analysis based on dynamic analysis can also be further subdivided into two categories, named as basic dynamic analysis and advance dynamic analysis as depicted in Figure 3.3.

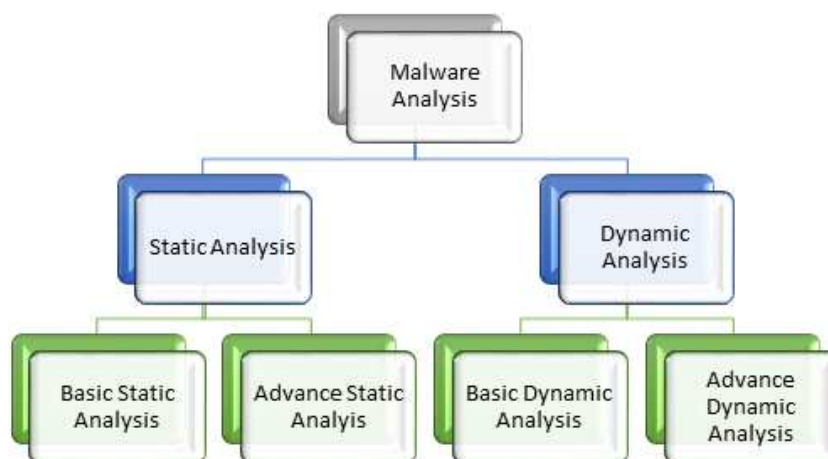


Figure 3.3: Malware Analysis Method

3.4.1 Basic Static Analysis

This method is carried out against the programs which are alleged as malware by passing it through different AV solutions, furthermore applying hashing and performing structural analysis of portal executables are also part of this technique. Some of the commonly used tools for performing basic static analysis are virustotal.com, MD5deep, PEID, PEview, D4dot, RDG Packer, Exinfo PE etc.

3.4.2 Advance Static Analysis.

In this analysis technique, malware functionality is explored by examining its static properties which is the process in which malware is reverse engineered in order to analyse the code rather than executing it. Numerous approaches have been proposed by the researchers to perform static analysis for e.g. few of them are performed by extracting byte code sequence from the binary, by disassembling the binary file in order to the extract the opcode sequences, to mine control flow graph from assembly file, sometimes by mining API calls from binary file, all these extraction methods are based on the characteristics of binary file. Each of above-mentioned techniques constitutes features set which later on are used for detecting malware. The most common tool for performing the advanced static analysis is IDA disassembles, which is used to convert machine language code into human understandable language. By using this tool, a significant amount of information is collected for the malicious program which later on can be used to identify the characteristics [48] of malware. Some of the commonly used tools for performing basic static analysis are virustotal.com, BinText, Dependency Walker, IDA etc.

Numerous approaches have been proposed by the researchers in the past based on static analysis. The author in [9] proposed a technique based on the static analysis to detect malicious and benign samples. In [35] 4gram features were extracted from the portal executable. Later on, extracted features were used to differentiate between malware and benign samples. In latest study Opcode was used as a technique to detect malicious files. In this research Opcode of malware samples were obtained by reverse engineering technique, furthermore in author view from Opcode one can understand the sequence of operation performed by malware and these Opcode instructions can play an important role in distinguishing legitimate software from malicious software.

A similar study was proposed in [103], where opcode-based similarity measure was developed. The proposed technique is similar to the simple substitution of traditional cryptanalysis method. This research depicts good results for detecting metamorphic malware. Researchers in [61], propose a method using static analysis to detect malware, in this method API call sequences and assembly code were combined and later on the matrix was created on the basis of similarity which was used to find whether the specific portion of code contains malicious traces or not. This paper proposes two different detection methods named as SAVE (Static Analyser for Vicious Executables) for assembly call and MEDic (Malware Examiner using Disassembled Code) utilises API call for analysis.

Although there are lots of advantages of static analysis e.g. it is quite fast and does not require any control environment to execute malicious software but in, there are some disadvantages of this technique which makes it ineffective e.g. this technique can be easily thwarted by packing, metamorphic, polymorphic Techniques, etc.

3.4.3 Basic dynamic analysis

In order to overcome the deficiencies and discrepancies in static analysis focus of security analysts have shifted to a dynamic approach. In this method, malicious samples are executed in a controlled, confined and simulated environment to model the behaviour of malware. The main advantage of this approach is that it is not affected by evasion technique like obfuscation and it can capture the polymorphic or metamorphic strains of malware. Moreover this technique is completely independent of the source code of the malware. Moreover, with the help of different monitoring tools and analysis software the behaviour of malware is captured. Below mentioned are some of the tools which are plays vital role for performing dynamic behaviour analysis (Table 3.1).

Table 3.1: Tools that plays a vital role for performing dynamic behaviour analysis

Tools		Description
1	VirtualBox	VirtualBox provides a controlled virtual environment to execute malicious software in order to analyze malware.
2	Process Monitor	This program is designed to capture, monitors and displays all activities taking place in a running system.
3	Process Explorer	This tool is designed to monitor the resources for the Windows operating systems.
4	ApateDNS	It is a tool which is used to find the desired IP address as requested by malware, regardless of which hostname is being resolved, moreover the tool possesses the capability to logs all DNS queries it processes.
5	Wireshark	It is a freeware and open-source software used for network packet analysis, network protocols analysis, for troubleshooting etc. it allows the users to visualize the network traffic at the run time.
6	Sandboxes	It is a security mechanism for running malicious software or malware in a safe and control environment without damaging the actual system. Cuckoo, SNDBOX, Norman Sandbox, GFI Sandbox, Anubis, Joe Sandbox, VMRayalyzer are the few examples of famous sandboxes.



Figure 3.4: Example of sandbox architecture

3.4.4 Advanced Dynamic Analysis

In the advanced method of dynamic analysis tools like debugger etc. are used to analysis. Some of the disadvantages of these methods are as follows

- Time and resource intensive analysis
- In some cases, this technique is prone to analysis evasion as nowadays advance malware observer the environment before their executions and in case of finding the virtual environment they halt the launching of their payload.

3.5 Signature-Based Techniques

It is considered as a simplest and most efficient and effective way to detect the variants of malware. The technique refers to static analysis, which is based on examining malicious samples in order to collect the information to characterise it either it benign or malicious. This technique is done with intend to extract the sequence of bytes and later on use them as a signature. In this technique, security analyst create handcraft

signature for malicious files which later on are maintained in a database which needs to be updated continuously to detect emerging threats. So, whenever any file is passed through signature-based system first of all the code of that file is compared with existing database repository which contains collection of signatures constructed on the basis of sequence of program instruction or bytes, if signature of file matches with any one of the existing signatures it is considered as malicious else benign.

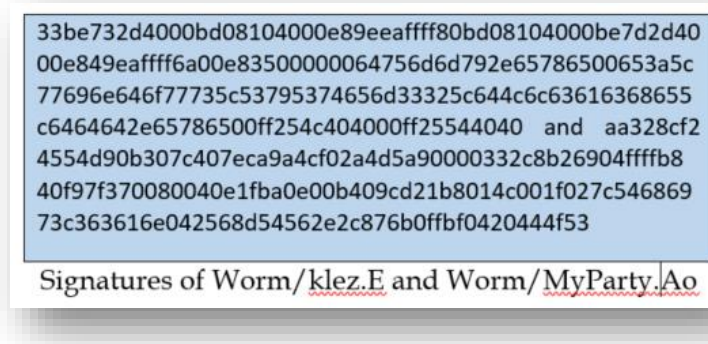


Figure 3.5: Image containing Signature of Worm

Most of the commercial antivirus companies use this technique to differentiate malicious files from a benign file. In [50] researchers discuss the excellent overview of this technique. Most antivirus companies use this technique by collecting numerous signatures to categorize file as clean or malicious. One of the major drawbacks of this technique is that it requires human intervention in order to keep the database of signatures up to date and researchers in [99] showed that metamorphic strain of malware could easily thwart this mechanism which leads to false negative alerts. Another interesting study was done in which authors have proposed obfuscation signature engine. This engine works by scanning the opcode instruction pattern as generated by a known engine to detect malicious software.

This technique proves to be more efficient as it works by identifying the set of instruction opcode sequence and matching this against the signature of the engine rather than scanning the byte stream to match with engine signature.

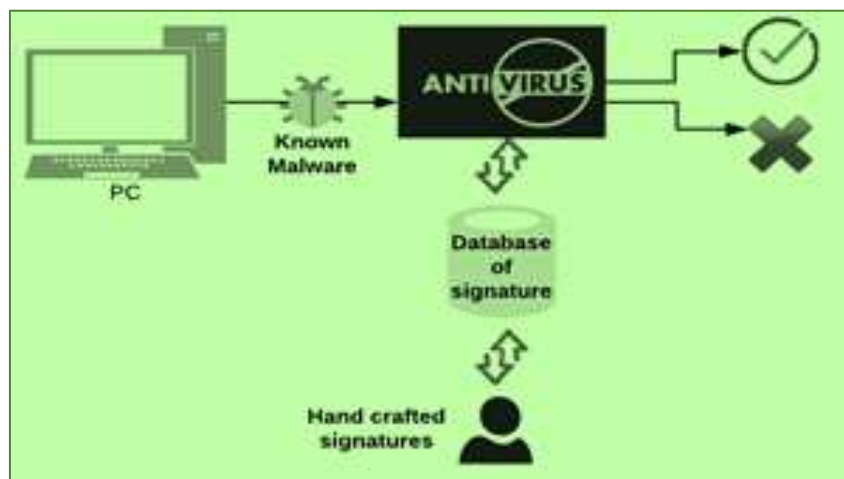


Figure 3.6: signature-based system

3.5.1 Snort

Snort is a lightweight network instruction detection system that was developed in 1998. It is one of the most extensive and widely used network instruction detection and prevention software deployed both in network and research environment. Snort possesses the capability to analyze the protocol and data flow in real time [82]. Snort is a single-threaded application and consists of six parts which are as follows

- Catching data package
- Analyzing the code of data
- Pre-processing the package
- Parsing the rule
- Detecting the engine
- Logging

Moreover, snort possesses the ability to be configured and operation in four different modes, it can work as a sniffer, as NIDS, as packer logger and instruction prevention system. Some features are elementary features of snort, e.g. packet sniffing and logging functionality; however, its fame is because of its capability to use as NIDS and instruction detection system. With time, the new features have been added to snort to increase the strength and functionality of it, e.g. IPS is one of the new features which has been added to cope up with the malicious traffic, e.g. to take preventive measures again dropping and re-direction packets to another destination.

3.5.1.1 Pcap

In order to capture the raw packets snort uses libpcap and before forwarding to detection engine snort firstly decode and latterly on pre-process it as shown in Figure 3.7.

3.5.1.2 Decode and Pre-processing

The pre-processing step consists of the following functionality

- Early packet droppings
- Classification,
- Layer three IP fragment reassembly,
- Layer four TCP session reconstruction,

3.5.1.3 Detection engine

This engine is responsible for inspecting several packet headers as well as payloads against several thousands of rules which are stored in a database of pre-defined attack signatures, as shown in Figure 3.7. If any of the rules match with a database of pre-defined attack signature a prompt action is taken depending upon the configuration of that rule. Generally, 'alert' and 'log' are the mostly used, alerting facility is indication any suspected packet, moreover, with the help of logging facility, all information related to the packet is saved. Snort also possess the feature of displaying the 'alert, and 'log' in a number of different formats along with a variety of methods. E.g. binary, ASCII, libpcap etc. The binary format is considered as fast and flexible, whereas with the help of libpcap user can inspect by different tools, and lastly ASCII format is considered to easier and fast.

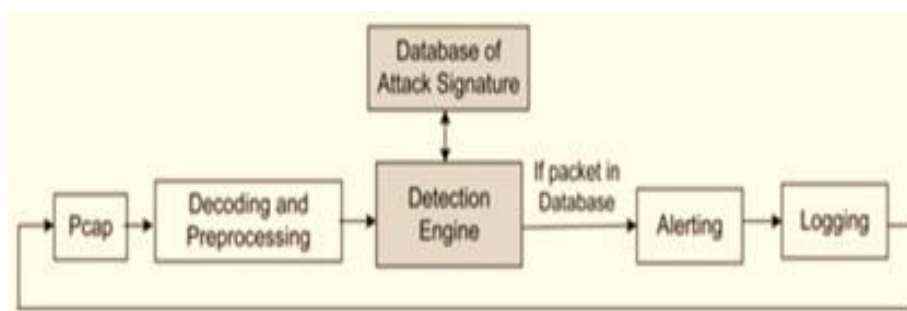


Figure 3.7: Snort Engine [82]

Snort executes its components and task in a sequential manner as depicted in Figure 3.7, therefore kernel or libpcap buffer all the packets and which later on fed to snort sequentially for processing in case of detecting malicious packets Snort block the traffic and generate alert and log this activity. In contrast, if incoming traffic is non-malicious and contains normal packets, Snort will allow this traffic to enter into network without any delay. One of the most integral components computationally intensive part of snort is a detection engine which is on one side is very effective and on the other end, it is very complex. The detection engine works by analyzing every packet against and matches packet payload strings against thousands of Snort rules which are also called pre-signatures are populated at runtime payload. Till date, there are approximately 8000 rules in snort for detecting malicious packets.

3.5.2 Suricata

In 2009 next generation IDS was introduced by the Open Information Security Foundation (OISF) known as 'Suricata'. It is signature-based network intrusion-detection which was initially funded by Department of Homeland Security's Directorate for Science and Technology. It utilises externally developed rule sets to monitor the ongoing activities and generate alerts to the security analyst on detecting malicious activity in the network [83].

3.5.2.1 Development and features

It is designed as a multi-threaded system in a way to take maximum benefit of multiple cores; moreover, it is designed in a way that it can work with traditional and existing network security components. One of the distinct features of this device is that it provides security analyst with the visibility into the Application layer. Furthermore, its effective HTTP streams parsing makes it one of the fastest IDS till now. Another powerful feature of this device is that it possesses the ability to inspect the HTTP traffic without relying on the port number to distinguish between types of network traffic; moreover, it also allows the security analyst to extract files from HTTP session for analysis by scrutinising inside protocol streams.

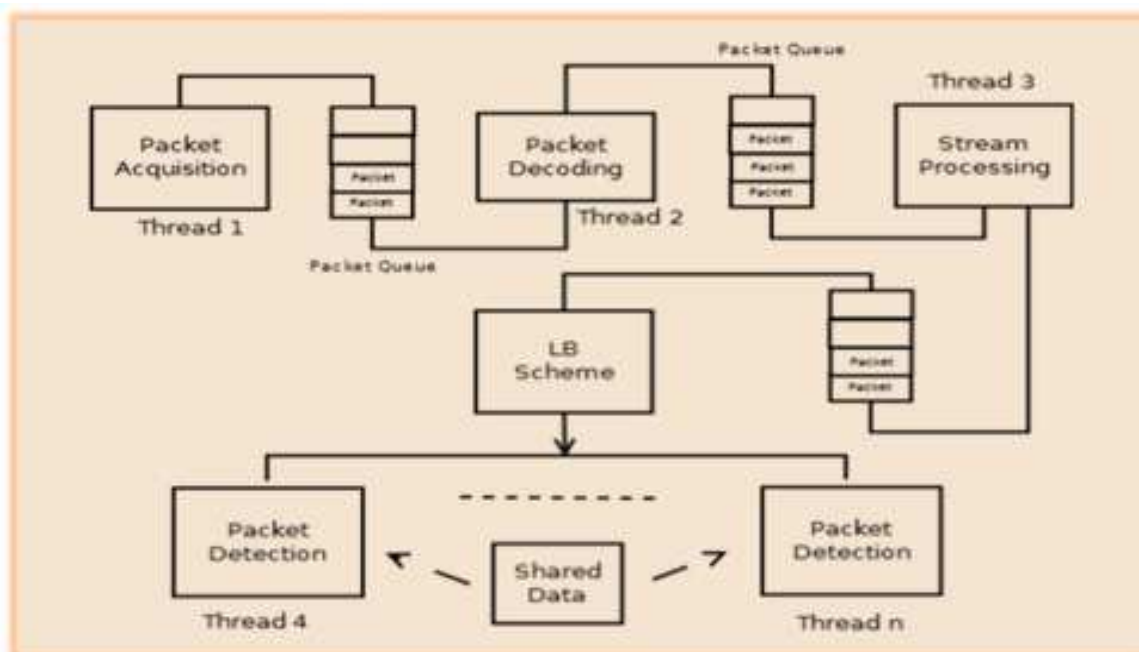


Figure 3.8: Suricata architecture [83]

3.5.2.2 IDS/IPS

It is designed in a way that it can use and work with other IDS/IPS ruleset e.g. snort ruleset can be integrated with Suricata to monitor network traffic and generate alerts on detecting suspicious activity. Suricata runs on two different versions of Linux (2.4 and 2.6) operating system and it possesses the functionality to monitor passive traffic as well as inline traffic, moreover it can also handle high-speed traffic levels e.g. multiple fast gigabit traffic, although version 2.4 also possess the same capability as in version 2.6 but some advanced features are not available in this version e.g. no inline option is available in this version.

3.6 Behaviour Based Techniques

In this technique, the behaviour of the system is monitored against the defined set of requirements and against security policy which is baseline model for normal behaviour of the system. Furthermore if any activity deviates from that normal profile, it will be considered an anomaly. The process of designing, generally start by collecting information on what makes normal behaviour for the network and what constitutes abnormal behaviour for the network, moreover in the learning stage, it is trained with a set of rules that contain normal behaviour of any normal application, and deviation from this behaviour will be considered as an attack, misconfiguration or anomaly [46]. One of the biggest benefits of this technique is that it can potentially recognize unforeseen attacks. However, in contrast one of its drawbacks is that it contains a high false alarm rate. In this technique, the major task is to understand and make an analysis of the behavioural aspects of either known or unknown malware.

The basic behavioural aspects comprise of several factors, for instance; source of malware, types of attachments and some statistical aspects. Furthermore, these two techniques can also be applied by utilising different kinds of analysis, like; static, hybrid or dynamic analysis.

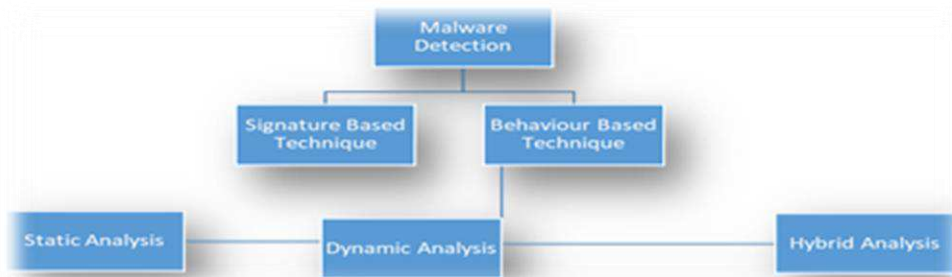


Figure 3.9: Malware Detection Techniques

3.6.1 Bro IDS

Bro a Unix-based and open source Network Intrusion Detection System was developed in 1998 by Vern Paxson in Network Research Group at Lawrence Berkley National Lab, and by the International Computer Science Institute. It possesses the capability to intercept malicious activity by performing passive monitoring. Bro works as network traffic analyzer and classification engine which provides a number of features ranging from file extraction, hashing to forensic which can be used to protect the organization from advanced threats by performing following tasks which are as follows

- It filters the network traffic and removes all the irrelevant elements which are of less important for analysis.
- Depicting network traffic data into high-level coherent events.
- Translate the event into tangible and actionable information which can be used to secure the network.
- It extracts the network related information and activities from metadata and uses a programming language to provide an indication when that activity will be malicious.

BRO can be used for multiple functionalities e.g. it can be used for

- To perform behavioural monitoring,
- It can be used as Policy enforcement,
- In most of the network, it is providing Policy-based intrusion detection
- To perform a multi-layer analysis which includes the finding of specific attacks as defined by signature and malicious activities e.g. connection details related to the host and also tell how the hosts are connecting to different services.
- For logging network-related activities.

3.6.1.1 Components of Bro IDS

Bro IDS consists of following basic components (Table 3.2)

Table 3.2: Bro IDS

Component		Description
1	Libpcap	The purpose of this component is to remove all irrelevant elements from network traffic and to forward the packet to the event engine
2	Event Engine	It combines the packets received from the libpcap to make an event which describes the executed actions.
3	Policy Script Interpreter	Comparison of high-level events with the policy scripts. If it detects any anomaly it will take action according to policy else, it discards the event.

Table 3.2: Bro IDS components



Figure 3.10: Bro IDS working

3.6.1.2 Machine learning concepts and definitions

Machine learning is an approach in which computer learn from examples and experience rather than explicitly programmed. So, in other words machine learning is able to extract intricate patterns hidden in data to make formal model, a model is a mathematical representation of underlying data properties. In the case of anomaly or malware detection, hidden properties of samples or dataset is used to train the model

for classifying it as malware or benign. Machine learning based anomaly detection can be categorized into two subclasses. One of them is based on supervised machine learning and other is based on unsupervised machine learning. In supervised machine learning, the labelled training set is required. This consists of samples of normal and anomalous traffic [167]. It is used to construct the predictive model. Algorithms used in supervised machine learning are decision trees, supervised neural networks, support vector machine learning, Parameterization of training model, Bayesian networks and k-nearest neighbours. Among these algorithms, Bayesian networks and k-nearest neighbours are popular. The rate of detection of supervised methods is better than the unsupervised methods. In unsupervised machine learning, the training data is not required [179].

Numerous researchers have proposed different frameworks, models and mechanisms based on supervising and unsupervised machine learning to cope up with this problem. One such framework is proposed by researchers in [109] used flow-based traffic analysis and supervise machine learning to detect botnets attacks by capturing traffic patterns of malicious botnets. In this, study eight different machine-learning algorithms e.g. NB, SVM, C4.5, random tree, random forest etc. Furthermore, it was found from the experiments that Random tree classifier is best performing machine-learning algorithm, moreover the authors have also evaluated the size of traffic needed per flow required to capture the patterns of malign traffic. Series of experiments were performed to check the efficacy of the system by capturing the traffic traces from P2P botnets and malign applications. Authors have shown that their model can detect the accurate and timely botnet traffic if monitored only for certain period of time along with a number of packets per flow by just using purely flow-based traffic analysis and supervised machine learning.

The use of deep learning for detection of network attack in cyberspace is increasing day by day because of its capability of learning pattern, a resilient mechanism to novel attacks because of its capability to extract a high level of features. In [1] proposed a distributed deep learning framework to detect an attack in range of networks e.g. IOT/Fog network etc. In authors view deep learning can discover a hidden pattern in traffic dues to its self-taught and compression capability which helps discriminate malicious traffic from benign traffic. The performance of the proposed system is measure using accuracy, detection rate, false alarm rate etc. which show that performance of proposed architecture base on deep learning to detect intrusion in a system is much than the shallow model.

Unsupervised machine learning in the context of anomaly detection is based on the following two assumptions. The first assumption is that most of the network traffic is normal whereas some are abnormal. Then the second assumption is the malicious traffic is statistically different from the normal traffic. Using these two assumptions, the data groups which are frequently appeared considered as normal and the data groups which are infrequently appeared considered as malicious. The algorithms used in unsupervised machine learning are SOM– self-organizing maps, C-means, K-means, expectation-maximization meta-algorithm (EM), one-class support vector machine and ART-adaptive resonance theory. Among them, SOM is one of the popular techniques.

This technique needs a branded training usual that covers both usual and irregular examples for building the predictive model. Hypothetically, supervised approaches are supposed to deliver healthier discovery rate than unverified methods. The greatest shared oversight procedures are overseen neural nets, parameterization of exercise perfect, provision course mechanism knowledge, k-nearest neighbours, Bayesian networks and decision trees. K-nearest neighbour (KNN) is one of the most conservative nonparametric methods that are used in oversight knowledge aimed at anomaly detection. It computes the estimated spaces between dissimilar points on the input vectors and then assigns the unlabelled point to the class of its K-nearest neighbours. The Bayesian network is an additional general model that can encode probabilistic relations between variables interest. This technique is generally used for anomaly detection in combination with arithmetical schemes. These oversight techniques have numerous compensations, counting the competence of indoctrination interdependencies between variables and of forecasting events, lengthways with the aptitude for incorporating both previous knowledge and statistics.

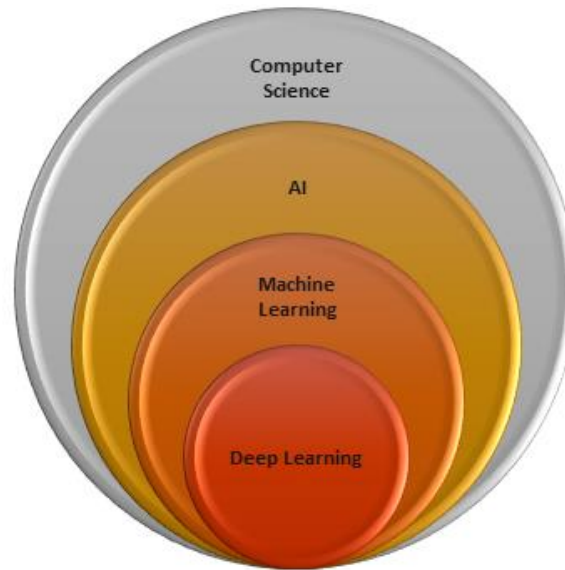


Figure 3.11: Visual representation of the relationship between data-related fields

Network intrusion detection systems always play vital roles in detecting advances attacks e.g. Denial of Service (DoS), DDoS etc, most of NIDS in the past were developed based on shallow architecture, but with the advent of deep learning the mechanism for developing efficient and flexible NIDS is totally change and now days advance NIDS are designed on approaches based on deep learning. In [134] researchers have proposed one such solution for the development of an efficient NIDS, which is based on a sparse autoencoder and soft-max regression. In this study, writers use self-taught learning one of the deep learning technique on an NSL-KDD dataset for network intrusion.

In a recent study proposed in [61] introduces an intrusion detection based on a deep belief network (unsupervised learning) to detect the attacks and classify attacks. In this paper, authors have performed a number of experiments using NSL-KDD dataset. The proposed system did not only detect attacks but also can classify into five groups. Furthermore, it was able to accurately identify and classify network based on limited, incomplete and nonlinear data sources. The authors showed that their system is able to achieve 95% accuracy for only fifty iterations which shows the efficacy and efficiency of their system is much higher than any other system proposed until now.

3.7 Evasion of Malware and Anti-Evasion Approaches

3.7.1 An overview of Evasion Approaches and Malware Camouflage Evolution

Since the emergence of malware, the challenge is to enhance and prolong the lifespan of the malware. This task is conveniently achievable when the working of antivirus meets the required standard. In order to make the malware successful, the malware code's camouflage plays a significant role. Generally, the four generations of malware provide a foundation in the development of stealth methodologies. These generations are; Encryption, Oligo morphism, Metamorphism, and Polymorphism which are displayed in Figure 3.12. The explanation of each of the generation is described below [100].

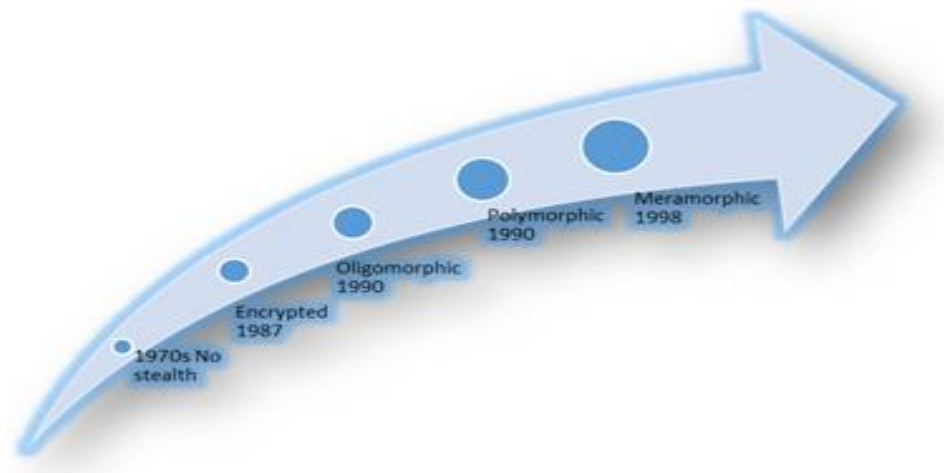


Figure 3.12: Phases of Malware for development of stealth methodologies [99]

3.7.1.1 Encryption

The most important issue for the malware authors is to escape and keep themselves safe from all code analyzer technicians. This helps them to have more time for the lifetime of their produced malware. The first technique used by the malware programmers for this purpose was encryption and in 1987, the first-ever encrypted virus was produced. [13] It is notable that the two fundamental parts of the encrypted virus are: main body and a decryption loop. Decryptor has the function to decrypt or encrypt the main body's code. Whenever the virus initiates to run on the victim's system, the first step is the decoding of the main body into a machine executable code by the decryptor loop and also converts it into meaningful data. The overall working structure of an encrypted virus is shown in (Table 3.3).

Table 3.3: Structure of an encrypted virus [13]

	Before decryption	After decryption
Decryptor	For i=1 to size of (body) Decrypt byte(1); Jump to body	For i=1 to size of (body) Decrypt byte (1); Jump to Body
Virus body	Encrypted Bytes (Not Visible Before Decryption)	Infector() ; ... Payload() ;

3.7.1.2 Oligomorphism

The next development in hiding the malware produces the oligomorphic viruses. The other name to oligomorphic viruses is given as semi-polymorphic. It is the advanced and next phase after encryption. The effort was to give a unique appearance to the decryptor loop of encrypted virus in every attack and infecting a system. [73] While producing this virus, different types of decryptors are selected randomly to attack a new system. This is the technique to avoid using identical codes for various victims. In 1990, Whale was the first virus and it was a DoS virus. [128] Oligomorphism is not considered as the main issue for the antivirus software. The reason is that it becomes slightly more difficult for antivirus to observe this kind of malware. Oligomorphism is not like an encrypted virus and antivirus needs to check all decryptors rather than merely checking one decryptor which takes longer time.

3.7.1.3 Polymorphism

Polymorphism is the most advanced and complicated form of viruses among previously described terminologies, i.e. oligomorphism and encryption. [120] The term ‘Polymorphism’ is generally used for decryption engine and in polymorphic malware thousands and thousands of decryptors can be produced simply by manipulating the instructions in the next variant. Although Polymorphic viruses have a resemblance to the encrypted and oligomorphic viruses in terms of utilising code encryption, on the other hand, they have the capability to create infinite new decryptors. Usually, decryptors consist of variable elements which are fixed size instructions and are designed with the intention to change the size or shape of the code. In general obfuscation techniques are used to produce Polymorphic malware. It contains potentially highly variables elements which can allow changes like subroutine creation, algorithmic register initialization etc., and making it difficult to capture the signature of unpacking stub.

In 1990, Mark Washburn created, 1260, which was the very first Polymorphic virus that was of the chameleon family. The purpose of this virus is to make it more difficult to analyze and this is done by changing the actual appearance of the virus. The fundamental principle is to bring a modification in the code’s appearance constantly for making it unobservable in each different copy. This is the obligatory phase before it is launched on a victim. It is done to escape the detection and to avoid any exploitation by the antivirus scanner engine. These techniques are quite difficult in terms of practical implementation and management.

3.7.1.4 Metamorphism

Igor Muttik has defined Metamorphosis in these most precise and concise words; “Metamorphics are body-polymorphic”. [128] This technique is different from the previously explained other three types because it does not have any encrypted phase or part. This is the reason it does not require any decryptor. However, on the other hand, it has some similarity with the polymorphic virus in using a mutation engine. In addition to this, the whole body is mutated rather than changing merely the decryptor loop. Therefore, metamorphic malware contains a mutation engine whose job is to generate and change the code every time it is being executed while keeping the algorithm same. This mechanism is achieved by changing the registers, substituting the instructions with equivalent ones with diverse operands, inserting garbage code, scrambling the sub-routines within the program etc., so as a result of every time a new signature will be produced which will easily thwart the signature-based detection system.

In 1998, the first metamorphic virus, AGG, was produced for the sake DoS. Besides that, W32 was the first metamorphic whereby an attempt was done by 32 bits metamorphic virus, which launched an attack to target Portable Executable files.

Two of the major evasion techniques are listed below (Table 3.4).

Table 3.4: Major evasion techniques

Evasion techniques		Description
1	Packing	In this technique packed executable is created by applying compression or some time encryption algorithms; the resultant packed executable contains an unpacking stub along with the original packed code. The unpacking is quite complicated as can easily thwart static analysis because when this packed binary is executed, the operating system will first load unpacker stub in memory which will, later on, unpack the application bring all the necessary exports, imports and later on transfer control to the original entry point of the executable.
2	Obfuscation	As time is changing, vendors are finding ways and developing advanced techniques to outsmart them. Currently, the same classic way of malware penetration is employed but now, advanced techniques and variants are used which are quite different from those of early days of first boot sector viruses. In these methods, the fundamental purpose is to escape the Intrusion Detection System (IDS). Here, in this section, some modern techniques of concealing the

		malware detection will be discussed. Most of these are utilised in metamorphic and polymorphic malware. [70]
3	Dead code insertion	This is one of the simplest techniques which gives instructions for changing the appearance of a program and not its behaviour. [70] Nop is an instance of this kind of instructions. In this instruction, the original code is obfuscated. This technique can be undermined and defeated by the antivirus scanners in such a way that ineffective instructions are deleted. However, to avoid this detection, authors use more complex code sequences.
4	Register Reassignment	This is also a very simple technique in which the code and behaviour remain the same while registers are modified and switched from generation to generation. [47] The only challenge to this technique is from wildcard searching that has the ability to make this method useless.
5	Subroutine Reordering	In this technique, the already set order of the original code's subroutines is obfuscated randomly. A wide variety of $n!$ variants can be produced in subroutine ordering. [70] Here, the number of subroutines is denoted by n . for instance; 10 subroutines generated by Win32/Ghost, it becomes $10! = 3628800$ different generations.
6	Instruction Substitution	As its name depicts, instruction substitution replaces some instructions with some other which are similar to the earlier ones. In this way, it evolves an original code. For instance; mov can easily be replaced with the other word like push/pop. This technique has a great capability of code modification using a collection of words with equivalent instructions [47].
7	Code Transposition	In this technique, the order of the original code's instructions takes place while it keeps the behaviour the same.
8	Code Integration	This technique was put forward by the Win95/Zmist malware. While practically applying this technique, in the first step decompilation of the program takes place by Zmist in order to turn them into manageable objects and then in an unobservable way, it becomes a part of them by adding itself between those objects and finally, the integrated code is reassembled which ultimately becomes the part of a new generation.

3.8 Anti-Evasion Approaches

Generally, two basic techniques, which are opposite to the evasion approaches, are used for anti-evasion of malware [159].

3.8.1 Malware Deobfuscation

Deobfuscation is a technique which is used for reverse engineering obfuscated code. It is opposite to the obfuscation technique created for malware penetration. It is undoing the obfuscation phases. There are several methods for deobfuscation. One common method is explained below in three deobfuscation steps. [159]

3.8.2 Unpacking

This is the first step in deobfuscation which involves starting the malware, taking its image and afterwards, copying it to write in an executable manner which can be used for further analysis, like; analyzing the code or in disassembling. It is noteworthy that there are, generally, two types of unpackers: generic unpackers and particular unpackers. Both of these have different tasks to perform.

3.8.3 Binary rewriting and editing

In this phase, all kind of rewriting is eliminated which has been created on the original binary code. After that, the results are authenticated and validated.

3.8.4 Malware binary reconstruction

In this phase, the actual malware executable is reconstructed. This is done by making the import table again. Here, the original entry point of malware is identified to eliminate the obfuscation.

3.8.4.1 Malware Unpacking

This process has the task to recover the original code, which has been packed, from a packed malware binary. While seeking to achieve this, in this process malware binary is started to run and then its process image is taken. Here, in the image of a process, unpacking routine, as well as original code, is displayed. Then, the image process is analyzed which is also referred to as raw dumped binary. This is the phase of reverse engineering where the original binary code is traced. Although, it is a successful method in some cases, the original code is packed in such a complicated way that it cannot be fully sought [159].

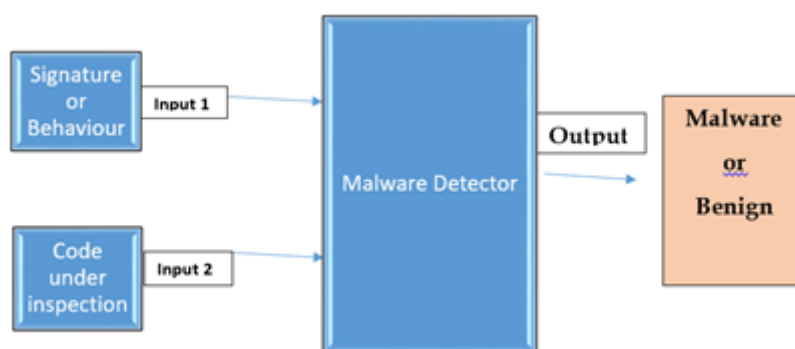


Figure 3.13: Malware detector

To counter evasion there are some anti-evasion techniques which are significant to understand in order to detect malware. The important part is to understand malware's behavioural aspects and execution of malware binary makes it easily possible. On execution of malware, few signs of abnormalities are observable. The quick detection of malware completely relies on the fact that how much is the capability to manage and handle the obfuscated malware. In order to handle this problem, there is a component utilised which is known as malware detector. It can be defined in these words; malware detector is a system which utilises signatures and various other heuristics parameters to detect malware. Since Polymorphism and Metamorphism are the two most popular obfuscation techniques, therefore, the malware detector acts as a safeguard and to DE obfuscate this malware. For that sake, along with DE obfuscating techniques, reverse engineering methods are also used which usually initialize process with any static program analysis. There are two major input parts of the malware detector. Signature or behavioural parameters of the given code, Executable code under inspection

Henceforth, these two input components are obligatory for the malware detector to conclude the code either as malware or benign. This process is shown in Figure 3.13.

3.8.4.2 Malware Normalization

This is the process where an obfuscated version of Malware is accepted and deobfuscates the program and then, ultimately, executables are normalized. This technique raises the detection rate. The overall process involves the following steps [159].

- Step 1: In the first step, decompression software decompresses Malware PE binary code.
- Step 2: In this step, the decompressed code is disassembled by using standard disassemblers.

- Step 3: In this phase, the normalizer takes the disassembled code where obfuscation is eliminated and after certain checks, the normalized code is produced.
- Step 4: Afterwards, a sign of normalization code is extracted by the malware detector after seeking the normalized code and then it is compared with the signatures already present in the signature repository.
- Step 5: This comparison is carried out by the highest level of signature matching of normalized signature code with the repository signature. For this purpose, any type of sequence alignment algorithm can be used. In the final stage, the database keeps the normalized code sign in the record in order to use it in future for the sake of comparisons with other similar variants.



Figure 3.14: Malware Normalization and Signature Comparison [70]

4. The quest for privacy in the IoT

4.1 Introduction to IoT and its Applications

As Cyber-Trust ecosystem builds upon the IoT technologies, it is important to highlight some of IoT common applications and existing concerns that related to the data privacy. IoT devices are implemented in Smart Homes, Health Sector as well as environmental and public safety sector. This chapter investigates the state-of-the-art for IoT applications as well as highlighting the need for privacy-preserving data mining and presenting common Heuristic and cryptographic techniques and tools to overcome challenges of privacy in the data collected by Cyber-Trust components on WP6 as well as in other interconnected components.

4.1.1 Smart Homes

The IoT introduces significant benefits for smart home applications over conventional communication technology [38]. The smart home is built mainly by IoT devices, the environment of these homes enables people to control and monitor their homes remotely by connecting their physical devices via the internet, even power consumption can also be accessed and monitored. The smart home is probably where there are likely to encounter internet-enabled things. Smart Homes also include other components like smart speakers, cameras, smart plugs, thermostats, light bulbs, and smart fridge [124]. As the smart home utilises internet connectivity, this introduces security and privacy-related issues. For Smart Homes, internet service providers might have control of the inclusive collected and operational information and monitor the user's behaviours with or without the consumer's consent. For example, RFID based sensors also can be used with the network to control the Smart Homes as well as to track objects in Smart Homes and monitor the conditions [137]. As data can be gathered and transmitted over wireless channels, and able to understand and monitor the pattern and behaviour of consumers activities without any difficulty. However, still there is a chance of information leak, and it can be stolen by a third party [124].

4.1.2 Healthcare

The application of IoT in medical health is enormous. For example, the objective of eHealth is to improve the health quality, efficiency and cost of health care. This helps physicians to monitor patients remotely, along with allowing patients to control and manage individual health records efficiently likewise monitoring heart rate, blood pressure, diabetes level and fitness level [38]. As the IoT has improved the medication and health care service, it also imposes some issues. These health records are exchanged through the internet. Their accessibility and availability on internet initiated some serious privacy issues. For instance, in June 2015, a malware conceded into blood gas analyser and a huge privacy-violation attack occurred when it gained access to confidential health records [124, 39].

4.1.3 Supply Management

In supply management, IoT-enabled technologies are a vital role. It provides the consummate ability of computer systems or software to exchange and make use of information throughout the product lifecycle. This utilises some of the RFID techniques. It enables to record the product information beyond the manufacturing level to the purchasing and consumption levels [137].

Consequently, manufacturing companies can track customer information based on that product information. Moreover, IoT through intelligent transportation system (ITS) plays a major role in vehicular ad hoc networks. The daily electricity consumption from a grid can be monitored and managed through the application of IoT technology [38]. Such information can be used to reveal the habits and behaviours of consumers and expose them to privacy invasions [124].

4.2 The need for Privacy-preserving data mining

Data Mining is the process of extracting information from a large amount of data and present interesting information from large amounts of data stored either in databases, data warehouses, or types of data repositories. This extracted information can be used for many purposes; this includes decision making, process control, and information management [75]. Therefore, data mining is considered one of the most important frontiers in database systems and one of the most promising interdisciplinary developments in the information industry especially with the emergence of artificial intelligence and machine learning technologies. As the aim of data mining is to extract valuable information from larger datasets, there are more chances of vulnerability and misuse in which user privacy can be misused or exploit [157]. As a result, privacy-preserving has become a more significant concern concerning the success of data mining [95]. Privacy-preserving data mining (PPDM) ensures the protection of the individuals' data privacy or sensitive knowledge without losing the value of the information. Consumers have become aware of the privacy intrusions on their data and are very hesitant to share sensitive information. This may lead to the unintentional results of data mining. To maintain privacy along with data mining, several methods have been developed to counter such issue [157, 155]. As IoT devices generate a huge volume of data, this postures new open research challenges on security and privacy issues. The distribution of IoT services is intended for the goal to enhance the privacy preserving in consumers private life. These objectives can be accomplished by assuring users' authentication, data confidentiality, data integrity and anonymity levels.

- **Users Authentication:** Authentication process identifies the user or the object which are authorised to access the information by means some mechanisms, likewise password, digital signature, challenge and response and so on. Deploying an effective authentication measures ensure only those authorised objected can access the confidential information and manage, change IoT data.
- **Data confidentiality:** Data confidentiality involves a set of rules or a promise usually executed through confidentiality agreements that limits access or places restrictions on certain types of information. This can be accomplished by means such as the adoption of various encryption schemes, such as Rivest, Shamir, and Adelman cryptosystem (RSA).
- **Data integrity:** Data integrity validate any unauthorised changes to the original information. In the context of IoT, integrity can be preserved by leveraging hash algorithms and utilising existing of technologies such as blockchain.
- **Anonymity Level:** A privacy policy defines how data referring to individuals can be collected, processed and diffused according to the rights that individuals are entitled to [156]. Depending on the specified purpose, a certain level of anonymity may be guaranteed. The anonymity represents the absence of identifiable data of a user or of data that allows inferring identifiable data (e.g., first name, surname etc.). For example, from the user's point of view, it is crucial to guarantee that their privacy is not violated. In order to avoid such violation of user privacy, a key role is played by the correct definition of privacy policies and the related defined confidentiality and anonymous level of information. The main reason that makes privacy a fundamental IoT requirement lies in the envisioned IoT-aided application domains.

A framework for privacy-preserving data mining is proposed by the researcher [95] is shown in Figure 4.1. Data from different data sources or operational systems are collected and are preprocessed using extract, transform and load (ETL) tools. This transformed and clean data from the level 1 is stored in the data warehouse. Data in a data warehouse is used for mining. In level 2, data mining algorithms are used to find patterns and discover knowledge from the historical data. After mining, privacy preservation techniques are used to protect data from unauthorised access. Sensitive data of an individual can be prevented from being misused [157].

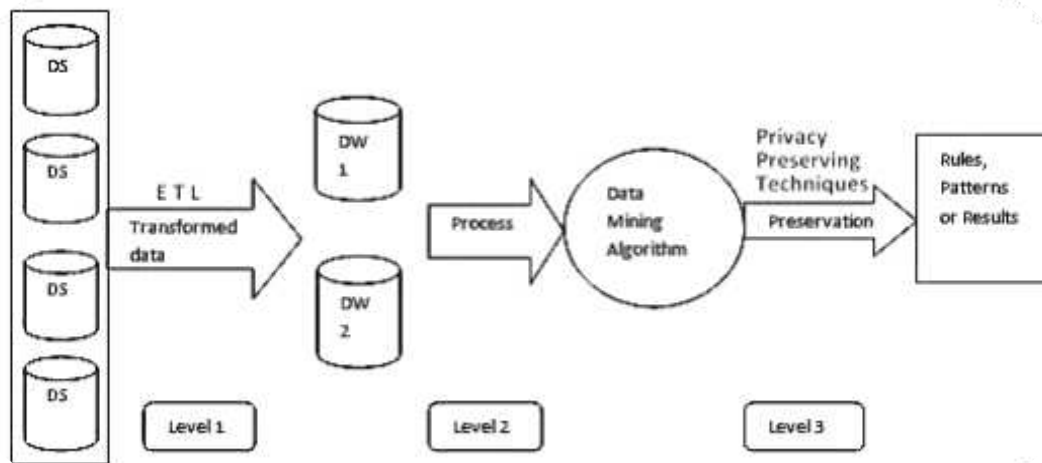


Figure 4.1: The framework of privacy-preserving data mining [95]

As discussed in the previous section the IoT is a multi-domain technology with a network of devices and services to exchange information. Each domain can apply its security, privacy, and trust requirements [94]. The privacy-preserving data mining techniques, propose the setup for minimising the risks of revealing sensitive information and provide sensitive content analysis. It introduced some of the privacy and security issues in IoT technological aspects. Some of the issues, in perspective of users, datasets and fundamental technologies are discussed. There are some of the major issues related to IoT privacy in organisations over the technical challenges. Figure 4.2 summarises the four critical aspects of IoT privacy [124]. IoT devices become part of the internet network, and the generated data is transmitted and exchanged over the internet, rendering user privacy a sensitive subject in many research works [140, 105]. Although an abundance of research has already been proposed concerning privacy, many topics still need further investigation. Privacy in data collection, as well as data sharing and management, and data security matters remain open research issues to be fulfilled [5].

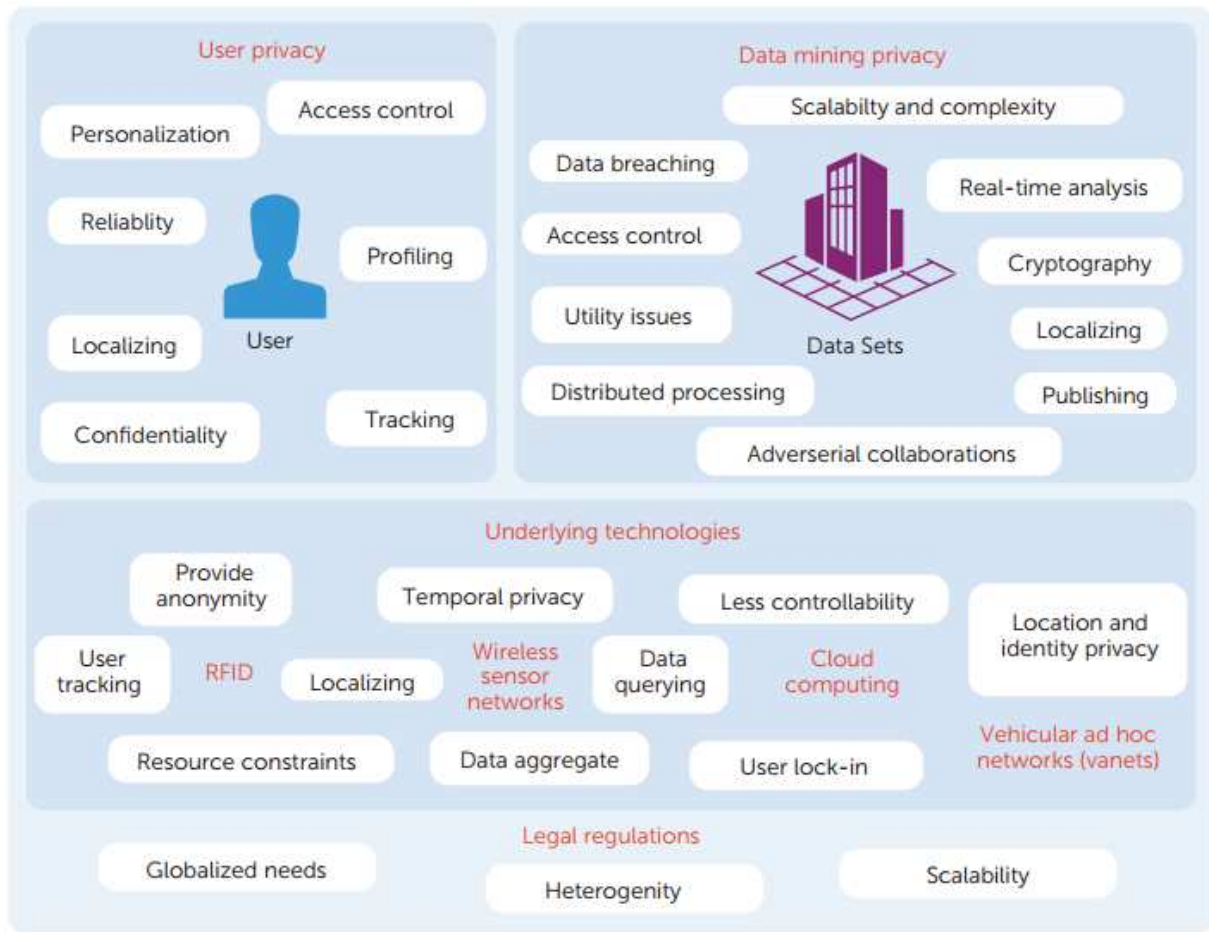


Figure 4.2: Summary of the critical aspects of IoT privacy [124]

The user, privacy awareness issue, is addressed, proposing a privacy management scheme which enables the user to estimate the risk of sharing sensitive data. It also aims at developing a robust sensitivity detection system, able to quantify the privacy content of the information [155]. In order to establish more secure and readily available IoT devices and services at low cost, there are many security and privacy challenges to overcome [94]. Some of these challenges for IoT privacy issues are discussed in the following subsections.

4.2.1 User Privacy

One of the serious issues of IoT data mining is the user privacy issue. It is the identification of personal information during communication over the Internet [137]. For instance, if a consumer buys an RFID-tagged object in some situations, a consumer's personal information could be automatically linked to the object and known to the communication system providers. Such user information leakage can lead to privacy threats in terms of tracking, localising, and personalisation.

Similarly, assume consumer possesses a set of objects that are linked together. If adversaries can distinguish ownership of individual objects, they might be able to estimate the ownership of the remaining objects. These types of scenarios allow user profiling and tracking to be vulnerable to such security breach.

Smartphones and other mobile devices connected to the Internet could also disclose the user's sensitive information such as geographic location and compromise privacy. In practice, users have different levels of privacy awareness and concern, and thus are ready to disclose information at different levels. In general, IoT users might encounter privacy threats in terms of tracking, profiling, access control and confidentiality, data protection, content confidentiality and reliability, and privacy detection. Because of the IoT's range, various privacy risks and challenges must be considered before deploying an application or solution.

4.2.2 Privacy Issue in Data Mining

Other privacy issues identified in this area relate to data mining, the context of applications, utility issues, cryptography, and adversarial collaboration [24]. Scalability matters for IoT applications that contain numerous smart objects or that manage biometric data are in some cases collected, processed and stored in large volumes of real-time, highly distributed data. Distributed processing can also lead to unprecedented challenges related to data mining privacy, along with liability for data breaches (that is, the release of secure information to distrustful entities) in which distinct levels of data quality. Privacy threats related to data sharing and transmitting arise with the disclosure of location and temporally sensitive data traffic. While collecting large sets of raw data, it is challenging to balance the privacy preservation in data cleaning and the intentional reduction of data quality and original purpose without losing information needed for data mining and analysis. Collecting, sharing, and transmitting sensitive data connected to humans are the most critical privacy issues in the context of applications. Computational and theoretical limitations can be associated with privacy preservation over high-dimensional datasets. Because individuals and cooperative users have different privacy constraints, the records in a given dataset should be treated differently for anonymisation purposes. The collected data might be used and published for purposes other than the original objective without user consent. Access control and maintenance of such data, with the assurance of privacy protection for the corresponding data owner, should be carefully considered. As computer storage mediums can store large volumes of data, they offer high availability at low cost. Consequently, once information is generated, it is most likely stored infinitely, and thus “digital forgetting” can lead to privacy violations from the data owners’ perspective [124].

4.2.3 Confidentiality Issues in Data Mining

A critical problem that arises in any collection and mining of data is the confidentiality of the data being collected. The need for privacy is sometimes due to the law (e.g., for medical databases). Data confidentiality represents a fundamental issue in IoT scenarios, indicating the guarantee that only authorised entities can access and modify data. In the IoT context, not only users but also authorised objects may access data. This requires addressing two important aspects: first, the definition of an access control mechanism and second, the definition of an object authentication process, with a related identity management system [38]. As data in IoT applications will be related to the physical realm, ensuring data confidentiality is a primary constraint for many use cases like Smart Homes and healthcare.

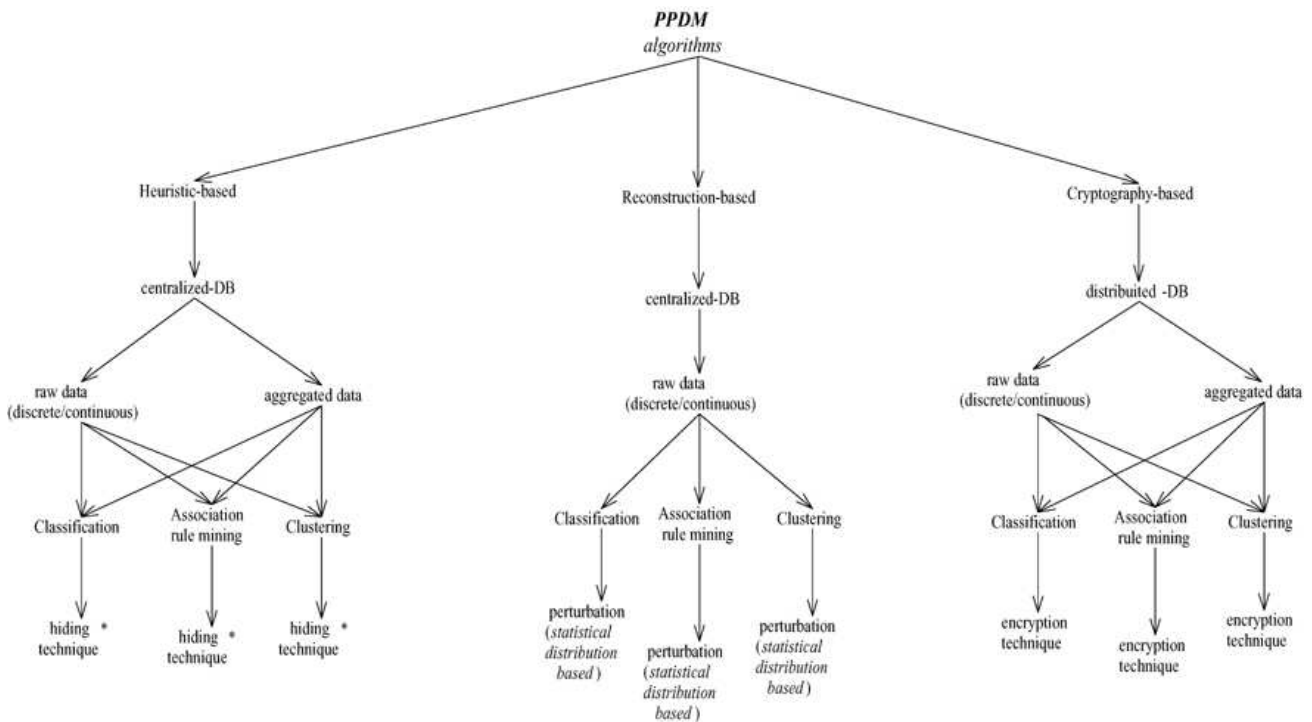
4.2.4 Semi-Honest Adversaries

When there is data being transmitted or mined between two or more parties, a malicious adversary who is able to intercept the channel could alter the data. In which this could lead to a very damaging effect. Then the output obtained is the result of the algorithm on the other party’s database alone. Although this attack cannot be prevented, we would like to prevent a malicious party from executing any other attack. However, for this initial work assuming that the adversary is semi-honest (also known as passive). It correctly follows the protocol specification yet attempts to learn additional information by analyzing the transcript of messages received during the execution. This remark that although the semi-honest adversarial model is far weaker than the malicious model (where a party may arbitrarily deviate from the protocol specification), it is often a realistic one. This is because deviating from a specified program which may be buried in a complex application is a non-trivial task. Semi-honest adversarial behaviour also models a scenario in which both parties that participate in the protocol are honest. However, following the protocol execution, an adversary may obtain a transcript of the protocol execution by breaking into one of the parties’ machines [178].

4.3 Heuristic-Based Techniques and tools

Privacy preserving data mining is a trending area of research in data mining. The possible side effects of data privacy are analysed in the data mining algorithm. The governing aim of PPDM is to develop techniques and approaches for amending the original information in such a way that the sensitive data remains private even

after the mining process occurred [157]. For this reason, there is a need to develop mechanisms that can lead to privacy control systems to convert a given database into a new one in such a way to preserve the general rules mined from the original database. The procedure of transforming the source database into a new database that hides some sensitive patterns or rules is called the sanitisation process [151]. For instance, a small number of transactions must be modified by deleting one or more items from it or even adding noise to the data by turning some items from 0 to 1 in some records. The resulted database is called the sanitised database. On the one hand, this approach slightly modifies some data, but this is perfectly acceptable in some real applications. Moreover, heuristic approaches can also be classified into distortion-based schemes and blocking based schemes. These approaches have been getting the focus of attention for the majority of the researchers due to their efficiency, scalability and quick responses [107]. Figure 4.3 delineates the catalogue of the PPDM algorithm in the consent of heuristic and cryptography-based approaches. It does not cover all of the new PPDM algorithms [45]. In this section, different PPDM approaches in IoT, likewise data perturbation, blocking based, cryptographic techniques are discussed.



* hiding technique = {perturbation, blocking, swapping, aggregation, generalization, sampling}

Figure 4.3: A classification of the developed privacy preserving data mining algorithms [45]

4.3.1 Data Perturbation

Data Perturbation is a technique for amending data using a random process. This technique deceptively alters sensitive data values by altering them by adding, subtracting or any other mathematical formula [77, 64]. This technique can handle different data types, such as character type, Boolean type, classification type and integer. In discrete data [77], there is a requirement of preprocessing the original dataset. The preprocessing of data is categorised into attribute coding and obtaining sets coded dataset. The method of the average region to disperse the continuous data is used in this approach. Data distortion or data noise are different names for data perturbation. Distortion is accomplished by applying different methods such as adding noise, data transpose matrix, adding unknown values [166]. In some perturbation methods, it is challenging to preserve the original data fully. Some of these are distribution-based techniques. In order to overcome this problem, the new algorithm, rule mining and the distribution-based algorithm were developed which are able to reconstruct the distributions. This means that for every individual problem in classification, clustering, or association rule mining, a new distribution-based data mining algorithm needs to be developed.

4.3.2 Cryptographic Technique

Cryptography is a technique through which sensitive data can be encrypted and becomes unreadable. In which it can be leveraged to preserve sensitive data. In [178], the authors introduced a cryptographic technique that provides security and protection of sensitive attributes. There are different cryptographic algorithms exist (i.e. RSA, AES). Although cryptographic schemes protect the confidentiality and privacy of the data from leakage, however, cryptographic-based approaches have some disadvantages such as that it is computation expensive. Therefore, it is challenging to apply cryptographic algorithms for huge databases.

4.3.3 Blocking based technique

In blocking-based technique [152, 2], authors declare that there is a sensitive classification rule which is used for hiding sensitive data from malicious intruders. In such a technique, there are two steps that are used for preserving privacy. First, is to identify transactions of the sensitive rule. Second, is to replace the known values to the unknown values. In this technique, there is scanning of original database and identifying the transactions supporting sensitive rule.

Moreover, then for each transaction, the algorithm replaces sensitive data with unknown values. This technique applies to those applications in which one can save unknown values for some attributes. Authors in [152], hide the actual values and replace '1' by '0' or '0' by '1' or with any unknown values in a specific transaction. The replacement of these values does not depend on any specific rule. The main aim is to preserve the sensitive data from unauthorised access. There are different sensitive rules according to the requirements. For every sensitive rule, the scanning of the original database is required. When the left side of the pair of rules is a subset of attribute values pair of the transaction and the right-hand side of the rule should be same as the attribute class of the transaction then the only transaction supports any rule. The algorithm replaces unknown values in the place of an attribute for every transaction which supports that sensitive rule. These steps will continue until the unknown values hide all the sensitive attributes.

4.3.4 Condensation Approach

Another approach used is condensation technique. Charu C. Aggarwal introduced it and Philip [24], which builds constrained clusters in the data set and after that produces pseudo-data. The basic concept of the method is to contract or condense the data into multiple groups of predefined size. For each group, individual statistics are maintained. This approach is used in dynamic data update such as stream problems. Each group has a size of at least 'k', which is referred to as the level of that privacy-preserving approach. The higher the level, the higher is the amount of privacy. They use the statistics from each group in order to generate the corresponding pseudo-data. This is a simple privacy preservation approach, but it is not very efficient as it could lead to loss of information.

4.3.5 Hybrid technique

This is the techniques through which one can combine two or more techniques to preserve the data. The authors [152] proposed a hybrid technique in which they used randomisation and generalisation. In this approach first, they randomise the data and then generalised the modified or randomised data. This technique protects private data with better accuracy. Also, it can reconstruct the original data and provide data with no information loss. Many other techniques can also be combined to make a hybrid technique such as data perturbation, blocking based method, cryptographic technique and condensation approach etc.

4.3.6 Data Anonymization

Anonymisation methods have an essential tool to preserve privacy when releasing sensitive data set from a larger volume of data. Most survey says common type of attack for anonymisation algorithms is based on PPDM and PPDP is presented in [81] and their data privacy is explained. There are different tools for data anonymisation these include:

- Oracle Advanced Security, Oracle
- IBM Security Guardium
- IBM Dynamic Data Masking Informatica
- Micro Focus Data Express, Micro Focus
- IMASK, Mentis
- CA Data Manager
- CA Technologies Compuware
- IRI Field Shield , IRI
- Data Base Protector Protegrity
- Thales eSecurity, Thales
- Soflab GALL, Soflab Technology
- Privitar Publisher, Privitar Ltd

De-anonymization is the reverse process in which anonymous data is cross-referenced with other data sources to re-identify the anonymous data source. Generalisation and perturbation are the two popular anonymisation approaches for relational data.

4.4 Cryptography-Based Techniques and tools

Cryptography provides tools for privacy-preserving computations, which are closely related to privacy-preserving data mining. For example, in [52] a privacy-preserving social network analysis is being proposed, which allows several crime investigators to collaborate without actually exchanging “sensitive” private information, since the investigator can compute important metrics by means of a social network analysis while keeping the entire social network unknown; hence, the investigator can request data from other sites to augment his view without revealing personally identifiable data.

Generally, such cryptographic tools include secure multiparty computation, homomorphic encryption and zero-knowledge proofs. There are also several other relative tools such as order-preserving encryption; in this Section though we shall focus on the first three cryptographic primitives since these constitute possible candidates for alleviating privacy issues in the context of the Cyber-Trust project. These approaches are based on the assumption that identities of the users are not hidden, whereas the act of assigning feedback to a user (in the Cyber-Trust case, to a device) is also not concealed; however, the value of the submitted feedback and any other related information is considered private. It should be pointed though that privacy in computations can also be enhanced under a different assumption, namely via considering that the true

identity of the users that submit their feedback is hidden, which in turn means that a user is being associated with one or more pseudonyms which are unlinkable to his real identity. Again, cryptographic primitives for appropriately deriving such pseudonyms can also be used (e.g. several design goals, depending on the context, can be considered, such as user-pseudonym unlinkability or pseudonym-pseudonym unlinkability – a general survey, focusing on the specific case of privacy-preserving reputation systems, is provided in [117]).

4.4.1 Secure Multiparty Computation

Distributed computing is the scenario where a number of computing parties carry out a combined computation of a certain operation. For example, these parties may be devices or servers that maintain a distributed system while the operation could be the database update. The main goal of secure multiparty computation (SMC) is to allow the participating entities to carry out distributed computing operations in a secure manner. Considering the two-party problem where two entities, with secret inputs X and Y respectively, wish to compute the median of the union $X \cup Y$ without revealing any information except from the output [11]. The two parties in order to perform this task, execute an interactive protocol, by sending messages to each other and the outcome of this operation is to obtain the desired output.

4.4.2 Security in the multiparty computation

Since, distributed computing is about computing under the threat of machine failures and other **accidental incidents**, SMC is concerned with the probability of adversarial behaviour. This means that, it is a common phenomenon that the protocol in a single execution to come under attack by an external adversary or a subset of the participating entities. The adversary or the entities that have under its influence, executing such operations usually aim to obtain private information or cause the combined computation to fail and lead to incorrect output. Secure protocols should endure any adversarial behaviour and thus, various security properties have been introduced that aim to claim and prove security for multiparty computation:

1. **Privacy:** Except for the combined output, no party should obtain any other information about the other parties' input.
2. **Correctness:** The output that each party receives, is certain to be correct.
3. **Independence of Inputs:** The parties that are under the influence of the adversary, select different inputs from the inputs of the honest parties.
4. **Guaranteed Output Delivery:** The adversary should not be capable to disrupt the computation and prevent any party from receiving its input (e.g. by executing a "denial of service attack").
5. **Fairness:** The scenario where the adversary and the parties that have under its influence obtain their outputs if and only if the honest parties receive theirs, too.

It is of great importance the fact that these properties do not constitute a definition of security but are a set of requirements that any protocol must hold in order to be secure. Furthermore, it is assumed that the adversary controls less than the 50% of the total number of the participating entities. Otherwise It is difficult to maintain the properties of the protocol for SMC [30].

4.4.2.1 Adversarial power.

The previously mentioned security properties omit a scenario where the adversary and the parties that are under its influence attack the protocol. Thus, it is of great importance that the adversarial capabilities such as: the power that it possesses, its complexity and the corruption methods to be extensively explained.

1. **Corruption strategy:** The adversary is capable to corrupt parties using either (a) the *static corruption model*, in which the number of the parties that it has under is influence are fixed for the entire execution of the protocol, or (b) the *adaptive corruption model*, in which the corruption of the parties during computation is allowed. In the second case, the adversary is enabled to choose which party to corrupt, and the time to do so.

2. **Permitted adversarial behaviour:** There is a case, where the adversarial actions do not aim to harm the other entities of the systems. In the *semi-honest adversarial model*, the corrupted parties, just like the honest, follow the protocol, but the adversary obtains the internal state of the *semi-honest parties* that has under its influence. That means that, the adversary acts as an eavesdropper in order to obtain private information, and thus this model is called “weak adversarial model and the corrupted parties are named “honest-but-curious and passive”. The second and obvious case is when the corrupted parties act malicious and under the adversary’s instructions. Thus, it is crucial to provide security in the presence of malicious adversaries.
3. **Complexity:** The computational complexity of the adversary is about its capability to run in *Polynomial Probabilistic Time* (PPT) or to be *computationally unbounded*. This distinction regarding the adversary yields two models for secure computation: the information-theoretic model [16], [31], and the computational model [62], [180]. In the information-theoretic model the adversary is not bound to any complexity and is not running in polynomial time but is capable to eavesdrop and interfere when honest parties communicate. In the other case, the adversary is assumed to be a PPT algorithm while, the results in the computational setting assume cryptographic assumptions such as trapdoor permutations.

4.4.2.2 Feasibility of secure multiparty computation.

The above-described properties of security are restrictive, and it seems that it is tolerated no adversarial behaviour. Thus, to obtain secure protocols, in the presence of adversaries, powerful feasibility results demonstrate that any distributed computing task can be computed securely. Denoting as m the total number of the parties in a protocol, and as t the number of the parties under the influence of the adversary:

1. $t < \frac{m}{3}$. If the number of the adversaries is less than 33% of the total number of the participating entities in the protocol, then secure multiparty computation can be achieved, in both computational and information-theoretic setting, satisfying *Guaranteed Output Delivery* and fairness [90].
2. $t < \frac{m}{2}$. If the honest majority is guaranteed, then then secure multiparty computation can be achieved, in both computational and information-theoretic setting, supposing that the participating entities have access to a broadcast channel [90].
3. $t > \frac{m}{2}$. If the adversaries exceed the number of honest parties, then secure multiparty computation can be achieved, in the computational setting only, by assuming that the participating entities have access to a broadcast channel [90], and the existence of enhanced trapdoor permutations [62], [180].

Summarizing, secure multiparty computation is feasible for any distributed task. In the computational setting this is possible for any number that the adversary has under its influence, but if there is no honest majority, then the properties guaranteed output delivery and fairness are not obtained.

4.4.3 Homomorphic Encryption Techniques

Before being stored to a possibly untrusted repository, private information should first be encrypted; e.g. in Cyber-Trust this could be smart home owner’s data that are being stored at the ISP. However, this might be challenging if the stored data need also be used for performing various computations in a secure manner. For instance, if the Cyber-Trust platform needs to perform a query about the vulnerabilities of a device and the data are encrypted (using an ordinary encryption scheme), then getting back meaningful results will not be possible without having access to the decryption key. On the other hand, if the third party is malicious or gets compromised and the data are not encrypted, then the attacker can easily get access. However, many encryption schemes are homomorphic and allow anyone to manipulate encrypted data, even if the secret key is not known. Rivest Adleman, and Dertouzos are the first who posted this dilemma [142] which until 2009 remained unsolved, where Craig Gentry introduced the Fully Homomorphic Encryption (FHE) scheme using ideal lattices [57] that allows outsourcing computations securely.

4.4.3.1 Homomorphic Encryption

A simple example of homomorphic encryption is concatenation which entails putting two messages or even ciphertexts in a sequence, that is, it is possible to encrypt first and then concatenate the encrypted messages or to concatenate first and then encrypt the new message and the result to be the same. Although, concatenation is not a very interested operation, provides random computations on encrypted information and can answer the dilemma. In a homomorphic encryption scheme, it is necessary to perform two functions, which are addition and multiplication.

4.4.3.2 Somewhat Homomorphic Encryption

There are many partially homomorphic encryption schemes and even RSA [143], the first public key cryptosystem that enables the computation of the sum of two plaintexts and El Gamal's encryption scheme [49] that computes a message product, belong to this category. For many years it was necessary to choose between addition and multiplication but these operations were not adequate for random computations. The first advance was to invent "somewhat homomorphic encryption" for arbitrary computations [19], but these encryption schemes inevitably add noise (Figure 4.4).

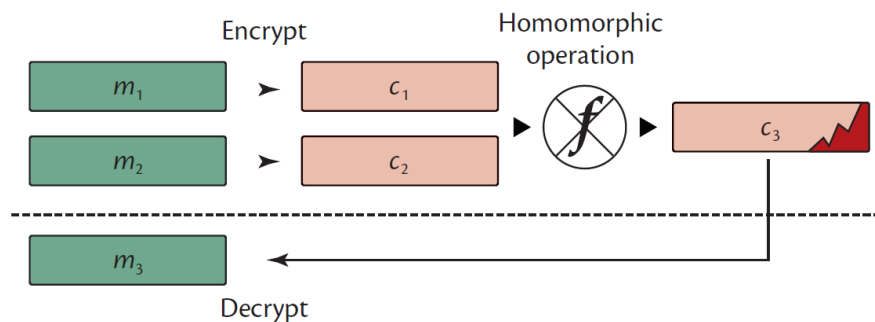


Figure 4.4: Somewhat Homomorphic Encryption [92]

A small amount of noise does not affect the output of the decryption but the repeated encryption of the plaintext may render to an undecipherable ciphertext. To put it simply, each operation in a somewhat homomorphic encryption scheme comes with a cost, but for simple encryptions the noise just vanishes on the decryption process. The search for an encryption method that was not plagued was conducted for almost twenty years and various ideas were introduced until Gentry's revolutionary insight.

4.4.3.3 Fully Homomorphic Encryption

Fully homomorphic encryption (FHE) scheme, means that there are no limits on the manipulations that can be executed. With known ciphertexts c_1, \dots, c_t that encrypt the messages m_1, \dots, m_t under a given key and any computable function f , it is possible to compute a ciphertext that encrypts $f(m_1, \dots, m_t)$. To put it simply, this allows general computations on encrypted data without leaking any information about the messages m_1, \dots, m_t or the value of $f(m_1, \dots, m_t)$.

The key insight behind Gentry's idea was to invent an encryption scheme with very few operations and low complexity and thus the decryption process to be conducted homomorphically [58]. Such schemes are called "bootstrappable" [57] and the basic idea behind this technique is to eradicate the noise, by partially decrypting the ciphertext and re-encrypting to perform additional computations (Figure 4.5).

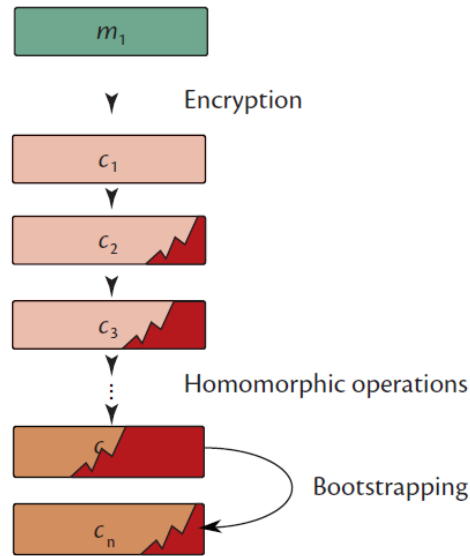


Figure 4.5: Fully Homomorphic Encryption (FHE) [92]

The next step is to construct a bootstrapping or refreshing process. The decryption process is performed homomorphically using an encrypted secret key and aims to remove the noise, but by doing so, some noise is also reintroduced, but less than it is removed. Besides the fact that bootstrapping is a public operation, without revealing the secret key there is no concern to compromise security. The double encryption process is the starting point from which the noise is cleaned from the ciphertexts. It is of great importance to ensure that the noise does not exceed a certain threshold during homomorphic decryption for further operations to be allowed.

4.4.3.4 Limitations and Generations

Since, bootstrapping operations are very costly and ciphertexts are too large, FHE is not so practical and not being used widely. C. Gentry and S. Halevi, Implemented FHE ranging the size of the public key from 70 MiB to 2.3 GiB and the results showed that to run just a single bootstrapping operation took from 30 seconds to 30 minutes, respectively [55]. Nevertheless, the efficiency of the implementation also presented feasibility and lead the first studies to focus on reducing the complexity of the scheme [20], [164], as well as the improvement of efficiency [BR12], [56], [21]. The latest version of Gentry scheme [59], resulted in improvements that enabled bootstrapping in 0.61 seconds but RSA encryption and decryption can be performed in less than a millisecond using the same machine.

4.4.4 Zero-knowledge proofs

Zero-knowledge (ZK) proofs [148] have a long history in cryptography, having used in several frameworks (see also the Deliverable D7.1). They allow a user to prove that she acquires some specific secret knowledge, without revealing this knowledge; more generally, an interactive proof allows a prover to convince a verifier that a statement is true without revealing any information other than the fact that the statement is valid. Therefore, zero-knowledge proofs could be considered as a private enhancing technology, since their inherent nature rests with revealing the least possible information. For example, as described in [51], ZK proofs could be an appropriate technical solution for alleviating several privacy issues arising in a smart metering system; indeed, since storing and further processing of the whole customer's profile (with respect to, e.g. water or electricity consumption) raises privacy concerns, a ZK proof can be used to allow the user prove to the service provider that all the relevant measurements have been truthfully computed – which is prerequisite for issuing a valid bill – without revealing the actual measurements themselves.

A zero-knowledge proof must satisfy four properties:

- **Completeness:** if the statement is true, the honest verifier will be convinced of this fact by an honest prover.
- **Soundness:** if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.
- **Zero-knowledge:** if the statement is true, no verifier learns anything other than the fact that the statement is true.
- **Polynomial verification:** The verifier must do his private computation in polynomial time.

In the traditional form of ZK proofs, an interactive approach is being followed – that is the verifier sends several challenges to the prover, whereas the latter provides the responses which can be computed correctly in case that the prover acquires a secret knowledge. More precisely, an interactive proof for a decision problem can be generally described as follows:

1. There are two participants, a prover P and a verifier V
2. The whole process (proof) consists of a well-determined number of rounds.
3. In the beginning, both participants get the same input.
4. In each round, the verifier challenges the prover, and the prover responds to the challenge.
5. Both the verifier and the prover can perform some private computation
6. In the end, the verifier states whether he is convinced or not.

Ideally, given any input x , the following property is desirable: anything that the verifier V can compute efficiently after the interaction with the prover P on x , could also be computed before the interaction. To this end, we need to show that V could generate the same interaction (i.e. the same “dialogue”) without the prover’s help, and that the distribution of the generated interactions is identical to the distribution of the real interactions. In such a case, we refer to perfect zero-knowledge proofs. However, perfect zero knowledge is a very strong requirement, and therefore we might be interested in a weaker model, which can be applied to a wider set of problems – that is the distributions above are statistically indistinguishable (i.e., the statistical distance between the distributions is negligible). In such a case, we refer to almost-perfect or statistical zero-knowledge (while the notion of computational zero knowledge is also being used) [15].

There are many varieties of zero-knowledge proofs. Non-interactive zero-knowledge proofs were introduced in [98] which was a significant improvement in terms of efficiency. However, zero-knowledge proofs have started to be considered more practical after subsequent works, such as [74], [136], [42], which determined the so-called Succinct Non-Interactive Zero Knowledge Proofs (zkSNARKs). These models allow getting proofs of any statement efficiently. ZkSNARKs have also been described in deliverable D7.1 (Distributed Ledger state-of-the-art report), since they fit well in blockchain applications that are being discussed therein.

4.5 Reconstruction-Based Techniques and tools

Some benefits of the information technologies paired with IoT devices are only possible through the collection and analysis of (sometimes sensitive) data. However, this may result in unwanted privacy violations [139]. In order to protect the owner’s exposure from information leakage, researchers proposed new effective data mining techniques that hide (by modifying or even removing) sensitive information from the original data [25, 26]. However, transforming the data may reduce its utility, resulting in inaccurate or even infeasible extraction of knowledge through data mining. It is obvious thus, that the balance among data privacy and data utility is very fragile (Figure 4.6).

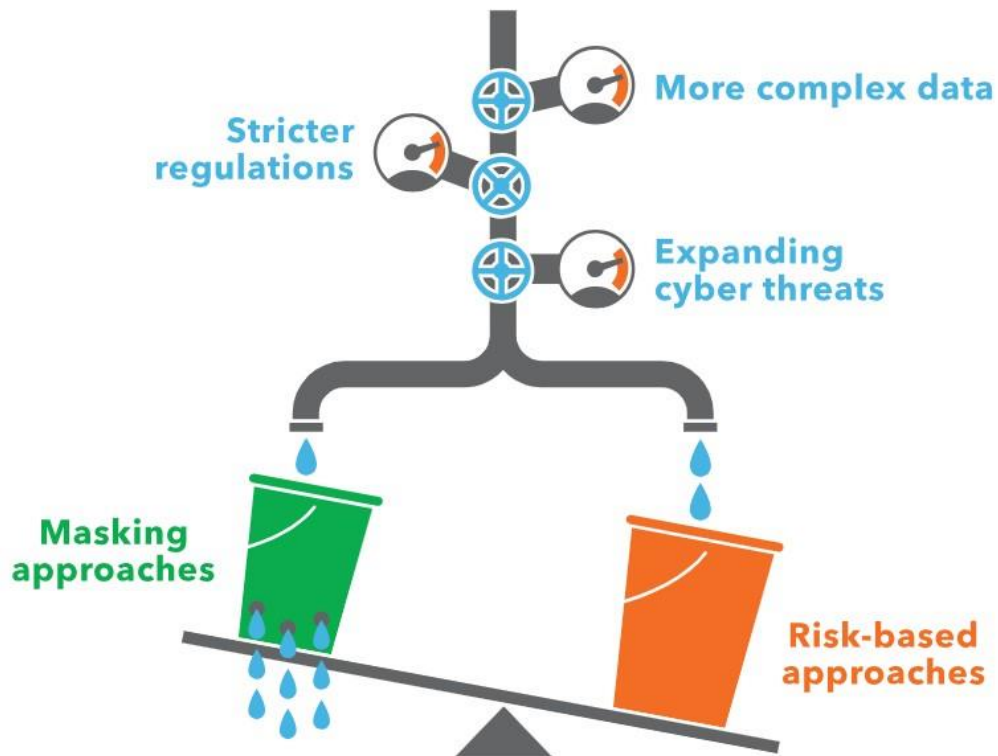


Figure 4.6. The fragile balance among data privacy and data utility (source [68])

To this end, the research community designs and proposes methodologies to extract knowledge from data while ensuring a certain level of privacy; this paradigm is known as Privacy-Preserving Data Mining (PPDM) [139, 135] and comes with the following objectives: (1) hide sensitive information contained in the original data, (2) keep the same characteristics between hidden and original data, and (3) get the same data accuracy as in the original data set. Figure 4.7 schematically presents the mechanism of PPDM.

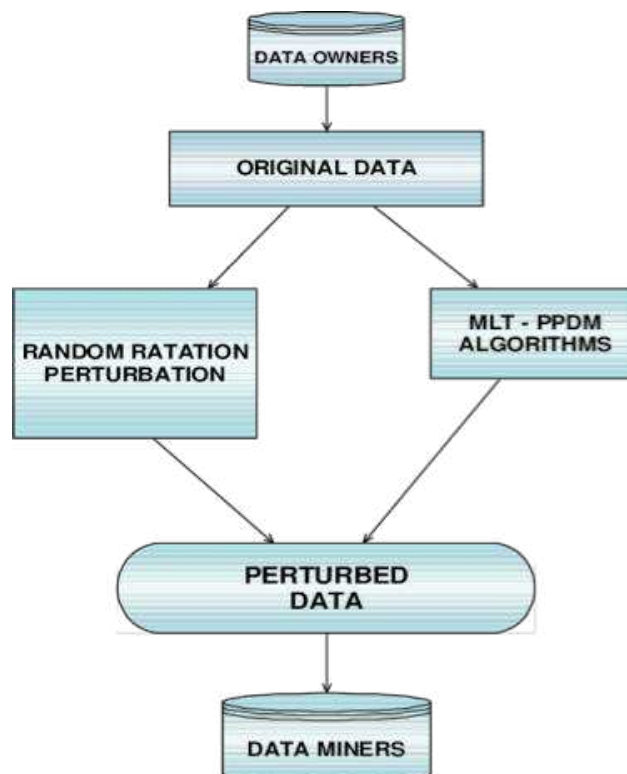


Figure 4.7: The mechanism of PPDM

Naturally, due to the explosion of smart devices and IoT in recent years, PPDM has drawn extensive attention amongst researchers, resulting in numerous techniques for privacy under different assumptions and conditions. The vast majority of the PPDM techniques modify or even remove some of the original data in order to preserve privacy [25]. This data quality degradation is the natural trade-off between the privacy level and the data quality, which is formally known as a utility. However, modifying or purifying data is an NP-hard problem. Thus privacy protection data mining algorithms apply methods of distortion [176], such as random perturbation, blocking, and condensation, to deal with the challenges imposed by PPDM.

More specifically, to extract knowledge from data while preserving privacy, PPDM encompasses techniques that use (1) data transformation, such as additive noise [135] and multiplicative noise [76], (2) primitives for adjusting the privacy-utility tradeoff of more evolved data mining, such as privacy models like k-anonymity [126, 125], l-diversity [3] and others [114, 177, 28], and (3) the more classical data mining approaches, such as association rule hiding [173, 95], downgrading classifier effectiveness [25, 149], or query auditing and inference control [7, 115]. PPDM also accounts for distributed privacy techniques, such as homomorphic encryption [138, 29], or secure sum [27] and others [27, 104, 146, 106] that are employed for mining global insights from distributed data without disclosure of local information. Due to the variety of proposed techniques, several different taxonomies for PPDM methods have been proposed [25, 26, 145, 6, 97, 87], as well as several metrics to evaluate the privacy level and the data quality/utility of the different techniques [44, 145, 43, 147]. Figure 4.8 presents a classification of PPDM techniques based on the location of computation [6].

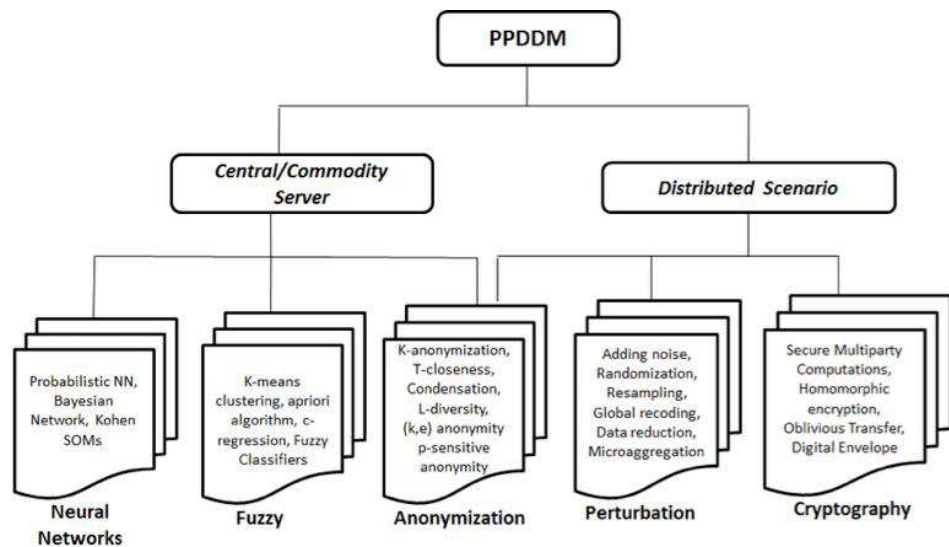


Figure 4.8: Classification hierarchy of PPDM techniques based on the location of the computation

In this report, we are considering a classification of techniques based on the data lifecycle phase as in [26], at which the privacy-preservation is ensured, namely data collection, data publishing, data distribution, and data mining. In any of these phases, it is important to note that the techniques used must both preserve privacy and ensure data quality. However, even at a given data phase, there is no single optimal PPDM technique. The appropriate choice is often a matter of weighing the different trade-offs between the desired privacy level, the information loss, which is measured by data utility metrics, the complexity, and even the practical feasibility of the available techniques. Another aspect to take into consideration is the type of adversarial behaviour and the corresponding privacy breaches that can be explored. Below, we briefly present the different privacy-preservation techniques used considering the different phases of the data lifecycle.

- **Data collection.** To ensure privacy at data collection time, the sensory device transforms the raw data by randomizing the captured values, before sending to the collector. The assumption is that the entity collecting the data is not to be trusted. Therefore, the original values are never stored, but used only in the transformation process. Consequently, randomization must be performed individually for each captured value. Most common randomization methods modify the data by

adding noise with a known statistical distribution, so that when data mining algorithms are used, the original data distribution may be reconstructed, but not the original (individual) values. Thus, the randomization process in data mining encompasses the following steps: randomization at data collection, distribution reconstruction (subtracting the noise distribution from the first step) and data mining on the reconstructed data [26]. Since the original data is modified into perturbed data, specific data mining algorithms, such as clustering and classification, (that can leverage knowledge discovery from distributions of data and not from individual entries) are required [26].

- **Data publishing.** Entities may wish to release data collections either publicly or to third parties for data analysis without disclosing the ownership of the sensitive data. In this situation, sensitive attributes are person-specific private attributes that should not be publicly disclosed (e.g. diseases in medical records) and thus, preservation of privacy may be achieved by anonymizing the records before publishing. The anonymization of records in a database is possible by implementing different privacy models, which attempt to preserve records' owner identity by applying one, or a combination of the following data sanitizing operations: (1) replacement of a value for a more general one (parent), (2) removal of some attribute values to prevent information disclosure, (3) de-association of quasi-identifiers(QIDs) and sensitive attributes in two separate tables making it more difficult to link QIDs to sensitive attributes, (4) replacement of the original data for synthetic values with identical statistical information. Based on these operations, a set of privacy models, such as k-anonymity [126, 125], l-diversity [3], R-susceptibility [79] and other techniques [114-28], has been proposed.
- **Data distribution.** There are situations where multiple entities seek to mine global insights (in the form of aggregate statistics) over the conjunction of all (partitioned) data, without revealing local information to the other entities, which may be possible adversaries. A generalization of this problem is the well-studied secure multiparty computation (SMC) problem from the cryptography field [117]. In such a distributed scenario, a dataset may be partitioned either horizontally or vertically. In the horizontal case, each entity contains different records with the same set of attributes, and the objective is to mine global insights about the data. For example, consider a hospital with different departments, where each department has different patients, and the attributes associated with each patient are common to all departments, such as type of disease and client's QID. In vertically partitioned datasets, entities contain records with different attributes pertaining to the same identity. The junction of the dataset in this case allows to infer knowledge that could not be obtained from the individual datasets; stores with complementary items may be sequentially visited by the same clients, thus creating patterns that would not exist in each store's database. Distributed privacy-preserving algorithms, such as homomorphic encryption [138, 29], secure sum [27], or other techniques [27, 104, 146, 106], exist for both types of partitioning.
- **Data mining.** The outputs of the data mining algorithms may be extremely revealing, even without explicit access to the original dataset. An adversary may query such applications and infer sensitive information about the underlying data. The most common techniques to preserve privacy to the output of the data mining are association rule hiding [173, 95], downgrading classifier effectiveness [25, 149], and query auditing and inference control [7, 115]. Note that in all these methods, the application may be affected: if the utility of the data used to build the application is lower than the original value of the data, the application itself is downgraded, or the access to the data is restricted. Thus, when building an application, one has to choose the technique that best fits his requirements considering that the trade-off between privacy and utility is always present.

5. Conclusion

This deliverable performed a thorough review of the current state-of-the-art in Cyber-Trust's profiling, detection and mitigation, namely IoT devices profiling methods, state of the art in malware detection and mitigation, as well as the quest for privacy in the IoT. The main findings of this report are summarized below:

- IoT devices are considered not only a security threat, but also the main privacy disquiet, as these devices gather plenty of personal data, for example, user identity, location, energy consumption, and telephone numbers. In this case, a lot of sensitive, important, and private information can be disclosed about the daily life activities of the users including using washing machines, watching TV, and leaving or returning home.

Furthermore, these devices not only can gather users' private data but also can control their environments, and this fact represents the key concern. Thus, users are highly uncomfortable revealing personal data to public or private servers without a well-established trust model. Therefore, the lack of any well-designed IoT-oriented privacy and security techniques will prevent user adoption to any IoT technology.

- IoT devices create a complex set of network behaviours both locally and across the internet. A lack of agreed standards and the fact manufacturers have yet to sign up to the IETF MUD standard means that network security has to be driven by the Cyber-Trust platform alone rather than in concert with manufacturers. Open-source applications such as MUDgee and SiLK allow Cyber-Trust to do this, creating comprehensive network flow profiles, while additional capabilities such as IPS (dealt with separately in this document) can be run locally on the gateway and within the cloud depending on computational demands. IoT devices expose thousands of ports to arbitrary local endpoints within a smart home, and remote elements over the internet. This creates a large state space within which to conduct profiling and analysis operations that are timely, accurate and within the bounds of available resources.
- The use of protocols such as MUD enable Cyber-Trust to enhance device network behaviours, and there is a natural synergy between device and network at this point to enable a mutually-beneficial enhancement to both network and device monitoring. However, MUD only goes so far, and especially with complex devices like the Amazon Echo, or STUN protocol-based IoT devices like IP cameras, there is a necessity to focus attention and development activities to defining how behavioural profiles can be determined that can feed capabilities such as anomaly-based IPS and even raw ICMP alerts with a high degree of accuracy. Flow relationships and associated clustering activities are useful techniques for rendering intelligible the complexity of an IoT network within the paradigm of the IP infrastructure, however, there remains the hard constraint of computational resources and this will be expanded upon as development activities begin.
- Intrinsically malware is a malicious piece of software designed by malware authors with the intention to destroy, destruct and damage the normal functionality of Information Systems. It has become the most disastrous and pernicious cyberweapon like never before and with the passage of time, there is a major increase in the development of malware and its complexity. Currently, there is a drastic rise in the production of malware variants, with an exponential increase in the power shell malware.
- IoT is a multi-domain technology with a network of devices and services to exchange information. Each domain can apply its security, privacy, and trust requirements. The privacy-preserving data mining techniques, propose the setup for minimizing the risks of revealing sensitive information and offer sensitive content analysis. It introduced some of the privacy and security issues in IoT technological aspects. Although an abundance of research has already been proposed concerning privacy, many topics still need further investigation. Privacy in data collection, as well as data sharing and management, and data security matters remain open research issues to be fulfilled.

- Privacy is a concern, user-names, hostnames, IP addresses are all privacy-sensitive user identifiers, and even non-identifying fields such as time-stamps, URLs, Payloads and attack signatures can be considered as privacy-sensitive even though they are not user identifiers in themselves. To counter this, hash-functions, homomorphic encryption, perturbation and more are all techniques that can be employed within the gateway and cloud capabilities to preserve user privacy, however proper assessment needs to be made of the effects of privacy-preservation techniques on the flow, packet and deep packet inspection analysis techniques before architectural integration into the network can occur, i.e. if a chosen technique obfuscated ports or destination IPs does that affect the ability to properly measure behaviour and so anomalous activity.

6. References

- [1] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Futur. Gener. Comput. Syst.*, vol. 82, pp. 761–768, May 2018.
- [2] A. A. Parmar, U. P. Rao and D. R. Patel, "Blocking Based Approach for Classification Rule Hiding to Preserve the Privacy in Database," in *2011 International Symposium on Computer Science and Society*, Kota Kinabalu, Malaysia, 2011.
- [3] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. I-diversity: Privacy beyond k-anonymity. In *ACM Transactions of Knowledge Discovery Data*, vol. 1, no. 1, p. 3, 2007.
- [4] A. Odlyzko, "Privacy, economics, and price discrimination on the Internet," in *Proceedings of the the 5th international conference*, pp. 355–366, Pittsburgh, Pennsylvania, September 2003.
- [5] A. Riahi, Y. Challal, E. Natalizio, Z. Chtourou and A. Bouabdallah, "A Systemic Approach for IoT Security," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, Cambridge, MA, USA, 2013.
- [6] A. Shah and R. Gulati. Privacy preserving data mining: Techniques, classification and implications—A survey. *International Journal of Applied Computation*, vol. 137, no. 12, pp. 40–46, 2016.
- [7] A. Shoshani. Statistical databases: Characteristics, problems, and some solutions. In *Proceedings of VLDB*, vol. 82, pp. 208–222, 1982.
- [8] A. Vedder, "KDD: The challenge to individualism," *Ethics and Information Technology*, vol. 1, no. 4, pp. 275–281, 1999. View at Publisher · View at Google Scholar ·
- [9] A. Walenstein, D. J. Hefner, and J. Wichers, "Header information in malware families and impact on automated classifiers," ed, pp. 15–22, 2010"
- [10] Acarali, D., Rajarajan, M., Komninos, N. and Herwono, I., 2016. Survey of approaches and features for the identification of http-based botnet traffic. *Journal of Network and Computer Applications*, 76, pp.1-15.
- [11] Aggarwal, G., Mishra, N. and Pinkas, B., 2004, May. Secure computation of the k th-ranked element. In *International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 40-55). Springer, Berlin, Heidelberg.
- [12] Andrea Romei and Salvatore Ruggieri, 'A Multidisciplinary Survey on Discrimination Analysis' (2014) 29 *The Knowledge Engineering Review* 582.
- [13] B. Bashari Rad, M. Masrom, and S. Ibrahim, "Camouflage in Malware: from Encryption to Metamorphism," 2012.
- [14] B. Custers, "Effects of unreliable group profiling by means of data mining," *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*): Preface, vol. 2843, pp. 291–296, 2003.
- [15] B. Schoenmakers, "Zero-knowledge", chapter in *Encyclopedia of Cryptography and Security* (Henk C.A. van Tilborg & Sushil Jajodia (eds.)), 2nd ed., Springer, pp. 1401-1403, 2011.
- [16] Ben-Or, M., Goldwasser, S. and Wigderson, A., 1988, January. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing* (pp. 1-10). ACM.
- [17] Bernardini, C., Marchal, S., Asghar, M.R. and Crispo, B., 2019. PrivICN: Privacy-preserving content retrieval in information-centric networking. *Computer Networks*, 149, pp.13-28.
- [18] Bhattacharyya, S., Katramatos, D. and Yoo, S., 2018. Why wait? Let us start computing while the data is still on the wire. *Future Generation Computer Systems*, 89, pp.563-574.
- [19] Boneh, D., Goh, E.J. and Nissim, K., 2005, February. Evaluating 2-DNF formulas on ciphertexts. In *Theory of Cryptography Conference* (pp. 325-341). Springer, Berlin, Heidelberg.
- [20] Brakerski, Z. and Vaikuntanathan, V., 2014. Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing*, 43(2), pp.831-871.
- [21] Brakerski, Z., Gentry, C. and Vaikuntanathan, V., 2014. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3), p.13.

- [22] Build a Native Android UI & iOS UI with Xamarin.Forms - Xamarin. [Online]. Available: <https://www.xamarin.com/forms>. [Accessed: 13-Oct-2017].
- [23] Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular." [Online]. Available: <https://ionicframework.com/>. [Accessed: 20-Feb-2019].
- [24] C. C. Aggarwal and P. S. Yu, "A Condensation Approach to Privacy Preserving Data Mining," in *Advances in Database Technology - EDBT 2004, 9th International Conference on Extending Database Technology*, Heraklion, Crete, Greece, 2004.
- [25] C. C. Aggarwal and P. S. Yu. A general survey of privacy-preserving data mining models and algorithms. In *Journal of Privacy-Preserving Data Mining*, pp. 11–52, Springer, 2008.
- [26] C. C. Aggarwal. *Data Mining: The Textbook*. Springer, NY, USA, 2015.
- [27] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 2, pp. 28–34, 2002.
- [28] C. Dwork. Differential privacy. In *Automata, Languages and Programming*, vol. 4052, pp. 1–12, Springer-Verlag, 2006.
- [29] C. Gentry. A fully homomorphic encryption scheme. Ph.D. dissertation, Department of Computer Science, Stanford University, Stanford, CA, USA, 2009.
- [30] Chaum, D., Crépeau, C. and Damgard, I., 1988, January. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM symposium on Theory of computing* (pp. 11-19). ACM.
- [31] Chaum, D., Crépeau, C. and Damgard, I., 1988, January. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM symposium on Theory of computing* (pp. 11-19). ACM.
- [32] Chrome.system.memory. [Online]. Available: <https://github.com/MobileChromeApps/cordova-plugin-chrome-apps-system-memory>. [Accessed: 20-Feb-2019].
- [33] Cloud. Talkin, lot past and present: The history of iot, and where its headed today, 2016, <http://talkincloud.com/cloud-computing/iot-past-and-present-historyiot-and-where-its-headed-today?page=2>.
- [34] Cyber-Trust Deliverable 2.1 threat landscape trends and methods
- [35] D. Bilar, "Opcodes as predictor for malware," *Int. J. Electron. Secur. Digit. Forensics*, vol. 1, no. 2, p. 156, 2007.
- [36] D. Evans, *The internet of things - how the next evolution of the internet is chaging everything*, "White Paper. Cisco Internet Business Solutions Group (IBSG), 2011.
- [37] D. Lyon, *Surveillance as Social Sorting*, Routledge, 2005. View at Publisher ·
- [38] D. Miorandi, S. Sicari, F. D. Pellegrini and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497-1516, 2012.
- [39] D. Storm, "MEDJACK: Hackers hijacking medical devices to create backdoors in hospital networks," *Computerworld*, 8 June 2015. [Online]. Available: <https://www.computerworld.com/article/2932371/cybercrime-hacking/medjack-hackers-hijacking-medical-devices-to-create-backdoors-in-hospital-networks.html>. [Accessed 12 Feburary 2019].
- [40] Daniel, B., "Server Log Analysis with the ELK Stack", logz.io, 2018. <https://logz.io/blog/server-log-analysis/> [Accessed 25/2/2019]
- [41] Desktop OS market share 2013-2018 | Statista, 2019. [Online]. Available: <https://www.statista.com/statistics/218089/global-market-share-of-windows-7/>. [Accessed: 19-Feb-2019].
- [42] E. Ben-Sasson, A. Chiesa, E. Tromer, M. Virza, "Succinct Non-Interactive Zero Knowledgefor a von Neumann Architecture", *Cryptology ePrint Archive*, Report 2013/879, 2013.
- [43] E. Bertino and I. N. Fovino. Information driven evaluation of data hiding algorithms. In *Proceedings of International Conference on Data Warehousing Knowledge Discovery*, pp. 418–427, 2005.
- [44] E. Bertino, D. Lin, and W. Jiang. A survey of quantification of privacy preserving data mining algorithms. In *Privacy-Preserving Data Mining*, pp. 183–205, Springer, 2008.

- [45] E. Bertino, I. N. Fovino and L. P. Provenza, "A Framework for Evaluating Privacy Preserving Data Mining Algorithms," *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 121-154, 2005.
- [46] E. Darra and S. K. Katsikas, "A survey of intrusion detection systems in wireless sensor networks," *Intrusion Detect. Prev. Mob. Ecosyst.*, pp. 393–458, 2017.
- [47] E. Konstantinou, S. Wolthusen, and R. Holloway, "Metamorphic Virus: Analysis and Detection," 2008.
- [48] Eilam and Eldad, "Reversing: The Hacker's Guide to Reverse Engineering," .
D. Bilar, "Opcodes as predictor for malware," *Int. J. Electron. Secur. Digit. Forensics*, vol. 1, no. 2, p. 156, 2007.
- [49] ElGamal, T., 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4), pp.469-472.
- [50] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 2018, pp. 178–183.
- [51] F. Kerschbaum, "Privacy Preserving computation", *Annual Privacy Forum 2012*, Springer, pp. 41-54, 2012.
- [52] F. Kerschbaum, A. Schaad, "Privacy-preserving social network analysis for criminal investigations", *Proc. of the 7th ACM Workshop on Privacy in the Electronic Society, WPES'08*, 2008.
- [53] G. L. Orgill, G. W. Romney, M. G. Bailey, and P. M. Orgill, "The urgency for effective user privacy-education to counter social engineering attacks on secure computer systems," in *Proceedings of the the 5th conference*, p. 177, Salt Lake City, UT, USA, October 2004.
- [54] G. T. Marx, "The surveillance society: the threat of 1984-style techniques. in. The Futurist," in June 21-6, p. 21, The surveillance society, the threat of 1984-style techniques. in. The Futurist, 1985.
- [55] Gentry, C. and Halevi, S., 2011, May. Implementing gentry's fully-homomorphic encryption scheme. In *Annual international conference on the theory and applications of cryptographic techniques* (pp. 129-148). Springer, Berlin, Heidelberg.
- [56] Gentry, C. and Halevi, S., 2011, October. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science* (pp. 107-109). IEEE.
- [57] Gentry, C., 2009, May. Fully homomorphic encryption using ideal lattices. In *Stoc* (Vol. 9, No. 2009, pp. 169-178).
- [58] Gentry, C., 2010. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3), pp.97-105.
- [59] Gentry, C., Sahai, A. and Waters, B., 2013, August. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual Cryptology Conference* (pp. 75-92). Springer, Berlin, Heidelberg.
- [60] Gerhards, R., "RFC 5424: The Syslog Protocol", Network Working Group, March 2009, <https://tools.ietf.org/html/rfc5424> [Accessed 25/2/2019].
- [61] Global IT Research Institute, IEEE Communications Society, and Institute of Electrical and Electronics Engineers, "The IEEE 20th International Conference on Advanced Communications Technology : "Opening New Era of Intelligent Things!" : ICACT 2018 : Elysian Gangchon, Chuncheon, Korea (South) : Feb. 11-14, 2018 : proceeding & journal."
- [62] Goldreich, O., Micali, S. and Wigderson, A., 1987, January. How to play any mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing* (pp. 218-229). ACM.
- [63] Guo, W., Mahendran, V. and Radhakrishnan, S., 2018. Join and spilt TCP for SDN networks: Architecture, implementation, and evaluation. *Computer Networks*, 137, pp.160-172.
- [64] H. Kargupta , S. Datta , Q. Wang and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Third IEEE International Conference on Data Mining*, Melbourne, FL, USA, 2003.

- [65] Hamza, A., Ranathunga, D., Gharakheili, H.H., Benson, T.A., Roughan, M. and Sivaraman, V., 2019. Verifying and Monitoring IoTs Network Behaviour using MUD Profiles. arXiv preprint arXiv:1902.02484.
- [66] Hon, Millard and Singh (n 59) 23; European Commission, 'Radio Frequency Identification (RFID) in Europe: Steps towards a Policy Framework' (2011) <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=LEGISSUM:l24120a>.
- [67] <https://tools.netsa.cert.org/silk/> accessed 01/08/2017.
- [68] <https://www.iqvia.com/solutions/real-world-value-and-outcomes/privacy-preservation-and-data-linkage>
- [69] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.
- [70] I. You and K. Yim, "Malware Obfuscation Techniques: A Brief Survey," 2010.
- [71] Internet of Things (IoT): Security Analysis & Security Protocol CoAP, *International Journal of Recent Trends in Engineering and Research*, vol. 3, no. 3, pp. 417–425, 2017.
- [72] Internet Security Threat Report ISTR, 2019 [Online]. Available: <https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-executive-summary-en.pdf>
- [73] J. a P. Marpaung, M. Sain, and H.-J. Lee, "Survey on malware evasion techniques: State of the art and challenges," *Adv. Commun. Technol. (ICACT)*, 2012 14th Int. Conf., no. Mic, pp. 744–749, 2012.
- [74] J. Groth, A. Sahai, "Efficient Non-interactive Proof Systems for Bilinear Groups", *Cryptology ePrint Archive*, Report 2007/155, 2007
- [75] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2006.
- [76] J. J. Kim and W. E. Winkler. Multiplicative noise for masking continuous data. *Statist. Res. Division, U.S. Bureau Census*, Washington, DC, USA, Tech. Rep. 2003-01, 2003.
- [77] J. Liu , J. Luo and J. Z. Huang, "Rating: Privacy Preservation for Multiple Attributes with Different Sensitivity Requirements," in 2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver, BC, Canada, 2012.
- [78] J. Menn, "Social networks scan for sexual predators, with uneven results," in *Reuters*. <http://reut.rs/Nnejb7>.
- [79] J.A. Biega, K.P. Gummadi, I. Mele, D. Milchevski, C. Tryfonopoulos, G. Weikum. R-Susceptibility: An IR-Centric Approach to Assessing Privacy Risks for Users in Online Communities. *SIGIR 2016*: 365-374
- [80] John Gudge, 'Objects of Concern? Risks, Rewards and Regulation in the "Internet of Things"' (Social Science Research Network 2014) SSRN Scholarly Paper ID 2430780 12 <https://papers.ssrn.com/abstract=2430780>.
- [81] K. Alotaibi, V. J. Rayward-Smith, W. Wang and B. d. I. Iglesia, "Non-linear Dimensionality Reduction for Privacy-Preserving Data Classification," in *Privacy, Security, Risk and Trust (PASSAT)*, 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom), 2012.
- [82] K. Institution of Engineering and Technology. and A. Kahtani, *IET communications*, vol. 3, no. 12. Institution of Engineering and Technology, 2007.
- [83] K. Wong, C. Dillabaugh, N. Seddigh, and B. Nandy, "Enhancing Suricata intrusion detection system for cyber security in SCADA networks," in 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 2017, pp. 1–5.
- [84] Kent, K. and Souppaya, M., 2006. Guide to computer security log management. NIST special publication, 92.
- [85] Krystosek, P., Ott, N., Sanders, G. and Shimeall: *Network Traffic Analysis with SiLK, Analysts Handbook*. CERT® Situational Awareness Group. 2018.
- [86] L. D. Xu, W. He, and S. Li, "Internet of things in industries: a survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [87] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren. Information security in big data: Privacy and data mining. *IEEE Access*, vol. 2, pp. 1149–1176, 2014.

- [88] Lear, E., Drops, R. and Romascanu, D., 2018. Manufacturer Usage Description Specification draft-ietf-opsawg-mud-12. Date last accessed.
- [89] Lévy-Bencheton, C., Darra, E., Tétu, G., Dufay, G. and Alattar, M., 2015. Security and resilience of smart home environments good practices and recommendations. The European Union Agency for Network and Information Security (ENISA): Heraklion, Greece.
- [90] Li, K., Tian, L., Li, W., Luo, G. and Cai, Z., 2019. Incorporating social interaction into three-party game towards privacy protection in IoT. *Computer Networks*, 150, pp.90-101.
- [91] Lindell, Y., 2005. Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of Data Warehousing and Mining* (pp. 1005-1009). IGI Global.
- [92] Lindell, Y., 2005. Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of Data Warehousing and Mining* (pp. 1005-1009). IGI Global.
- [93] Lonvick, C., "RFC 3164: The BSD syslog Protocol", Network Working Group, August 2001, <https://tools.ietf.org/html/rfc3164> [Accessed 25/2/2019].
- [94] M. Abomhara and G. M. Køien, "Security and privacy in the Internet of Things: Current status and open issues," in 2014 International Conference on Privacy and Security in Mobile Systems (PRISMS), Aalborg, Denmark, 2014.
- [95] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios. Disclosure limitation of sensitive rules. In *Proceedings of Knowledge Data Engineering Exchange (KDEX) Workshop*, pp. 45–52, 1999.
- [96] M. B. Malik, M. A. Ghazi and R. Ali, "Privacy Preserving Data Mining Techniques: Current Scenario and Future Prospects," in 2012 Third International Conference on Computer and Communication Technology, Allahabad, India, 2012.
- [97] M. B. Malik, M. A. Ghazi, and R. Ali. Privacy preserving data mining techniques: Current scenario and future prospects. In *Proceedings of 3rd IEEE International Conference of Computer Communication Technology (ICCCCT)*, pp. 26–32, 2012.
- [98] M. Blum, P. Feldman, S. Micali, "Non-interactive zero knowledge and its applications", *Proc. of the 20th ACM Symp. on the Theory of Computing*, ACM, pp. 103-112, 1988.
- [99] M. Christodorescu, S. Jha, M. Christodorescu, and S. Jha, "Testing malware detectors," *ACM SIGSOFT Softw. Eng. Notes*, vol. 29, no. 4, p. 34, Jul. 2004.
- [100] M. Egele and C. Kruegel, "6 A Survey on Automated Dynamic Malware-Analysis Techniques and Tools," 2012.
- [101] M. Hildebrandt and S. Gutwirth, *Profiling the European Citizen*, Springer Netherlands, Dordrecht, 2008.
- [102] M. Hildebrandt, "Defining profiling: A new type of knowledge?" *Profiling the European Citizen: Cross-Disciplinary Perspectives*, pp. 17–45, 2008.
- [103] M. Irshad, H. M. Al Khateeb, A. Mansour, M. Ashawa, and M. Hamisu, "Effective methods to detect metamorphic malware: a systematic review," *Int. J. Electron. Secur. Digit. Forensics*, vol. 10, no. 2, p. 138, 2018.
- [104] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Proceedings of International Conference on Theory and Applied Cryptographic Techniques*, pp. 1–19, 2004.
- [105] M. Langheinrich, "Privacy by Design — Principles of Privacy-Aware Ubiquitous Systems," in *Ubicomp 2001: Ubiquitous Computing*, 2001.
- [106] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 448–457, 2001.
- [107] M. R. Chhatrapati and S. Sherasiya, "Privacy Preserving Data Mining Using Heuristic Approach," *IJIRST –International Journal for Innovative Research in Science & Technology*, vol. 1, no. 10, pp. 113-116, 2015.
- [108] M. Ritamaki and A. Ruhanen, "Embedded passive UHF RFID seal tag for metallic returnable transit items," in *Proceedings of the 2010 IEEE International Conference on RFID (IEEE RFID 2010)*, pp. 152–157, Orlando, FL, April 2010.

- [109] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in 2014 International Conference on Computing, Networking and Communications (ICNC), 2014, pp. 797–801.
- [110] Mahmoud Elkhodr, Seyed Shahrestani and Hon Cheung, 'A Review of Mobile Location Privacy in the Internet of Things', ICT and Knowledge Engineering (ICT & Knowledge Engineering), 2012 10th International Conference on (IEEE 2012).
- [111] Martini, 'DS-GVO Art. 21Widerspruchsrecht' in Paal and Pauly (eds), Datenschutz-Grundverordnung (1st edn, beck-online 2017) Rn. 37–40.
- [112] McAfee, "McAfee Labs Threats Report Malware Incidents Web and Network Threats," 2018.
- [113] Mobile OS market share 2018 | Statista, 2018. [Online]. Available: <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>. [Accessed: 19-Feb-2019].
- [114] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In Proceedings of 23rd IEEE International Conference on Data Engineering (ICDE), pp. 106–115, 2007.
- [115] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: A comparative study. In ACM Computer Survey, vol. 21, no. 4, pp. 515–556, 1989.
- [116] Niksefat, S., Kaghazgaran, P. and Sadeghiyan, B., 2017. Privacy issues in intrusion detection systems: A taxonomy, survey and future directions. Computer Science Review, 25, pp.69-78.
- [117] O. Goldreich. Secure multi-party computation. Technical Report, pp. 86–97, 1998.
- [118] O. Hasan, "A Survey of Privacy Preserving Reputation Systems", Technical Report, LIRIS UMR5205 CNRS/INSA de Lyon, 2017. Available in <https://hal.archives-ouvertes.fr/hal-01635314/document> (Last accessed: Feb. 21st, 2019).
- [119] Ohm (n 16); Kyle Ebersold and Richard Glass, 'The Internet of Things: A Cause for Ethical Concern.' (2016) 17 Issues in Information Systems http://www.iacis.org/iis/2016/4_iis_2016_145-151.pdf.
- [120] P. OKane, S. Sezer, and K. McLaughlin, "Obfuscation: The Hidden Malware," IEEE Secur. Priv. Mag., vol. 9, no. 5, pp. 41–47, Sep. 2011.
- [121] P. P. Ray, "A survey of IoT cloud platforms," Futur. Comput. Informatics J., vol. 1, no. 1–2, pp. 35–46, Dec. 2016.
- [122] P. P. Ray, "Internet of things based physical activity monitoring (PAMIoT): an architectural framework to monitor human physical activity," Proceeding IEEE CALCON, Kolkata, pp. 32–34, 2014.
- [123] P. P. Ray, "Internet of Things Cloud based smart monitoring of Air Borne PM2. 5 density level," in 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), 2016, pp. 995–999.
- [124] P. Porambage, M. Ylianttila, C. Schmitt, P. Kumar, A. Gurtov and A. V. Vasilakos, "The Quest for Privacy in the Internet of Things," IEEE Cloud Computing, vol. 3, no. 2, pp. 36-45, 2016.
- [125] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In Proceedings of PODS, 1998.
- [126] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In Proceedings of IEEE Symposium on Security and Privacy, pp. 384–393, 1998.
- [127] P. Security, "PANDALABS ANNUAL REPORT 2014," 2014.
- [128] P. Szor, The art of computer virus research and defense. Addison-Wesley, 2005.
- [129] Parsons, T., "End-to-End IoT Monitoring with Log Data", IoT zone, 2014. <https://dzone.com/articles/end-end-iot-monitoring-log> [Accessed 25/2/2019].
- [130] Patrick. Thibodeau, Online Profiling, <https://www.computerworld.com/article/2597220/retail-it/online-rofiling.html>.
- [131] Peter, C. / Balabit. "Logging IoT: Know what your IoT devices are doing", FOSDEM '18, 2018, https://archive.fosdem.org/2018/schedule/event/logging_iot/attachments/slides

- /2177/export/events/attachments/logging_iot/slides/2177/czp_fosdem2018_v1.pdf [Accessed 25/2/2019].
- [132] PhoneGap. [Online]. Available: <https://phonegap.com/>. [Accessed: 20-Feb-2019].
 - [133] Profiling and Targeting - Behavioural Advertisers Beware! <https://www.ecommercetimes.com/story/73966.html>.
 - [134] Q. Niyaz, W. Sun, A. Y. Javaid, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," 2016.
 - [135] R. Agrawal and R. Srikant. Privacy-preserving data mining. In Proceedings of the ACM International Conference on Management of Data (SIGMOD), vol. 29, no.2, pp. 439-450, 2000.
 - [136] R. Gennaro, C. Gentry, B. Parno, M. Raykova, "Quadratic Span Programs and Succinct NIZKs without PCPs", Cryptology ePrint Archive, Report 2012/215, 2012.
 - [137] R. H. Weber, "Internet of Things – New security and privacy challenges," Computer Law & Security Review, vol. 26, no. 1, pp. 23-30, 2010.
 - [138] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computing, vol. 4, no. 11, pp. 169–180, 1978.
 - [139] R. Mendes and J. P. Vilela. Privacy-Preserving Data Mining: Methods, Metrics, and Applications. In Journal of IEEE Access, vol. 5, pp. 10562-10582, 2017.
 - [140] R. Roman , P. Najera and J. Lopez, "Securing the internet of things," Computer, vol. 44, no. 9, pp. 51-58, 2011.
 - [141] Report. Gartner, Forecast: The Internet of Things, Worldwide, The Internet of Things, Forecast, 2017.
 - [142] Rivest, R.L., Adleman, L. and Dertouzos, M.L., 1978. On data banks and privacy homomorphisms. Foundations of secure computation, 4(11), pp.169-180.
 - [143] Rivest, R.L., Shamir, A. and Adleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), pp.120-126.
 - [144] Rsyslog team, "Encrypting Syslog Traffic with TLS (SSL)", https://www.rsyslog.com/doc/v8-stable/tutorials/tls_cert_summary.html [Accessed 25/2/2019].
 - [145] S. Dua and X. Du. Data Mining and Machine Learning in Cybersecurity. CRC Press, Boca Raton, FL, USA, 2011.
 - [146] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. Communications of ACM, vol. 28, no. 6, pp. 637–647, 1985.
 - [147] S. Fletcher and M. Z. Islam. Measuring information quality for privacy preserving data mining. International Journal of Computation Theory Engineerins, vol. 7, no. 1, pp. 21–28, 2015.
 - [148] S. Goldwasser, S. Micali, and C. Racko, "The knowledge complexity of interactive proof systems", SIAM Journal of Computing, vol. 18, no. 1, pp. 186-208, 1989.
 - [149] S. Ji, Z. Wang, Q. Liu, and X. Liu. Classification algorithms for privacy preserving in data mining: A survey. In Proceedings of International Conference on Applied Computer Science, pp. 312–322, 2016.
 - [150] S. John Walker, "Big Data: A Revolution That Will Transform How We Live, Work, and Think," International Journal of Advertising, vol. 33, no. 1, pp. 181–183, 2015.
 - [151] S. Kasthuri and T. Meyyappan, "Hiding Sensitive Association Rule Using Heuristic Approach," International Journal of Data Mining & Knowledge Management Process (IJDMP), vol. 3, no. 1, pp. 57-63, 2013.
 - [152] S. Lohiya and L. Ragha, "Privacy Preserving in Data Mining Using Hybrid Approach," in 2012 Fourth International Conference on Computational Intelligence and Communication Networks, Mathura, India, 2012.
 - [153] S. M. Babu, A. J. Lakshmi, and B. T. Rao, "A study on cloud based Internet of Things: CloudIoT," in 2015 global conference on communication technologies (GCCT), 2015, pp. 60–65.
 - [154] S. M. Riazul Islam, D. Kwak, M. Humaun Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," IEEE Access, vol. 3, pp. 678–708, 2015.
 - [155] S. Sicari, A. Rizzardi, L. Grieco and A.-. Porisini, "Security, privacy and trust in Internet of Things: The road ahead," Computer Networks, vol. 76, pp. 146-164, 2015.

- [156] S. Sicari, L. A. Grieco, G. Boggia and A. Coen-Porisini, "DyDAP: A dynamic data aggregation scheme for privacy aware wireless sensor networks," *Journal of Systems and Software*, vol. 85, no. 1, pp. 152-166, 2012.
- [157] S. Taneja, S. Khanna, S. Tilwalia and A. , "A Review on Privacy Preserving Data Mining: Techniques and Research Challenges," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2310-2315, 2014.
- [158] Sahay, R., Meng, W. and Jensen, C.D., 2019. The application of Software Defined Networking on securing computer networks: A survey. *Journal of Network and Computer Applications*.
- [159] Saidi, H., Porras, P., & Yegneswaran, "Experiences in malware binary deobfuscation". *Virus Bulletin*. V. (2010).
- [160] Salvatore Ruggieri, Dino Pedreschi and Franco Turini, 'Data Mining for Discrimination Discovery' (2010) 4 *ACM Transactions on Knowledge Discovery from Data (TKDD)* 9.
- [161] SandraWachter, Brent Mittelstadt and Luciano Floridi, 'Transparent, Explainable, and Accountable AI for Robotics' (2017) 2 *Science Robotics*.
- [162] Sarah Johanna Eskens, 'Profiling the European Citizen in the Internet of Things: HowWill the General Data Protection Regulation Apply to This Form of Personal Data Processing, and How Should It?' (Social Science Research Network 2016) SSRN Scholarly Paper ID 2752010 <https://papers.ssrn.com/abstract=2752010>.
- [163] Scott R Peppet, 'Regulating the Internet of Things: First Steps toward Managing Discrimination, Privacy, Security and Consent' (2014) 93 *Tex. L. Rev.* 85.
- [164] Smart, N.P. and Vercauteren, F., 2010, May. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography* (pp. 420-443). Springer, Berlin, Heidelberg.
- [165] Sophie C Boerman, Sanne Kruijkemeier and Frederik J Zuiderveen Borgesius, 'Online Behavioural Advertising: A Literature Review and Research Agenda' (2017) 0 *Journal of Advertising* 1.
- [166] T. Jahan, G. Narsimha and C. V. Guru Rao, "Data perturbation and feature selection in preserving privacy," in *Wireless and Optical Communications Networks (WOCN)*, 2012.
- [167] T. Ogino, "An Evaluation of Intrusion Detection System on Jubatus," Springer, Cham, 2015, pp. 359–364.
- [168] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: a top-down survey," *Computer Networks*, vol. 67, pp. 104–122, 2014.
- [169] Tal Zarsky, 'Transparent Predictions' (2013) 2013 *University of Illinois Law Review* http://papers.ssrn.com/sol3/Papers.cfm?abstract_id=2324240.
- [170] The importance of an integrated security strategy - Information Age." [Online]. Available: <https://www.information-age.com/importance-integrated-security-strategy-123470801/>. [Accessed: 14-Feb-2019].
- [171] The Shift to Linux Operating Systems for IoT | IoT For All. [Online]. Available: <https://www.iotforall.com/linux-operating-system-iot-devices/>. [Accessed: 19-Feb-2019].
- [172] TuyAPI." [Online]. Available: <https://github.com/codetheweb/tuyapi>. [Accessed: 19-Feb-2019].
- [173] V. S. Verykios. Association rule hiding methods. In *Wiley Interdisciplinary Rev., Data Mining Knowledge Discovery*, vol. 3, no. 1, pp. 28–36, 2013.
- [174] Vasiliadis, G., Koromilas, L., Polychronakis, M. and Ioannidis, S., 2014. {GASPP}: A GPU-Accelerated Stateful Packet Processing Framework. In 2014 {USENIX}{ATC} 14 (pp. 321-332).
- [175] Wang, Q., Hassan, W.U., Bates, A. and Gunter, C., 2018, February. Fear and Logging in the Internet of Things. In *Network and Distributed Systems Symposium*.
- [176] X. Qi and M. Zong. An Overview of Privacy Preserving Data Mining. In *Proceedings of International Conference on Environmental Science and Engineering (ICESE)*, pp. 1341 – 1347, 2011.
- [177] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proceedings of VLDB*, pp. 139–150, 2006.
- [178] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177-206, 2015.

- [179] Y. Wang, Y. Xiang, W. Zhou, and S. Yu, "Generating regular expression signatures for network traffic classification in trusted network management," J. Netw. Comput. Appl., vol. 35, no. 3, pp. 992–1000, May 2012.
- [180] Yao, A.C.C., 1986, October. How to generate and exchange secrets. In | 27th Annual Symposium on Foundations of Computer Science (pp. 162-167). IEEE.
- [181] Your. Proofprint, Fridge is Full of SPAM, 2014, <https://www.proofpoint.com/us/threat-insight/post/Your-Fridge-is-Full-of-SPAM>.
- [182] Zhang, B., Mor, N., Kolb, J., Chan, D.S., Lutz, K., Allman, E., Wawrzyniek, J., Lee, E. and Kubiawicz, J., 2015. The cloud is not enough: Saving iot from the cloud. In 7th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 15).
- [183] J. Leyden, (2017) We found a hidden backdoor in Chinese Internet of Things devices – researchers. Available: https://www.theregister.co.uk/2017/03/02/chinese_iot_kit_backdoor_claims/ [Accessed: 4-Mar-2019]

